AN OVERVIEW OF AUTONOMOUS UNDERWATER VEHICLE SYSTEMS AND SENSORS AT GEORGIA TECH

Michael E. West,* Thomas R. Collins, John R. Bogle,

Andrew Melim, and Michael Novitzky

I. INTRODUCTION

As the ocean attracts great attention on environmental issues and resources as well as scientific and military tasks, the need for the use of underwater vehicle systems has become more apparent. Underwater vehicles represent a fast-growing research area and promising industry as advanced technologies in various subsystems develop and potential application areas are explored. Great efforts have been made in developing autonomous underwater vehicles (AUVs) to overcome challenging scientific and engineering problems caused by the unstructured and hazardous ocean environment. With the development of new materials, advanced computing and sensory technology, as well as theoretical advancements, research and development activities in the AUV community have increased.

The Georgia Institute of Technology (GIT) is actively involved in three major research efforts: underwater vehicle sensing, underwater communications, and underwater vehicle autonomy including heterogeneous multi-vehicle collaboration. In order to test and experimentally validate the research, GIT has developed a new small man-portable Autonomous Underwater Vehicle called the Yellowfin. This new AUV provides a testbed for real world testing and experimentation of the advanced algorithm development. This paper will show the GIT development in this area.

II. UNDERWATER VEHICLE SENSING

Autonomous underwater vehicles may be outfitted with sensors dedicated to the motion control of the vehicle. Additionally, the AUV may be equipped with specific payloads that are tailored to the mission. GIT has developed capabilities within both domains.

a. Motion Control Sensors

The harsh, forbidding, and dark environments in which AUV must work make it difficult for the vehicles sensory system to determine its position. The success of future AUV's will be the ability to accurately navigate (determine the vehicle pose within geodetic or relative coordinates) and localize (determine the vehicle's specific distance from some fixed point) itself in this underwater domain. The underwater world limits the types of sensors available, as compared to that

* Senior Research Engineer, Electronic Systems Laboratory, Georgia Tech Research Institute, 400 W. 10th St. Rm. 272 Atlanta, GA 30332-0829. E-mail: Mick.West@gtri.gatech.edu

of above the water surface sensors and their vehicles. Electromagnetic energy propagates very little in water and, thus, instruments such as the Global Positioning System (GPS) have limited use in water. However, if truly autonomous systems are to be developed, good navigation sensory information is needed in order to achieve mission goals and provide safe operation.

The current practice for AUV navigation typically involves the use of at least one of the following three types of techniques: Acoustic Transponder Navigation, Dead Reckoning and Inertial Navigation, and Geophysical Navigation. Acoustic navigation usually involves acoustic energy beacons at different baselines: long, short, and ultra-short. Navigation by dead reckoning is defined as determining the location by knowing ones speed and heading. When augmented by reliable inertial measurement devices, dead reckoning becomes inertial navigation. Geophysical navigation uses Earth's geophysical traits such as magnetic field, gravity, or bathymetry. An overview of these systems is discussed in [1 2 3]. The navigational requirements for the sensors involved with the underwater systems is described in [4 5].

b. Payload Sensors

i. Acoustic

Georgia Tech's research in acoustic sensors is primarily conducted within the Mechanical Engineering Acoustics and Vibrations Research Laboratory (AVRL). The research is focused in the area of underwater acoustics and its application to sonar, submarine warfare, physical oceanography, marine biology, and medicine. Their largest project, funded by the Navy, is the development of a radically new concept for hull mounted submarine sonar arrays which was invented by Dr. Peter Rogers and AVRL.

The system, which uses motion sensors mounted on a compliant hull coating rather than the conventional hydrophones, is expected to be used on later versions of the new attack submarines currently being developed by the Navy. Other sonar work includes the development of a high-power, low-frequency acoustic source for the Defense Advanced Research Projects Agency for use as an off-board projector for a shallow water active submarine detection system. In the area of applications to submarine warfare AVRK is studying the effects of coatings on submarine radiated noise and target strength (submarine "stealth").

