Intelligent Autonomy for Unmanned Surface and Underwater Vehicles

Terry Huntsberger, Gail Woodward
Autonomous Systems Division
NASA Jet Propulsion Laboratory, California Institute of Technology
Pasadena, CA, USA
{terry.huntsberger, gail.woodward}@jpl.nasa.gov

***Abstract*—As the Autonomous Underwater Vehicle (AUV) and Autonomous Surface Vehicle (ASV) platforms mature in endurance and reliability, a natural evolution will occur towards longer, more remote autonomous missions. This evolution will require the development of key capabilities that allow these robotic systems to perform a high level of on-board decision-making, which would otherwise be performed by human operators. With more decision making capabilities, less *a priori* knowledge of the area of operations would be required, as these systems would be able to sense and adapt to changing environmental conditions, such as unknown topography, currents, obstructions, bays, harbors, islands, and river channels. Existing vehicle sensors would be dual-use; that is they would be utilized for the primary mission, which may be mapping or hydrographic reconnaissance; as well as for autonomous hazard avoidance, route planning, and bathymetric-based navigation. This paper describes a tightly integrated instantiation of an autonomous agent called CARACaS (Control Architecture for Robotic Agent Command and Sensing) developed at JPL (Jet Propulsion Laboratory) that was designed to address many of the issues for survivable ASV/AUV control and to provide adaptive mission capabilities. The results of some on-water tests with US Navy technology test platforms are also presented.**

Keywords-maritime autonomy, onboard planning/re-planning, environmental sensing, benthic habitat monitoring

I. INTRODUCTION

Autonomous (and semi-autonomous) surface vehicles (ASV's) and underwater vehicles (AUV's) have a broad range of uses for both military as well as civilian activities. These include but are not limited to, search and rescue operations, sea surface and underwater scientific research, humanitarian mine detection and disposal, and surveillance/reconnaissance. There are a number of areas where ASV's and AUV's are already being used for deep water archaeological exploration of ancient shipwrecks [1], cooperative multi-vehicle adaptive sampling of the underwater environment tied to predictive ocean models [2], data driven adaptive sampling of large marine habitats [3], survey of a hydroelectric dam [4], monitoring the environment for dynamic events such as the carp incursion into the Great Lakes [5], mapping of underwater plumes after oil platform accidents [6], and monitoring phytoplankton blooms [7, 8]. An example of an AUV (REMUS100) that is being used for bi-monthly water quality monitoring at the Kilo Nalu Nearshore Reef Observatory by the University of Hawai'i at Manoa on February 25, 2011 is shown in the upper left of Fig. 1 [9]. The onboard instruments produce data for water quality (fluorescence, and optical backscatter at 2 wavelengths: 470nm, 700nm), salinity, temperature, bathymetry, current profile and acoustic backscatter, and side scan sonar. Examples of the data for salinity, temperature, and backscatter taken along a fixed set of transects for a depth range between 2-4 meters are shown going clockwise from the upper right in Fig. 1. The AUV follows a pre-planned set of transects and the onboard autonomy is not capable of responding to "interesting" events such as episodes of freshwater runoff, sewage spill or other event which could lead to pollution or poor water quality. In addition, the AUV is constrained to stay away from the bottom because it doesn't have any onboard hazard detection or avoidance capabilities.

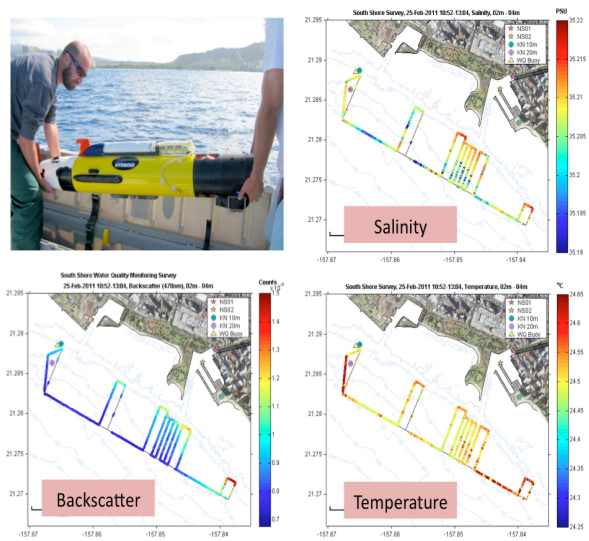

Figure 1. REMUS100 AUV being used for bi-monthly water quality monitoring off O'ahu's south shore on February 25, 2011. Clockwise from the upper left: Hand launch of the AUV; salinity measurements taken along a series of pre-planned transects at a depth of 2-4 meters; temperature along the same legs; backscatter along the same legs. (http://www.soest.hawaii.edu/OE/KiloNalu)

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Funding was provided by the Office of Naval Research, DARPA, NUWC-NPT, and Spatial Integrated Systems, Inc.
© 2011 California Institute of Technology. Government sponsorship acknowledged.

Damage monitoring and mapping of coral reef environments throughout the world is becoming important because they can serve as an early warning indicator for potential global climate change effects such as increased acidification of surface water due to CO_2 dissolving and bleaching from increased sea surface temperatures [10]. Long term monitoring of the coral reefs will require systematic measurements of state variables such as coral cover, macroalgal cover, and size-frequency distributions and key processes such as recruitment rates, individual growth rates, and partial mortality rates across a range of spatial scales over time [11].

Obtaining such systematic quantitative measurements is a difficult process using traditional methods such as SCUBA (Self Contained Underwater Breathing Apparatus) based quadrats, photo-quadrats, line intercept transects, or photo transects due to the large areas that need to be covered. In addition, a full benthic habitat characterization or even a REA (Rapid Ecological Assessment) survey should include quantitative data from not just coral, but all wildlife including fish, invertebrates, and algae (see Fig. 2). A relatively coarse level analysis of remotely sensed data such as IKONOS and aerial photography can be used to direct the placement for more detailed studies [12-14], and with the appropriate onboard autonomy, an ASV or AUV could then perform the detailed survey possibly in conjunction with humans [15-17] with up to an order of magnitude improvement in time [18].

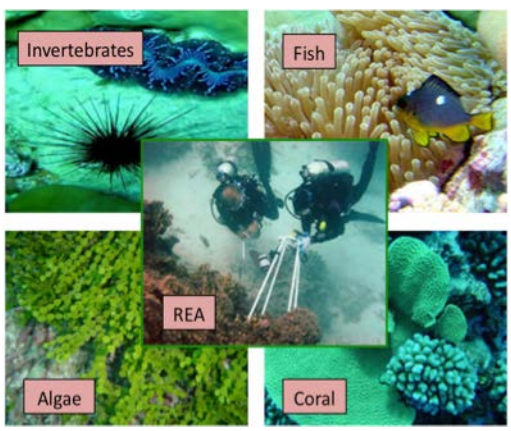

Figure 2. Component surveys needed for a full benthic habitat characterization (http://www.pifsc.noaa.gov/cred/eco_assess.php). Usually, the domain experts in each of these areas collects the data manually using quadrats or linear intercept transects on-site. Data acquired using photo transects is most often used only for archiving and manual off-line analysis. An AUV with onboard pattern recognition and adaptive control enables broader detailed coverage during photo transects, as well as potential coverage of all of the necessary components.

Control for a truly autonomous ASV or AUV will of necessity include reasoning/planning, perceptual, and behavioral components coupled with memory in the form of a world model. This type of autonomy is sometimes termed *intelligent autonomy*. Several key aspects of an intelligent autonomy approach to ASV/AUV control include:

- the handling of the inherently uncertain nature of the dynamic sea surface and underwater environments for operations,
- sensing for hazard detection/avoidance and situational awareness,
- behaviors for obeying the rules-of-the-road (COLREGS) during interactions with other manned and unmanned vehicles,
- adaptive behavior for opportunistic science,
- cooperation among heterogeneous vehicles on the sea surface as well as underwater and in the air,
- onboard resource-based planning for mission operations,
- integrated system health maintenance for long duration missions, and
- the human operator command interface.

JPL has developed a tightly integrated instantiation of an autonomous agent called CARACaS (Control Architecture for Robotic Agent Command and Sensing), a block diagram of which is shown in Fig. 3, to address many of the issues for survivable, autonomous ASV/AUV control [19-21]. CARACaS has undergone numerous sea tests on US Navy ASV's and AUV's over the last seven years. Among the technologies demonstrated are high speed static hazard (buoys, bridges) detection and avoidance, cooperative operations between multiple heterogeneous vehicles, cross-cueing and fusion of multi-modal sensors, adaptive onboard resource-based mission planning/re-planning, COLREGS-compliant navigation, and adaptive behavior in response to threats.

CARACaS is composed of sub-systems including the (1) Behavior Engine running under the Application Level (APL) of the R4SA (Robust, Real-Time, Reconfigurable Robotics Software Architecture) which is the real-time core of the system; (2) Dynamic Planner Engine that is currently using

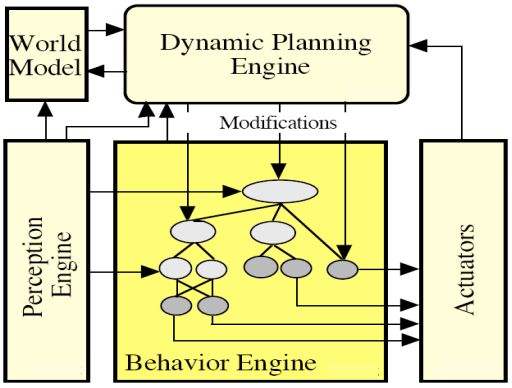

Figure 3. Block diagram of CARACaS. The network in the Behavior Engine is built from primitive (dark gray) and composite (light gray) behaviors. The Dynamic Planning Engine interacts with the network at both the primitive and composite behavior levels.

CASPER (Continuous Activity Scheduling Planning Execution and Re-planning) as the onboard planner/re-planner [22]; and (3) Perception Engine that is currently using the JPL Hammerhead wide baseline stereo [23] and the 360 Degree Sensor Head [24] systems on the ASV's and the iPUMA sonar system on the AUV's for hazard detection/avoidance and situational awareness (SA). The World Model serves as a common access point for vehicle state information,

mission/science goals and plans for both the vehicle and other agents if in a cooperative mission, vehicle health, and any appropriate maps for navigation and science data acquisition.

The next section discusses related autonomy work and is followed by a detailed description of CARACaS. Adaptation of a number of NASA technologies such as opportunistic science targeting used on MER (Mars Exploration Rovers) and wavelet based classification for coral reef monitoring is then presented. Some baseline autonomy experimental results on ASV/AUV missions are then given, followed by conclusions and future directions.

II. Related Work

A. Maritime Autonomy in General

A good overall review of autonomy for ASV's and AUV's can be found in [17]. Work reported by Benjamin, *et al.* gave details of successful in-water demonstrations of a behavior-based system that had the rules-of-the-road explicitly built into the behavior base [25, 26]. The vehicles in the demonstrations talked to each other using a wireless link in order to share position information. Larson, Bruch, and Ebken recently reported on a behavior-based hazard avoidance (HA) system for USV's that combines deliberative path planning with reactive response to close-in dynamic obstacles [27]. Their system used Digital Nautical Charts (DNC) for the initial long range path planning coupled with a passive stereo system developed at JPL for the reactive control. The current implementation of their system does not link any resource use based planning into the HA behavior.

An active sensor approach to hazard detection using a laser range finder for navigation was reported by Jimenez, *et al.* [28]. Snyder, *et al.* successfully demonstrated the components needed for autonomous in-water navigation in harbor and riverine environments [29]. Their system used 6 cameras arranged as a 360-degree color sensor coupled with sky/sea/shoreline segmentation, optic flow, and structural model techniques to determine the relative position of obstacles and safe paths.

Nervegna and Ricard recently described their simulation work in higher-level command and retasking of multiple heterogeneous air, surface, and underwater vehicles [30]. Their risk-aware mixed-initiative dynamic replanning (RMDR) system uses a mixed initiative interaction module (MIIM) for the operator interface and a dynamic replanning and situation assessment (DRASA) for onboard autonomous control.

B. Specific Applicable Autonomy

Of particular note in this area is the recent work done by Patterson and Relles [18] that compared an autonomous survey of Bonaire's reefs in the Caribbean to one that had previously been done in the same area. The resurvey used divers to ground truth the data acquired by the AUV's. The AUV's successfully mapped the bathymetry, and collected data on bottom type, fishes, coral cover and community type at locations along the entire leeward coastline. The AUV's were able to cover a much wider area in a shorter period of time than the traditional diver surveys (order of magnitude improvement), and the work established a framework for joint man/AUV survey operations. Also of note is the work of Rigby, Pizarro, and Williams [3] where an algorithm for adaptive mapping of a benthic habitat using Gaussian process classification to maximize the coverage of high science value areas was presented. Related to this is a recent paper by Bender, et al. [31] in which a Gaussian mixture model was used for classification, followed by the development of a spatial model based on a Gaussian process classifier. The coverage of a complicated reef habitat was superior to that of a pre-programmed transect survey.

Another recent advance is the work of Carreras, et al. [32] where onboard learning was used to augment the existing behavior base of the AUV. This methodology opens up the possibility of systems being able to adapt to unexpected occurrences. Similar work was done by Huntsberger [33] for the onboard learning of previously unknown patterns of behavior by other vehicles.

III. CARACaS Overview

The core component of CARACaS is R4SA which is responsible for synchronization of all the sub-systems and scheduling the processes underlying the Behavior Engine. R4SA was designed as an adaptable real-time embedded system and adheres to design principles of low coupling, high cohesion and testability concepts, as well as supports basic properties of responsiveness, timeliness, schedulability, predictability, and determinism. It is a modular, reconfigurable and expandable real-time embedded system, that accommodates various computational and interface needs to autonomous and commanded robotics systems. One of the main features of the R4SA system is its capability for fast prototyping and system integration that facilitates interfacing to various hardware and software systems providing closed loop controls.