In the area of marine bioacoustics, studies are being done on how fish use sound and how a fish's auditory system functions. The acoustic startle reflex in fish is also being studied. In response to a sudden loud noise fish exhibit an escape reflex. The reflex, which does not involve the brain, causes the fish to move away from the source of the noise. This group is trying to determine how the fish knows which way is away from the source.

Many of these fundamental research ideas are being translated toward AUV operation. For instance, AVRL researchers have created a sensor that detects the direction from which a sound is coming underwater using optical fibers (fig. 1). Additionally, in collaboration with H. Schmidt (MIT) and W. Kuperman (UCSD), AVRL is looking at multi-bistatic acoustic measurement to improve concurrent detection, classification and localization of man-made targets (fig. 1). A large acoustic water tank is just one of the facilities available to AVRL (fig. 1). This tank is suitable for measurements of radiation, scattering, structural response and transducer characterization.

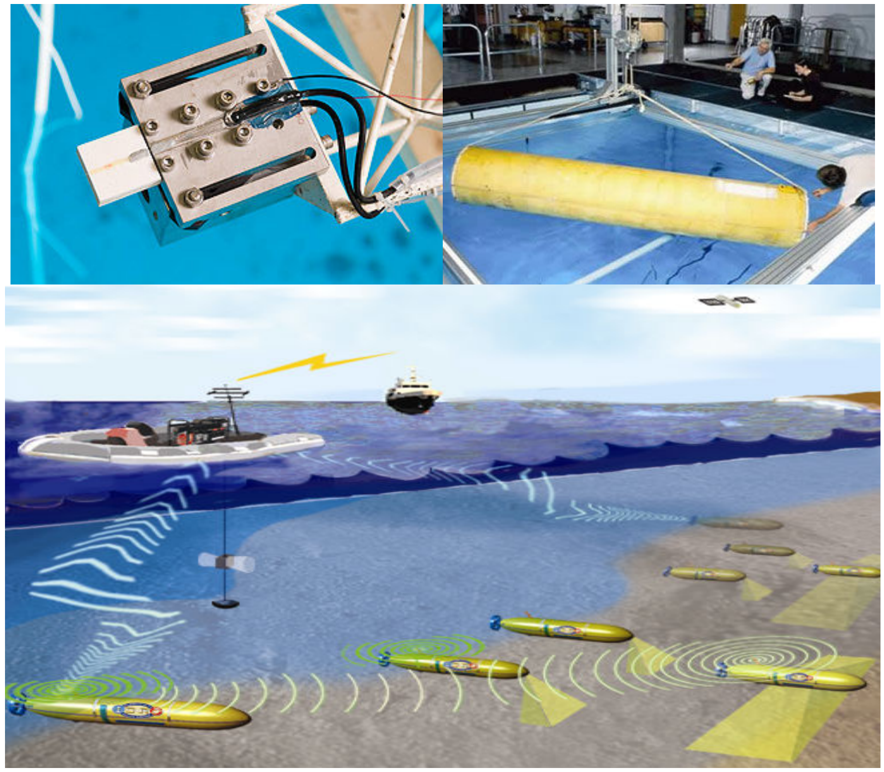

Figure 1: AVRL Sensor Technology (top left); Multi-bistatic acoustic measurement in collaboration with MIT and UCSD (bottom); AVRL Acoustic Test Tank (top right)

ii. Chemical

The Georgia Tech Research Institute has developed and field-tested an optical sensor capable of detecting a wide variety of chemical and biological entities including organic and inorganic compounds (aromatic and chlorinated compounds, ammonia, chlorine, TNT, chemical warfare agents, etc.), proteins and other biomolecules (avidin, biotin, ricin, IgG, IgE, DNA, etc.), real-time monitoring of nucleic acid hybridization, and whole organisms (including *Salmonella*, *Campylobacter* and Anthrax spores, and others). The sensor is fast with high sensitivity, and provides a direct measurement with no additional steps or consumable reagents. Some of the advantages of this sensor system for environmental applications are its low initial and per-sample cost, the ability to detect multiple analytes simultaneously, and the speed and sensitivity of detection. Systems can be designed for mobile, on-site field analysis with instant results or in-situ monitoring with automatic data logging and communication to a base monitoring station. The sensor provides opportunities for advancing research for monitored natural attenuation/enhanced passive remediation (MNA/EPR) scenarios by allowing the real-time measurement of chemical (and microbial) processes in the underwater environment.