A. R4SA

As shown in Fig. 4, the R4SA architecture is a 3-tiered architecture running under an overall system-level control. It is implemented using ANSI 'C'. The System layer is the basic scheduler that controls execution. The DD (Device Driver) Layer contains hardware interfaces (to motor controllers and analog/digital conversion). The DL (Device Layer) is an abstraction to generalized interfaces (not hardware dependent) to other devices, such as sensors (cameras, IMU, power, etc) as well as basic control for all the actuators in the system (including kinematics). The top layer is the APL (Application Layer) that contains all behaviors and application-related controls (e.g. high-level command sequences, telemetry interfaces, navigation).

1) System

The System provides basic functionality that is needed to run a real-time deterministic robotics system and is completely target independent. It consists of the following components:

- Command Processor - executes commands that are received from the GUI (Graphical User Interface) or internally through another thread running on the same hardware system. The commands that are accessible from the GUI are contained in the Command Dictionary.

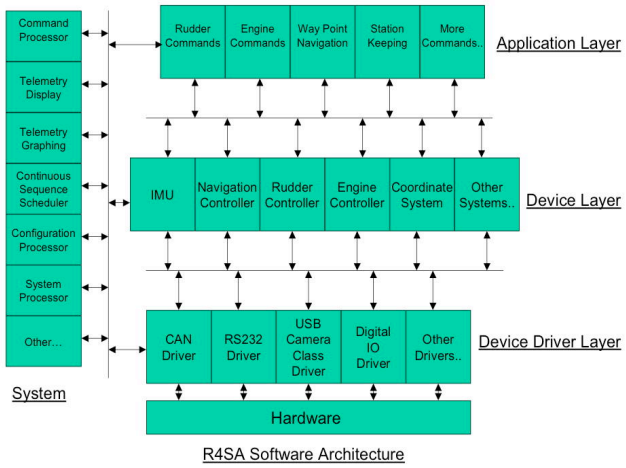

Figure 4. High level overview of the R4SA, a real-time, embedded kernel that shows the 3-tiered organizational structure that is coordinated through multiple schedulers running in the System Layer.

- Telemetry Display and Data Logging – displays and logs vehicle state information and can be configured to run at any desired rate.
- Continuous Sequence Scheduler - executes the hard real-time continuous sequences that the CARACaS behaviors run under, interfaces to the outside world, and maintains the file transfer capabilities for the system.
- Configuration Processor - uses the configuration files to setup vehicle specific parameters such as gains, communication addresses, and control bus interfaces at run time.
- System Processor - coordinates the synchronization of all of the threads.

The System processes are initialized at startup and include all of the periodic, real-time, continuous sequences, application layer, and PTPI (Process-To-Process-Interface) shared memory threads. This initialization process also interprets the symbol table for function calls and scheduling, and starts all sockets based on input from a configuration file. In order to monitor and modify parameter values during run time, the system initialization process occurs BRT (before real time). Each of the layers in R4SA has a series of initializations that are adjusted depending on which modules and interfaces need to be supported by the system. The majority of the development for autonomy code is done within the Application Layer.

PTPI is a client and server protocol that uses the QNX Real-Time Operating System (RTOS) shared memory interface for various data transfers between R4SA and other programs running as external processes that communicate with R4SA. The PTPI server interfaces with R4SA's local capabilities for adding, running and aborting commands. It also provides a mechanism to return the status of command execution by the R4SA command processor and the vehicle state data by request from the client. The PTPI client is linked with other programs running separately from R4SA that need to communicate with R4SA for commanding or monitoring state.

2) Device Driver (DD) Layer

The DD Layer provides a low-level software interface to the PC104+ hardware through ISA or PCI buses. The DD Layer also provides real-time access to hardware IO (Input/Output) such as digital IO (DIO) boards, analog to digital (A2D) boards, and the control bus interface boards. The DD Layer provides an interface to the Device Layer for manipulating the hardware. Each DD Layer component can be added or removed from compilation by using compile switches in the target configuration file. As in the System Layer, the DD Layer components are configured and initialized at real-time in the BRT mode. Actuator interfaces are added using the appropriate ICD (Interface Control Document) for any control bus.

3) Device Layer (DL)

The DL provides a more complex level of abstraction to the APL by combining robot specific algorithms and DD layer interfaces to the hardware. Some of the DL components are robot specific kinematics, coordinate systems, motion profilers, and controllers. The DL provides periodic and asynchronous operations with components configured and initialized by using configuration files in the BRT mode. The DL contains vehicle specific code, which is configured using configuration files during the compilation process. For example a climbing algorithm is specific to a climbing robot and it would not normally be part of ASV code.

4) Application Layer (APL)

The APL provides high-level functionality specific to or common between robots by using the DL. The APL algorithms are interfaced to the outside world by using a Command Processor specific interface definition. APL component algorithms can utilize each other to create very complex operations. For example, the hazard avoidance algorithm calls throttle/rudder sequence algorithms in order to provide safe vehicle navigation. The APL components are vehicle specific and components are included in the compilation process using the compile time configuration file. Each component in the APL layer is initialized and configured in BRT during startup by using configuration files.

B. CARACaS Specifics

The overall system architecture of CARACaS is comprised of the following:

- Behavior Engine: Runs in the Application Layer of the Robust Real-Time Reconfigurable Robotics Software Architecture (R4SA),
- Dynamic Planning Engine: CASPER runs under a separate thread and talks to R4SA through the shared memory Process-To-Process Interface (PTPI)
- Perception Engine: JPL Hammerhead stereo and the 360 degree sensor head servers (on ASV's) or the sonar and video servers (on AUV's) talk to R4SA through a standard socket interface.

The following two sub-sections will concentrate on the Behavior Engine and the Dynamic Planning Engine, since they are the key elements of the autonomy needed for benthic

habitat surveys. The Perception Engine is platform specific and will be discussed in the context of onboard pattern recognition capabilities for adaptive sampling of the habitat.

1) Behavior Engine

The Behavior Engine in CARACaS leverages the results of previous efforts at JPL in the multi-agent control architecture CAMPOUT (Control Architecture for Multi-robot Planetary Outposts) [34] in order to develop behavior composition and coordination mechanisms. Behaviors are built using the continuous sequence code constructs (running at a hard real-time rate) under R4SA. This allows the algorithm developer to control when the behaviors are actively controlling the actuators on the system by dynamically enabling active behaviors and putting inactive behaviors to sleep.

This enabling/disabling mechanism is performed through a behavior arbitration mechanism. There are a number of arbitration mechanisms supported by the Behavior Engine in CARACaS including subsumption [35], voting [36], and interval programming (IvP) [37]. These arbitration mechanisms can be mixed in a modular way within composite behaviors. As an example, the hazard avoidance/waypoint navigation behaviors are currently handled within the code using the subsumption approach. This process is shown in Fig. 5, where the activation of the *Hazard Avoidance* behavior (the presence of hazards along the current path is indicated using information from the Perception Engine) inhibits (over-rides) the rudder/throttle commands from the *Go_to_Waypoint* behavior. The *Hazard Avoidance* behavior is still aware that there is a waypoint goal and plans its safe paths accordingly.

An ASV navigating in close environments is required to follow the COLREGS rules for safe navigation [38]. CARACaS blends the *COLREGS* and *Navigation* behaviors using a Velocity Obstacles approach [39] shown graphically for a crossing scenario in Fig. 6. The *Navigation* behavior produces an output for the safe path that is analyzed as part of the blended control output. The Figure is color coded for interpretation of the recommended heading from each component (red for hazard, purple for COLREGS, green for safe, and shades of yellow for a buffer zone).

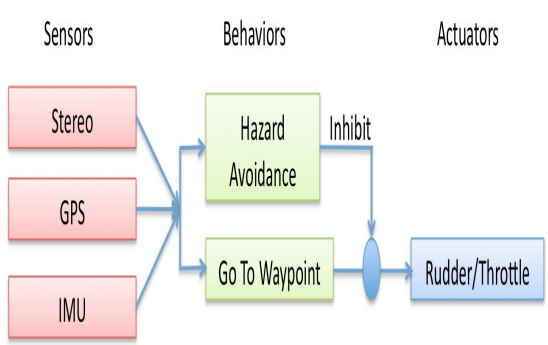

Figure 5. Behavior arbitration in the *Navigation* behavior. Conflicts between the *Hazard Avoidance* and *Go_to_Waypoint* behaviors is resolved using the subsumption approach, where the *Hazard Avoidance* behavior always gets precedence over the *Go_to_Waypoint* behavior. The *Hazard Avoidance* behavior performs a waypoint goal-biased search for clear paths. This ensures vehicle safety by avoiding collisions, while still making progress towards the goal.

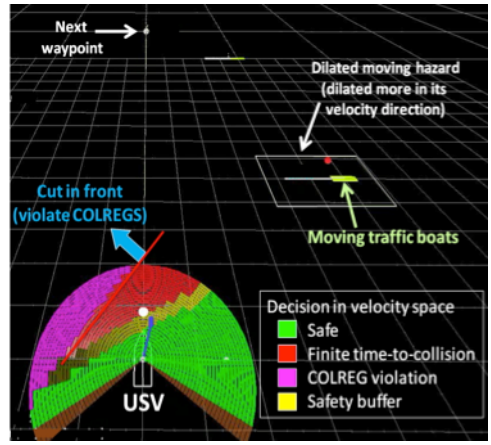

Figure 6. Snapshot of a simulation run where the ASV is encountering a COLREGS crossing situation. The ASV must give way to the vehicle coming from the starboard side, and pass behind. Color coding is used to designate what behaviors apply in the direction of movement (purple indicates that turning to port will violate COLREGS, red indicates that maintaining a straight course will result in a collision, and safe travel directions are

2) Dynamic Planning Engine

The Dynamic Planning Engine leverages the CASPER (Continuous Activity Scheduling Planning Execution and Re-planning) continuous planner developed at JPL (overview shown in Fig. 7) [22]. Given an input set of mission goals and the autonomous vehicle's current state, CASPER generates a plan of activities that satisfies as many goals as possible while still obeying relevant resource constraints and operation rules. A description of the autonomous vehicle, including resources and state information, as well as applicable mission and operations rules is encoded in the planner's ASPEN (Automated Scheduling and Planning Environment) Modeling Language (ML). Plans are dynamically updated using an iterative repair algorithm that classifies plan conflicts (such as a resource over-subscription) and resolves them individually by

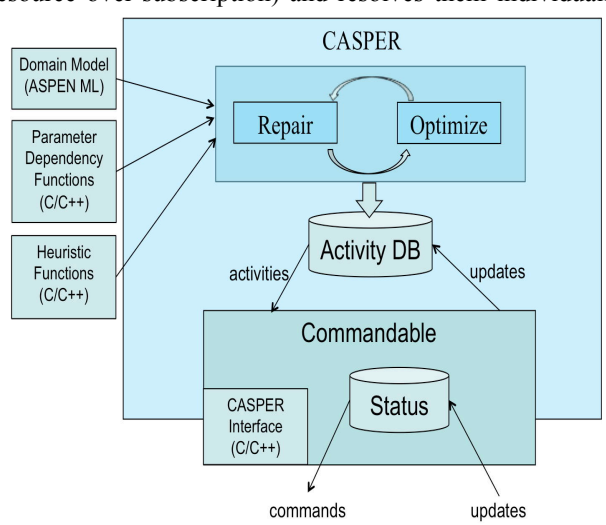

Figure 7. High level overview of the CASPER organization. The onboard algorithms use iterative repair algorithms to provide a resource based analysis at mission start and to re-plan as the mission progresses.

performing one or more plan modifications. CASPER takes a most-committed, local, heuristic *iterative repair approach* to producing and modifying plans.

a) Activity Database

The central data structure in CASPER is an activity. An activity represents an action or step in a plan/schedule. An activity has a start time, end time, and a duration. Activities can use one or more resources. All activities in a plan/schedule are elements of the Activity Database (ADB), which maintains the state of all of the activities in the current plan/schedule, and serves as the integrating component that provides an interface to all of the other classes.

b) Temporal Constraint Network

A Temporal Constraint Network (TCN) is a graph data structure that represents temporal constraints between activities. A temporal constraint describes the temporal relationship between an activity and other activities and/or the scheduling horizon, and imposes an ordering on the set of activities. The TCN implements a Simple Temporal Problem and represents a set of constraints, all of which must be satisfied at any given time, i.e., it represents the conjunct of all active constraints between activities in the ADB. Activities are represented in the TCN as pairs of time points, where each time point corresponds to the beginning or end of an activity, and the edges in the TCN graph represent the constraints on the temporal distance between the time points.

c) Resource Timelines

Resource timelines are used to reason about the usage of physical resources by activities. Capacity conflicts are detected if the aggregate usage of a resource exceeds its capacity at any given time. Several subclasses of resource timelines are implemented, including depletable resource timelines used to model consumable resources (e.g., fuel), and non-depletable resources that are used to model resources which are not actually consumed by usage, but are instead "reserved" for a period of time (e.g., a piece of equipment).

d) State Timelines

State timelines represent arbitrary attributes, or states, that can change over time. Each state can have several possible values; at any given time, a state has exactly one of these values. Activities can either change or use states. For example, a *start_engine* activity would set the state of the engine to be active, while a *move* activity would require that the state of the engine to be active. As activities are placed/moved in time, the state timeline updates the values of the state, and detects possible inconsistencies or conflicts that can be introduced consequently. The state timeline class will detect illegal transition sequences if they are introduced into the timeline.

e) Parameter Constraint Network

Each activity has a number of parameters that are either user-defined or computed by the system, such as start time, end time, duration, any resources it uses, any states it changes/uses, etc. In CASPER, it is possible to create dependencies between pairs of parameters within the same activity, or between pairs of parameters defined in different activities. A dependency is a constraint on the values of a parameter. These dependencies are represented and maintained in a Parameter Constraint Network (PCN). The PCN maintains all dependencies between parameters, so that at any given time, all dependency relations are satisfied.

f) Planning/Scheduling Algorithms

The search algorithm in a planning/scheduling system searches for a valid, possibly near-optimal plan/schedule. The ASPEN framework has the flexibility to support a wide range of scheduling algorithms, including the two major classes of AI scheduling algorithms: constructive and repair based algorithms. Constructive algorithms incrementally construct a valid schedule, taking care that at every step, the partial schedule constructed so far is valid. When a complete schedule is constructed, it is therefore guaranteed to be valid. Repair-based algorithms generate a possibly invalid complete schedule using either random or greedy techniques. Then the schedule is analyzed during every iteration, and repair heuristics that attempt to eliminate conflicts in the schedule are iteratively applied until a valid schedule is found.