Planar waveguides have evanescent fields sensitive to index of refraction changes immediately above the waveguide surface. A chemically-sensitive film on the sensing arm enables a guided beam to detect these changes. Interferometric configuration of the sensing and reference beams results in highly responsive measurements.

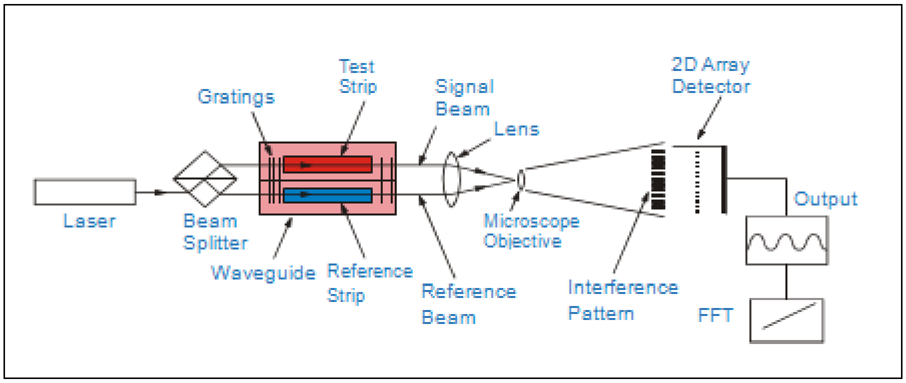

Figure 2: Chemical Sensor Diagram

A different receptor film can be deposited on each channel allowing for multiple analyte sensing. The current software converts the optical response to total phase change which corresponds to the concentration of the analyte present. The sensor's current optical configuration fits in a box approximately 6 x 5 x 3 inches in size containing the subsequent power supplies and electronics. The optical sensor head can be mounted on the box or be tethered to the box with the optical head being placed in the water source. Many of the sensor's other components are inexpensive and off-the-shelf, including a laser diode light source and a CCD.

In September of 2010, the chemical sensor was deployed in the Gulf of Mexico after the Deep Water Horizon Disaster. The goal was to validate the optical waveguide sensor technology for this application. The sensor chemistry is designed to be highly sensitive to aromatic hydrocarbons, and provides users with a direct, real time measurement, unlike traditional testing which requires samples to be collected and returned to a lab for later analysis. Initial testing results indicate the sensor was able to detect oil and surfactant/dispersant breakdown products in select water and soil samples. The sensor is currently being adapted for underwater vehicle use.

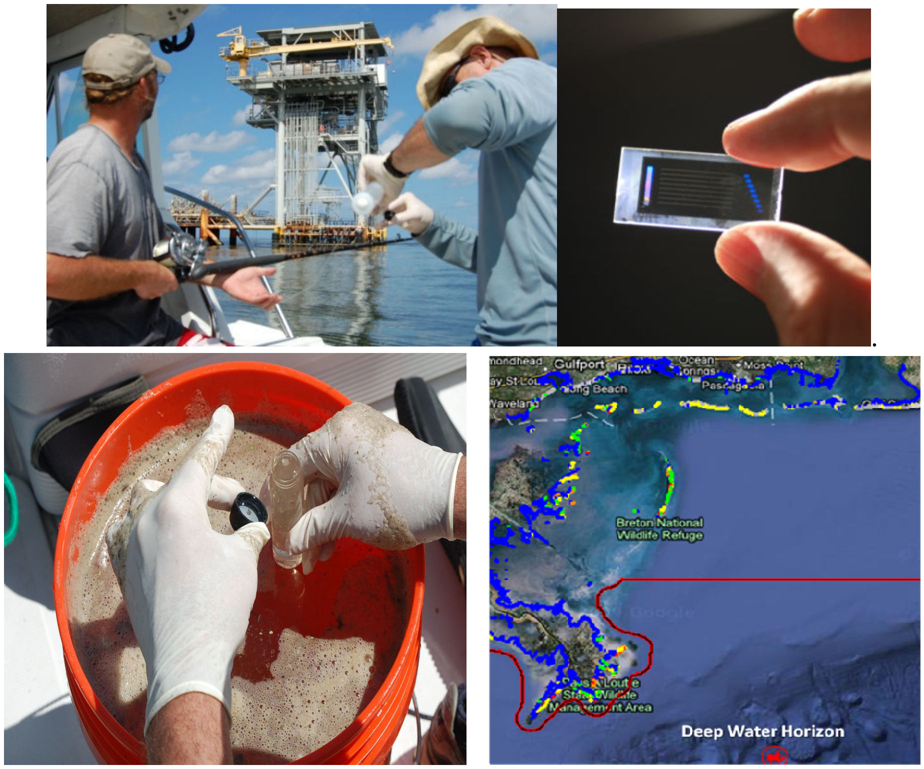

Figure 3: Deep Water Horizon Chemical Sensing

III. UNDERWATER VEHICLE COMMUNICATION

With the cost associated with remote operations, untethered autonomous vehicles is growing but communicating with these vehicles presents formidable challenges. The main approach today for through-water transmission involves acoustics in which transducers convert electrical energy into sound waves. Since the ocean rapidly weakens the acoustic energy as the frequency is increased, relatively low frequencies are desirable for longer-range communications. But at very low frequencies, the required transducer size is impractically large and the data rates are lower. The speed and direction of sound signals vary depending on surface waves, temperature, tides, and currents. GIT is researching both traditional acoustic methods of underwater communication and non-traditional methods.

a. Acoustic Modems

Challenges Facing Underwater Acoustic Communcation

Acoustic communications has been used for many years as the most efficient and reliable method of communicating over significant distances in the ocean. Acoustic modems have been developed to convert digital messages into analog signals that can be transmitted through the water using transducers. There are a variety of issues that make acoustic communication in the ocean a challenging problem.

The first issue is the speed of sound in the ocean is about 1500 m/s as compared to the speed of RF communications in the air at $3x10^8$ m/s. This relatively low speed limits the bandwidth of communications and often encourages engineers to minimize the amount of data to communicate.

The slow speed of sound also increases the impact of Doppler frequency shift due to relative motion of the source and receiver. The received frequency $f_d = f_0$ v / c where f_0 is the transmission frequency, v is the relative velocity, and c is the speed of sound.

A second challenge to acoustic communications in the ocean is the loss incurred as the sound waves propagate through the water. There are three types of loss: spreading, absorption, and scattering. Spreading loss is a geometric loss where signal strength decreases as $1/R^2$ as the waves spread spherically as they propagate away from the transmitting transducer. In shallow water, this loss eventually becomes cylindrical in shape due to the boundaries of the surface and the seafloor so that the loss is proportional to 1/R. The second type of loss is absorption loss which is highly frequency dependent, increasing with increasing frequency. Absorption loss is typically represented as an exponential function $e^{-\alpha(f)R}$ where $\alpha(f)$ is a function of frequency [6].

The third type of loss, scattering loss, can be caused by a number of sources, including bubbles. Bubble clouds can be formed by breaking waves near the surface and are more of an issue in shallow water than in deeper waters. Scattering can also occur due to large schools of fish or other biologics.

Another significant challenge to acoustic communication is the spatial and temporal variation of the speed of sound within the ocean. The sound speed in the ocean is a function of temperature, pressure, and salinity.

Multi-path scattering is a phenomenon in ocean acoustics where sound from a single source can arrive at a receiver through many different paths. Sound waves generated by a source will travel in a direct path from source to receiver but will also reflect off the sea surface and seafloor and arrive at the receiver as well, with some magnitude and phase change.