An example of the CASPER planning/re-planning cycle is shown in Fig. 8 for an adaptive multi-area search mission. CASPER was given nine prioritized areas to survey using a REMUS600 AUV with the three mission constraints of limited battery power level, limited mission time, and limited data storage. The mission was run on the REMUS simulator with the three mission constraints used for onboard planning/re-planning. CASPER generated a plan that included a subset of the goals that maximized the priorities of the goals included in the plan while respecting mission constraints. The two goals with the lowest priorities were excluded in the initial plan due to the mission time constraint (Fig. 8a). The initial survey path is shown in green in Fig. 8a. As the mission progressed, the surveys were taking longer than expected, and CASPER re-planned while on the way to the third survey area. Goals with lower priorities were dynamically shed and the start time of

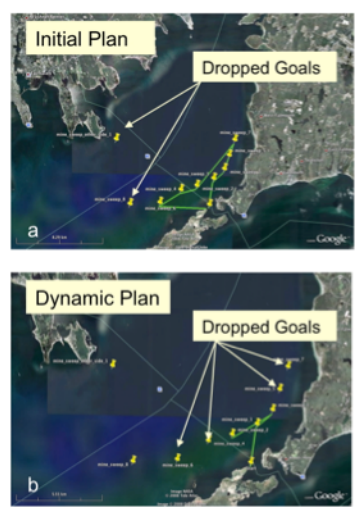

Figure 8. Example mission scenario with multiple survey areas that require a variable amount of time to survey depending on the complexity of objects found each zone. (a) CASPER drops the two lowest priority goals based on total mission time; (b) CASPER dynamically re-plans and drops more lower priority goals as the initial higher priority survey areas require more time than expected to explore.

other activities were moved back to accommodate these events (Fig. 8b). The final mission path is shown in green in Fig. 8b.

3) Perception Engine

The main objective of the Perception Engine is to produce maps that can be used as the basis for safe navigation. Any sensor suite that satisfies this objective can be employed. For an AUV, this typically would be a FLS (Forward Looking Sonar) system. There are a number of commercial systems available that have already been mounted and tested on AUV's such as the REMUS100 and REMUS600. The AUV differs from the ASV in that it is moving within a 3D environment that is constrained by the sea bottom and land masses as returned from bathymetric sensors (Fig. 9), whereas the ASV is generally travelling on the sea surface. Hazards for the ASV consist of anything above the sea surface or at a shallow enough depth compared to the draft of the boat.

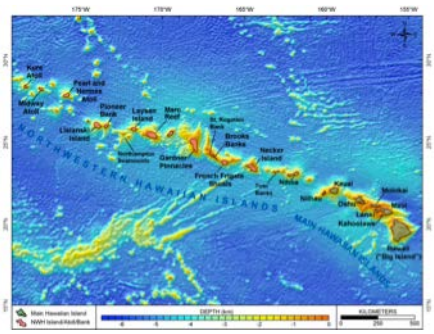

Figure 9. Bathymetric map of the Hawaiian Islands. Depth of water is indicated by shades ranging from dark blue (deepest) to orange-brown (shallowest).

A functional block diagram of the path planning process for an AUV is shown in Fig. 10. Moving from left to right, the sonar data is analyzed for potential hazards (sea bottom, floating objects within the water column, etc.) and the 3D path planner determines a safe path for the AUV based on its mission goals and resource constraints and commands the AUV controller to execute the path.

This process can be modified to include adapting to science opportunities that present themselves along the path. Such a system called AEGIS (Autonomous Exploration for Gathering Increased Science System) was developed by JPL and is currently running onboard MER (Mars Exploration Rover) [40, 41]. AEGIS uses the sensors onboard MER to flag "interesting" science targets during long drives, that could then be explored using onboard autonomous approach sequences based on the target positions. AEGIS is part of a larger framework called OASIS (Onboard Autonomous Science Investigation System) that is a suite of pattern recognition algorithms designed for working in conjunction with the onboard autonomy system to exploit science opportunities on rovers and satellites [42]. Algorithms from OASIS have been used for autonomously sensing dust devil wind storms on Mars using the MER cameras, and then sub-sampling the images to conserve bandwidth on the downlink to Earth. Coupled with the current baseline CARACaS autonomy system, OASIS gives ASV's and AUV's the capability to adapt the mission on-the-fly to return the maximum amount of science within the constraints of the mission goals and onboard resources.

Characterization of coral types for quantitative counts would be one area where such an opportunistic science

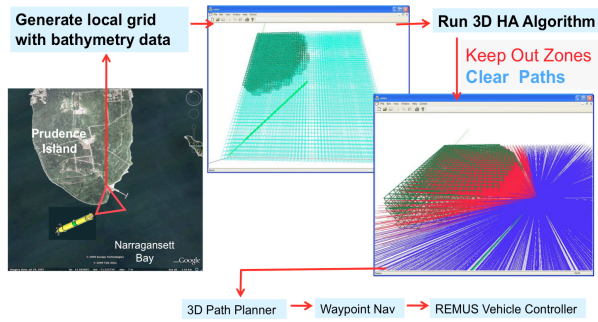

Figure 10. 3D path planning process under CARACaS for an AUV using sonar for a sample mission in Narragansett Bay. The sonar generates a 3D gridded set of data that includes bathymetry and any sensed hazards. The 3D hazard avoidance algorithm determines the safe paths and uses these for path planning.

capability could be employed. There is a wide variation in the appearance of corals both in size, shape, color, and texture. A pattern recognition algorithm that is effective at multiple scales addresses some of these issues. Such an algorithm was developed at JPL for the autonomous determination of best focus position for the fixed focus microscopic imager on MER. The algorithm is based on a wavelet texture measure [43] that was originally developed for automatic target recognition applications. A wavelet transformation is used for compression on the images that are downlinked from MER in order to conserve bandwidth. For the texture measure, the wavelet decomposition is performed on an image and the energy (additive value of the wavelet compression coefficients within sub-windows) in the horizontal, diagonal, and vertical channels is used derive a value for each channel.

An example of this process is shown on an image of a *Pocillopora damicornis* coral in Fig. 11. The wavelet transform was done to a single level, and the vertical, diagonal, and horizontal channels are shown nested in the Figure. Gray

Figure 11. Single level wavelet transform applied to an image of a pocillopora damicornis coral (image courtesy of Keoki and Yuko Stender, http://www.marinelifephotography.com). Top: Original image; Bottom: Average channel, and the vertical, diagonal, and horizontal detail channels of the wavelet transformed image. Gray scale is being used to denote the sign/value of the wavelet coefficients (white is positive, gray is negative, and black is zero).

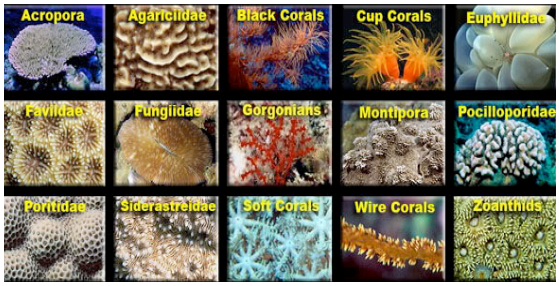

Figure 12. Sample images of fifteen types of corals found in Hawai'i (image courtesy of Keoki and Yuko Stender, http://www.marinelifephotography.com).

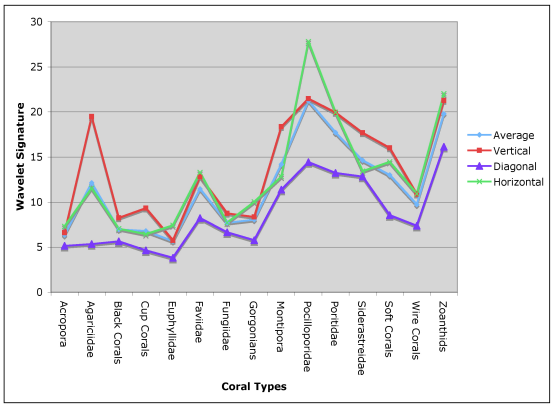

Figure 13. Wavelet texture signature for the 15 types of corals. For classification purposes a four-vector composed of the average, vertical, diagonal, and horizontal values are used.

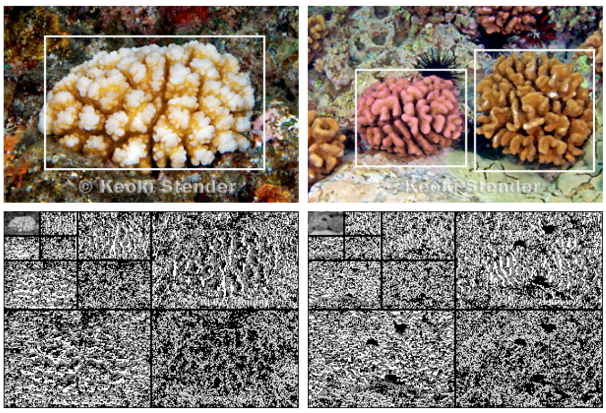

Figure 14. Test of wavelet texture signature discrimination capability for two different members of a single coral family (images courtesy of Keoki and Yuko Stender, http://www.marinelifephotography.com). Original image and thre level wavelet transform for Left: *Pocillopora damicornis*; Right: *Pocillopora meandrina*

values are used to indicate the sign of the coefficients. A collection of coral images can be used to train a pattern recognition algorithm that would run under the Perception Engine in CARACaS to detect specific instances in the field. An initial test was run on the high level set of coral types shown in Fig. 12. A plot of the texture energy components (normalized absolute value of the wavelet components over the image in each of the separate channels) with a 3 level wavelet transform is shown in Fig. 13. One last test was run to determine the discrimination capability of the wavelet texture signature for within a single family. The results of the test for *Pocillopora damicornis* (on left) and *Pocillopora meandrina* (on right) are shown in Fig. 14, where the algorithm was able to detect, differentiate between, and localize the different corals. These initial results are encouraging as a proof-of-principle demonstration of providing CARACaS onboard science target recognition capabilities similar to that already demonstrated for boat recognition using the 360 degree sensor head on a US Navy ASV [24].

IV. Baseline Experimental Studies

Experimental studies that have been run to date on the baseline CARACaS system include numerous sea tests of US Navy ASV's at Ft. Monroe, VA and San Diego, CA. The in-water tests on the REMUS600 AUV were run at the Lake Travis Test Station in Austin, TX. Due to space limitations, only a synopsis of the test results will be given here. Further details can be found in [23] and [24].

A. ASV Tests

An aerial view of the test areas in Ft. Monroe is shown in Fig. 15, where the zones are circled and two representative ASV's are shown in the inset. Of particular relevance to the type of autonomy discussed in this paper was the tests run with CASPER for adaptive area surveys in September, 2010. The runs were similar to those described earlier that were run in simulation. The primary difference was that CASPER was running onboard a live ASV in the Blue Water Zone in Fig. 15, and the demonstration of the dropping of goals due to a mission time constraint, and then adding them back in when the mission was ahead of schedule.

The ground truth results for that test are shown in Fig. 16. The ASV was originally given five areas to survey, but the mission time constraint would only allow three, so CASPER dropped the two areas with lower priority. As this mission progressed, the ASV finished the three areas (shown in green) and had time for at least one more, so one of the areas that had been dropped was added back in. The ASV finished that area (shown in blue) and still had time left to add the final area back in. On the way to the final area, the safety officer onboard the boat performed two manual over-rides (shown in red) to keep the ASV clear of a barge passing in the busy channel leading into Norfolk Harbor. The ASV kept track of the mission it was running and returned to the approach for the last area, thus demonstrating sliding autonomy. This kind of adaptability will

Figure 15. Test areas at Ft. Monroe, VA. These areas include littoral, riverine, and blue water environments.

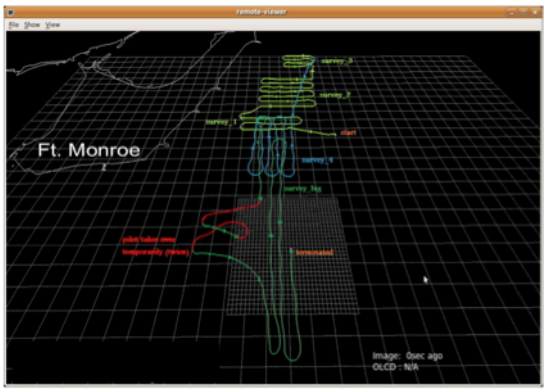

Figure 16. On-water testing of CASPER onboard an ASV running an adaptive area survey mission. The initial mission time constraint assessment was that only the top three areas could be surveyed (green), but there was time to add the middle area (blue) back in, and the final survey zone also was within the mission time constraint, so it was added back in as well.

enable opportunistic science to be done if the resources allow.

The second enabling technology was the demonstration of the hazard avoidance behavior on the REMUS600 AUV simulator. Waypoint navigation was demonstrated in-water at the Lake Travis Test Station in Austin, TX in order to ascertain the accuracy of the path following prior to any attempt at hazard avoidance. The hazard avoidance behavior was tested with a hardware-in-theloop setup using CARACaS running on a PC104+ stack connected to the REMUS600 simulator supplied by Hydroid and the iPUMA sonar simulator supplied by ARL/UTA. The test run was done using the Lake Travis bathymetry dataset produced by the iPUMA simulator. The result of the test is shown in Fig. 17 with some representative sonar inputs in the insets. Successful precision navigation while avoiding hazards is key for safe autonomous surveying of sensitive areas.