Given the relatively slow speed of sound, these signals potentially can all arrive before the direct signal has finished arriving, and thus must somehow be distinguished from the direct signal. Figure 4 is a pseudo-color plot of the time difference between the direct path and the first arriving multi-path signal as a function of depth and range. This delay is the maximum message length in time before the multi-path signals arrive and potentially corrupt the direct signal. The data was generated assuming the source and receiver were at the same depth which was chosen to be half the ocean depth. The dashed lines are lines of constant delay and indicate the boundary at which the multi-path signals will be a problem. The top line is a delay of 24 ms which corresponds to a message length of 16 bytes (JAUS message header size) and the second line is a delay of 96 ms which corresponds to a message length of 64 bytes (JAUS *ReportGlobalPose* message size).

The area above the dashed lines potentially will have corrupted signals due to multi-path scattering. The plot shows that multi-path scattering becomes less of a problem as range decreases and as depth increases. It also shows that virtually any shallow water communication will have to deal with the effects of multi-path scattering. There is also clear benefit to reducing the message size.

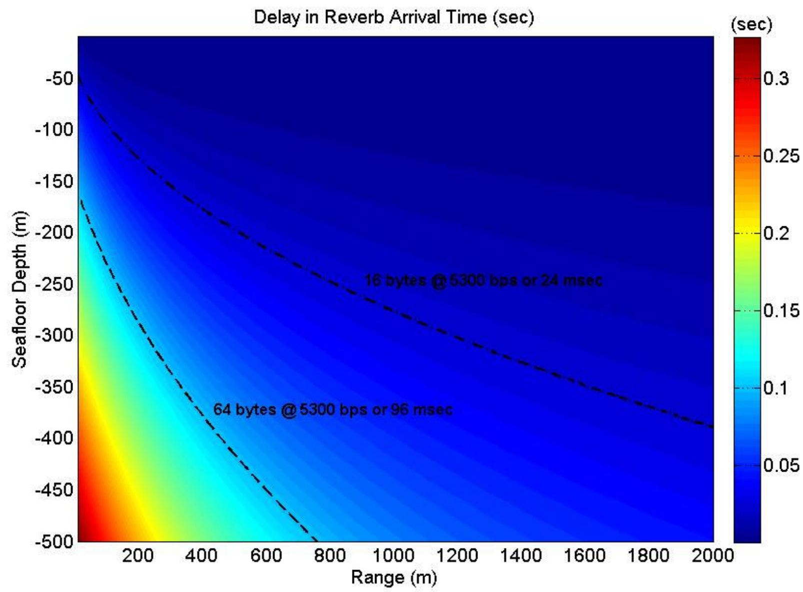

Figure 4: Delay Times Between Direct Path and Surface Reflection as a Function of Ocean Depth and Range

Ambient noise also poses a significant problem to underwater communications. There are numerous sources for noise in the ocean, both man-made and natural. Man-made noise is most prevalent in shallow water environments and includes boat, watercraft noise, and oil rigs. Natural noise sources include biologics such as fish and whales as well as the noise introduced from wave action on the surface. Waves also inhibit shallow water communication by creating bubble clods that can scatter sound waves and from Doppler effects due to the motion of the water column.

WHOI Micro-Modem

Numerous commercial companies have developed acoustic modems for use in ocean communication applications. The Woods Hole Oceanographic Institute (WHOI) has been a leader in research in ocean communications and has developed the WHOI micro-modem. The modem has several modulation schemes including low rate frequency hopping, frequency-shift-keying (FH-FSK) and variable rate phase-coherent keying (PSK).

The WHOI micro-modem is composed of 3 boards: a main board, a power-amplifier, and a floating-point coprocessor board. The WHOI micro-modem was designed to consume low power to support its implementation in underwater vehicles operating on a limited battery supply. On-board computers communicate with the micro-modem using serial communications over a standard RS-232 port. The power amplifier is designed to drive the ceramic transducer, to act as a single channel receiver, and to provide power conditioning to the system. The coprocessor board was designed to perform the computationally complex PSK equalization algorithms[7].

Figure 5: WHOI Acoustic Micro-Modem

The micro-modem implements a robust communications scheme known as frequency-shift keying with frequency hopping that has been shown by WHOI to function very well in shallow water environments. The scheme allows for low-rate communications where multi-path scattering is a problem. The modem can be configured to operate in one of 3 frequency bands: 7.6 to 12 kHz, 12.5 to 17 kHz, and 23 to 30 kHz, each having a 4 kHz bandwidth. The data rate including the error correction is 80 bps[8].

The micro-modem also implements several high-rate phase-shift keying (PSK) algorithms from with data rates of 300 to 5000 bps. The micro-modem can transmit these PSK signals but the co-processor is required to receive and process them.

The micro-modem can send data in one of two packet configurations. The first is known as a mini-packet and contains only 21 bits. This packet type is used for cycle initialization messages which alert the receiver modem that the sender modem is about to send a message to it. Mini-packets can also be used to send very short messages which are useful in navigation schemes. The other packet configuration has variable frame sizes, ranging from 32 to 248 bytes. The modem also has a ping command which can be used to estimate the distance between two nodes. The sender modem sends out a ping message to the receiver modem which responds after a fixed turn-around time. After receiving the response from the receiver modem, the sender modem then computes the one-way travel time from the receiver modem by subtracting the turn-around time.

GIT is investigating the ability to create JAUS messages and to send and receive them through the water column. They have written several software libraries to handle JAUS messages and to be able to code and decode them into the NMEA 0183 protocol used by the WHOI modems. For the purpose of these initial tests, a command line program was written that could be used for both transmitting JAUS messages and also for listening to the serial port for incoming messages. These initial tests were successful in validating that the software was able to form JAUS messages including the global pose data, to pack that message into a NMEA hex binary message, to send it out the modem to the transducer, to receive that message on another modem, to extract the JAUS message, and to parse the global pose data out of the JAUS message.

Figure 6 contains a couple of pictures from the testing. The picture on the left shows L-R the vehicle computer, the vehicle modem stack, a power amp, the deckbox, and the basestation laptop. The picture on the right shows the vehicle transducer and the towfish in the tank.