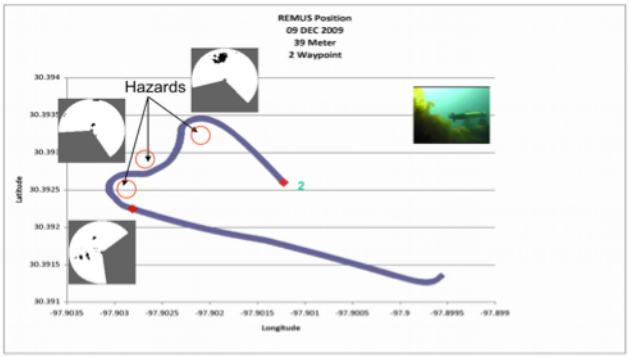

Figure 17. Simulated hazard avoidance run using iPUMA bathymetry data from Lake Travis, TX. The REMUS600 AUV successfully avoided all of the hazards that were flagged by the sonar. Representative sonar datasets are shown at the three detected hazards (white is clear, black is hazard, and gray is no data).

V. Conclusions and Current Directions

Baseline behavior for safe use of ASV's and AUV's for benthic habitat surveys has been demonstrated on the water using US Navy test boats. CARACaS has the necessary components for adaptive autonomous decision making that is based on a dynamic resource analysis while underway. In addition, the dual use of the sensors such as the FLS and any cameras on the platforms for both navigation and science can be accommodated using the AEGIS technology already demonstrated on Mars.

Incorporation of better localization for the AUV is vital for replicable results. Some of the recent work on SLAM (Simultaneous Localization and Mapping) for UAV [44, 45] will be the next enhancement to CARACaS. More extensive testing of the wavelet texture signature for coral identification is another research direction. Finally, getting more time on the water for testing is a main objective.

Acknowledgment

This work was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The continued support for on-water testing by Dr. Robert Brizzolara and Dr. Thomas Swean from the Office of Naval Research, Dr. Ali Farsaie and Rick Simon from Spatial Integrated Systems, and Mike Keegan from the Naval Undersea Warfare Center, Newport, RI is greatly appreciated. The author would also like to thank members of the JPL Intelligent Autonomy for Unmanned Vehicles Team for their tireless efforts in pushing the envelope.

References

[1] B. Bingham, B. Foley, H. Singh, R. Camilli, K. Delaporta, R. Eustice, A. Mallios, D. Mindell, C. Roman, and D. Sakellariou, "Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 1*, Vol. 27, No. 6, pp. 702-717, 2010.

[2] N. E. Leonard, D. A. Paley, R. E. Davis, D. M. Fratantoni, F. Lekien, and F. Zhang, "Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in Monterey Bay," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 1*, Vol. 27, No. 6, pp. 718-740, 2010.

[3] P. Rigby, O. Pizarro, and S. B. Williams, "Toward adaptive benthic habitat mapping using gaussian process classification," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 1*, Vol. 27, No. 6, pp. 741-758, 2010.

[4] P. Ridao, M. Carreras, D. Ribas, and R. Garcia, "Visual inspection of hydroelectric dams using an autonomous underwater vehicle," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 1*, Vol. 27, No. 6, pp. 759-778, 2010.

[5] P. Tokekar, D. Bhadauria, A. Studenski, and V. Isler, "A robotic system for monitoring carp in Minnesota lakes," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 1*, Vol. 27, No. 6, pp. 779-789, 2010.

[6] M. V. Jakuba, D. Steinberg, J. C. Kinsey, D. R. Yoerger, R. Camilli, O. Pizarro, and S. B. Williams, "Toward Automatic Classification of Chemical Sensor Data from Autonomous Underwater Vehicles," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2011)*, San Francisco, CA, September 25-30, 2011.

[7] R. N. Smith, J. Das, Y. Chao, D. A. Caron, B. H. Jones, and G. S. Sukhatme, "Cooperative multi-AUV tracking of phytoplankton blooms based on ocean model predictions," *IEEE/MTS Oceans Conference and Exhibition*, Sydney, Australia, May 2010.

[8] A. Elfes, G. W. Podnar, J. M. Dolan, S. Stancliff, E. Lin, J. C. Hosler, T. J. Ames, J. Higinbotham, J. R. Moisan, T. A. Moisan, and E. A. Kulczycki, "The Telesupervised Adaptive Ocean Sensor Fleet (TAOSF) Architecture: Coordination of Multiple Oceanic Robot Boats," *IEEE Aerospace 2008*, Big Sky, MT, 2008.

[9] http://www.soest.hawaii.edu/OE/KiloNalu/index.htm.

[10] R. W. Buddemeier, P. L. Jokiel, K. M. Zimmerman, D. R. Lane, J. M. Carey, G. C. Bohling, and J. A. Martincich, "A modeling tool to evaluate regional coral reef responses to changes in climate and ocean

chemistry," *Limnology and Oceanography: Methods*, Vol. 6, pp. 395-411, 2008.

[11] R. van Woesik, J. Gilner, and A. J Hooten, Standard Operating Procedures for Repeated Measures of Process and State Variables of Coral Reef Environments, Coral Reef Targeted Research and Capacity Building for Management Program, Currie Communications, Melbourne, Australia, September 2009.

[12] IKONOS Interpretive Rulemaking: Interpretive Rules Used in Preparation of Coral Reef Habitat Maps for the Hawaiian Islands, Report CLIN 0003, Analytical Laboratories of Hawaii, May 28, 2002.

[13] Benthic Habitats of the Hawaiian Islands: A Comparison of Accuracy of Digital Habitat Maps Prepared from Remote Sensing Imagery, Report CLIN 0004, Analytical Laboratories of Hawaii, June 24, 2002.

[14] M. Finkbeiner, B. Stevenson, and R. Seaman, Guidance for Benthic Habitat Mapping: An Aerial Photographic Approach, U.S. NOAA Coastal Services Center, March 2001.

[15] E. Steimle and M. Hall, "Unmanned Surface Vehicles as Environmental Monitoring and Assessment Tools," *OCEANS 2006*, September 18-21, 2006, pp. 1–5.

[16] J. Batlle, P. Ridao, M. Carreras, and V. Ila, "Toward Cooperative Underwater Robots for Ecological Surveillance of Marine and Underwater Environments," *3rd Workshop on European Scientific and Industrial Collaboration (WESIC 2001)*, Enschede, The Netherlands, June 27-29, 2001.

[17] G. N. Roberts and R. Sutton, Advances in Unmanned Marine Vehicles, Institution of Electrical Engineers, London, UK, 2006.

[18] M.R. Patterson and N.J. Relles, "Autonomous Underwater Vehicles resurvey Bonaire: a new tool for coral reef management," *11th International Coral Reef Symposium, Vol. 1*, Ft. Lauderdale, FL, July 7-11, 2008, pp. 539-543.