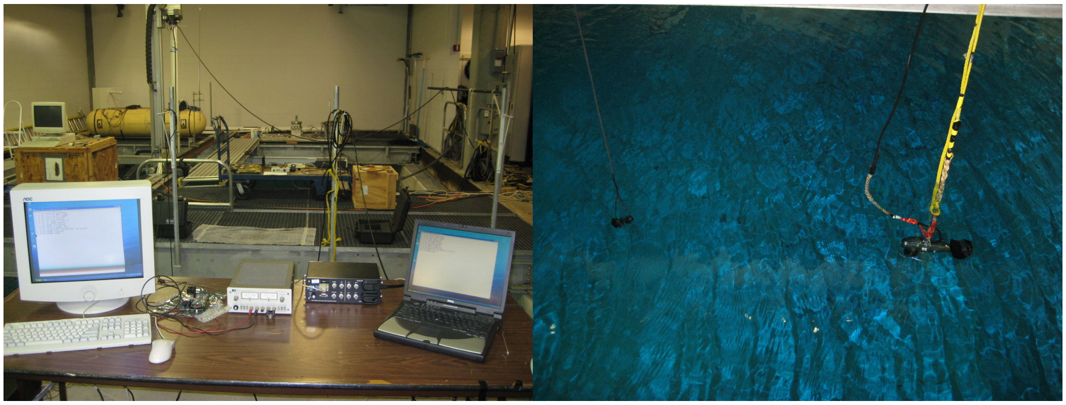

Figure 6: AVRL Acoustic Tank Testing

b. Underwater RF

GIT and partner universities and companies are researching the use of electromagnetic (EM) energy in water. Traditional use of EM energy attenuates tremendously when propagating through water. The underwater RF research investigation solves well-known problems associated with high throughput, robust communication in underwater environments. By robust, it is implied simultaneous increases in enabled underwater communication data rates, range, and reliability. The research borrows concepts from advanced multi-carrier modulation using orthogonal frequency-division multiplexing (OFDM), multiple input multiple output (MIMO) space time codes (e.g. Alamouti diversity), and zero carrier frequency modulation akin to that found in ultra wideband wireless (UWB-OFDM (see [9])). The research relies heavily on Turbo equalization and iterative decoding. OFDM has been demonstrated in power line communications[10], adaptive digital subscriber line (aDSL)[11], and advanced wireless communication applications to enable robust communication in severe frequency selective fading channels with low signal to noise ratios (SNRs). Underwater RF-communication can likewise adopt such an approach. We necessarily point out that alternative underwater communication methods based upon acoustic transmission cannot resolve fundamental issues associated with penetration behind objects, slow propagation, severe shadowing problems, and time-varying channels relative to the symbol period. Furthermore, acoustic systems are severely bandwidth limited. Light based methods are significantly distance-challenged. In this sense, we claim significant advantages for the proposed RF methods, which are based upon EM wave propagation.

We invoke the results of Al-Shamma'a, Shaw, and Saman who experimentally report in [12] [13] that a signal propagating underwater between transmit and receive antennas enclosed in an insulator sheath, after rapid near-field attenuation over 1 meter range due to conduction losses, undergoes relatively minor 1dB to 3dB far-field attenuation out to several hundred meters, yielding a total power loss of about 130 dBm over 1 km range. This is illustrated in Table 1 from the experimental results, demonstrating that a 5 MHz RF signal can be used for underwater communications at a distance of at least 90m. A 5MHz signal would theoretically enable a digital transmission data rate of up to 500kbits per sec.

We target underwater RF signals up to 20MHz, thereby enabling bandwidth expansion relative to acoustics. We do not transmit information sub-carrier signals directly at DC. Rather, we center the OFDM spectral information at DC, but transmit subcarriers at low IF (i.e. zero-carrier signal modulation.). This avoids the well known underwater attenuation problems at higher RF frequencies and elimination of DC signal content improves radio noise figures. Our method utilizes OFDM's high spectral efficiency and the other system components listed in Table 1 to transport information in a low SNR environment. We also apply iterative message passing systems which can be configured either as a Turbo encoder and Turbo decoder in AWGN dominant channels or as a Turbo equalizer using a precoder-channel (see [14]) with (recursive systematic) decoding for fading dominated channels. Very low IF modulation enables fine resolution digital mixers in places of analog synthesizers and mixers. For additional robustness, we instead focus upon fundamental optimizations possible due to the antenna system and incorporate high degrees of digital flexibility through software defined radio principles (see [15]).

The required digital SNR (Eb/No) is based upon the first investigator's Matlab simulation results for OFDM with both Turbo codes [16] and Alamouti space time codes[17], are shown in Fig 8**Error! Reference source not found.**. The Matlab simulations are based upon single 1x1 antenna and MIMO 2x2 antenna configurations using 16-QAM OFDM modulation. The uncoded BER simulation confirms the 20dB gain for Alamouti space time coding for 2x2 transmission compared to 1x1 transmissions. The *coded* BER Matlab simulations shown in Fig. 8 invoked 4-iteration Turbo coding using maximum a posteriori (MAP) decoders. From the 2.0dB Eb/No operation target, the joint 2x2 Alamouti-MIMO, turbo decoder transceiver corresponds to a 10-4 bit error rate (BER). At 2.0 dB Eb/No, the desired 10^{-4} bit error rate corresponds to a packet error rate (PER) target of approximately 1%. In Table 2, we derive a link budget based upon the reported underwater attenuation loss models. For a 2.0 dB sensitivity number, we formulated a preliminary underwater link budget model. The link budget anticipates an advanced antenna design conforming to a reasonably small form factor, a 10 MHz low IF center frequency, and an estimated 3 dB gain for both receiving and transmitting antennas. Such an antenna design is feasible using an array of broadband dipole or spiral antenna elements (the latter more likely because of the bandwidth needed). The antenna design occurs in a joint research effort with advanced antenna researchers at GTRI. The link budget assumes a supported transmit power of 33.5dBm. Table 1 and 2 indicates that we could reasonably anticipate a 400 KHz to 1 MHz communication link data rate for a range of 1000 and 100 meters, respectively. Specific data rates depend on assumptions associated with packet overheads, control channel overheads, training signal overheads, control channel overheads, and code rate assumptions. Much of this overhead issue is ignored in acoustics literature, but is reasonably included in our table. The individual carriers will also be adaptively modulated and coded according to the each sub-carriers channel SNR. Final data rate adjustment will vary by simple graduated amounts from link budget table.