[19] T. Huntsberger, H. Aghazarian, A. Castano, G. Woodward, C. Padgett, D. Gaines, and C. Buzzell, "Intelligent Autonomy for Unmanned Sea Surface and Underwater Vehicles," *Proc. AUVSI North America 2008 Unmanned Vehicles Conference*, San Diego, CA, June 2008.

[20] T. Huntsberger, H. Aghazarian, D. Gaines, M. Garrett, & G. Sibley, "Autonomous Operation of Unmanned Surface Vehicles (USV's)," *Proc. IEEE ICRA Workshop on Robots in Challenging and Hazardous Environments*, Rome, Italy, April 2007.

[21] E. Hansen, T. Huntsberger, and L. Elkins, "Autonomous Maritime Navigation - Developing Autonomy Skill Sets for USVs," *Proc. SPIE Defense and Security Symposium 2006*, Orlando, FL, April, 2006.

[22] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve the Responsiveness of Planning and Scheduling," *Fifth International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000)*, 2000.

[23] T. Huntsberger, H. Aghazarian, A. Howard, and D. Trotz, "Stereo Vision Based Navigation for Autonomous Surface Vessels," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 2*, vol. 28, No. 1, pp. 3-18, 2011.

[24] M.T. Wolf, C. Assad, Y. Kuwata, A. Howard, H. Aghazarian, D. Zhu, T. Lu, A. Trebi-Ollennu, and T. Huntsberger, "360-Degree Visual Detection and Target Tracking on an Autonomous Maritime Patrol Vessel," *Journal of Field Robotics*, *Special Issue on Maritime Autonomy*, *Part 1*, vol. 27, No. 6, pp. 819-833, 2010.

[25] M. Benjamin, J. Curcio, J. Leonard, and P. Newman, "Navigation of Unmanned Marine Vehicles in Accordance with the Rules of the Road," *IEEE International Conference on Robotics and Automation (ICRA2006)*, Orlando FL, May 2006.

[26] M. R. Benjamin, "Multi-Objective Autonomous Vehicle Navigation in the Presence of Cooperative and Adversarial Moving Contacts," *OCEANS 2002*, Biloxi MI, October 2002.

[27] J. Larson, J. Ebken, and M.H. Bruch, "Autonomous navigation and obstacle avoidance for unmanned surface vehicles," SPIE Proc. 6230: *Unmanned Systems Technology VIII, Defense Security Symposium*, Orlando, FL, April 2006.

[28] A.R. Jimenez, R. Ceres, and F. Seco, "A laser range-finder scanner system for precise manoeuver and obstacle avoidance in maritime and inland navigation," *46th International Symposium Electronics in Marine (ELMAR-2004)*, Zadar, Croatia, June 2004.

[29] F. D. Snyder, D. D. Morris, P. H. Haley, and R. T. Collins, "Autonomous Boat," Final Report, DARPA Contract NBCHC020090, September 2004.

[30] M. Nervegna and M. Ricard, "Risk-Aware Mixed-Initiative Dynamic Replanning (RMDR) Program Update," *AUVSI Unmanned Systems North America 2006*, Orlando, FL, Aug 2006.

[31] A. Bender, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Adaptive exploration of benthic habitats using Gaussian processes," *IEEE OCEANS 2010 MTS/IEEE Seattle*, Seattle, WA, September 20-23, 2010.

[32] M. Carreras, J. Yuh, J. Batlle, and P. Ridao, "A behavior-based scheme using reinforcement learning for autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, Vol. 30, No. 2, pp. 416-427, 2005.

[33] T. Huntsberger, "Process Algebra Approach for Action Recognition in the Maritime Domain," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2011) Workshop on Advances in Marine Autonomy Research,* San Francisco, CA, September 25-30, 2011.

[34] T. Huntsberger, P. Pirjanian, A. Trebi-Ollennu, H.D. Nayar, H. Aghazarian, A. Ganino, M. Garrett, S.S. Joshi, and P.S. Schenker, "CAMPOUT: A Control Architecture for Tightly Coupled Coordination of Multi-Robot Systems for Planetary Surface Exploration," *IEEE Trans. Systems, Man & Cybernetics, Part A: Systems and Humans, Special Issue on Collective Intelligence*, Vol. 33, No. 5, pp. 550-559, 2003.

[35] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[36] J. Rosenblatt, "The distributed architecture for mobile navigation," *J. of Experimental & Theoretical AI*, vol. 9, no. 2/3, pp. 339–360, 1997.

[37] M. R. Benjamin, *Interval Programming: A Multi-Objective Optimization Model for Autonomous Vehicle Control*, Ph.D. Thesis, Dep. of Computer Science, Brown University, Providence, RI, 2002.

[38] U. C. G. Commandant, "International Regulations for Prevention of Collisions at Sea," 1972 (1972 COLREGS), US Department of Transportation, US Coast Guard, August 1999, COMMANDANT INSTRUCTION M16672.2D.

[39] Y. Kuwata, M. Wolf, D. Zarzhitsky, and T. Huntsberger, "Safe Maritime Navigation with COLREGS Using Velocity Obstacles," *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'11)*, San Francisco, CA, September 25-30, 2011.

[40] T. Estlin, R. Castano, D. Gaines, B. Bornstein, M. Judd, and R. C. Anderson, "Enabling Autonomous Science for a Mars Rover," *SpaceOps 2008 Conference*, Heidelberg, Germany, May 2008.

[41] T. Estlin, R. Castano, R. C. Anderson, D. Gaines, B. Bornstein, C. de Granville, D. Thompson, M. Burl, and M. Judd, "Automated Targeting for the MER Rovers," *Space Mission Challenges for Information Technology Conference (SMC-IT 2009)*, Pasadena, CA, July 2009.

[42] R. Castano, T. Estlin, R. C. Anderson, D. Gaines, A. Castano, B. Bornstein, C. Chouinard, and M. Judd, "OASIS: Onboard Autonomous Science Investigation System for Opportunistic Rover Science," *Journal of Field Robotics*, *Special Issue on Space Robotics, Part III*, Vol 24, No. 5, pp. 379-397, May 2007.

[43] F. Espinal, T.L. Huntsberger, B. Jawerth, and T. Kubota, "Wavelet-based fractal signature analysis for automatic target recognition," *Optical Engineering, Special Section on Advances in Pattern Recognition*, Vol. 37, pp. 166-174, 1998.

[44] J. Aulinas, M. Carreras, Y. R. Petillot, J. Salvi, X. Llado, R. Garcia, and R. Prados, "Feature extraction for underwater visual SLAM," *IEEE/OES OCEANS 2011*, Santander, Spain, June 2011.

[45] Y. Petillot, F. Maurelli, N. Valeyrie, A. Mallios, P. Ridao, J. Aulinas, and J. Salvi, "Acoustic-based techniques for autonomous underwater vehicle localization," *Journal of Engineering for the Maritime Environment*, Vol. 224, No. 4, pp. 293-307, 2010.