The output of this research will be a real underwater communication prototype jointly integrated with both advanced underwater wireless communications and advanced antenna technologies. This investigation will establish modern underwater (ocean) RF path loss models by gathering empirical data measurements of the underwater channel's impulse response, measuring underwater channel autocorrelation functions, estimating near field attenuation parameters, and far field attenuation in the context of modern RF digital communication systems. Rather than rederive entirely new physical layer protocols, we choose to adopt spectrally efficient OFDM modulation parameters and physical data channels of those specified in 3GPP Long Term Evolution (3GPP-LTE: 4G) [18] for our underwater system. Since the underwater environment is noise-limited, we leverage the MIMO antenna algorithms primarily to increase the reliability via diversity rather than maximizing the capacity via spatial multiplexing The A/D samplers will directly sample the transceiver antenna ports, which enable software defined radio style flexibility. We based the system upon a zero-carrier frequency concept akin to UWB transmission systems found in over-the-air environments. The distinction here is that over-the-air UWB is defined for much broader GHz bandwidth channels compared to the underwater analogy. Since we are communicating RF information through a baseband channel, we populate OFDM modulated information subcarriers on the positive frequency axis only, and take transmit in-phase signal and corresponding OFDM cyclic prefix through the antenna. This has the effect of generating an OFDM signal with a Hermitian image on the negative frequency axis. In addition to measuring underwater signal power versus (dBm) versus range, we will also measure underwater packet error rate versus Eb/No for QPSK and 16QAM signal constellations. This second packet error rate indicator will serve as a measure of the ability of the receiver to demodulate in the presence of additive white Gaussian noise and fading due to the underwater channel.

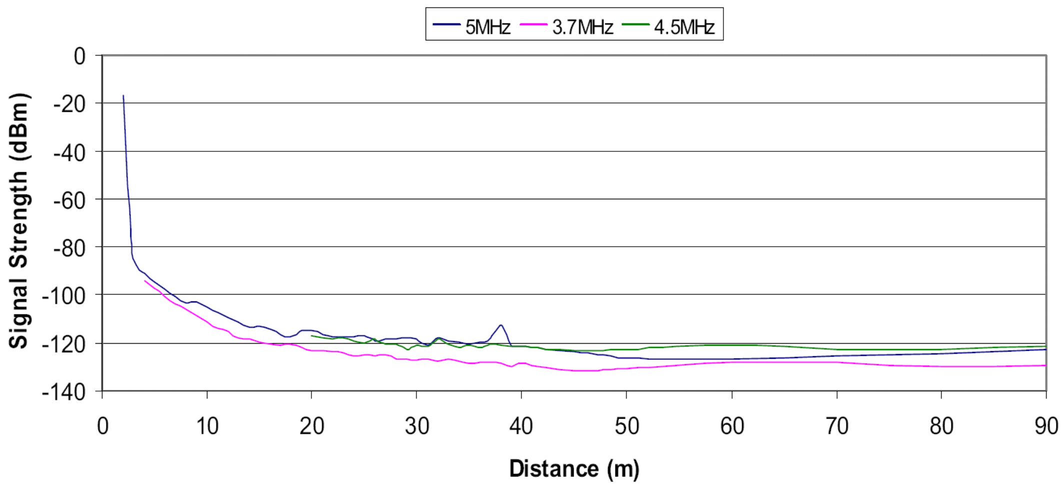

Figure 7: Signal propagation at 3.7 MHz, 4.5 MHz and 5 MHz with an RF power of 5 Watts in a seawater laboratory at the "Liverpool marina" long-range test site. The propagation loss in water does not follow the traditional free space path loss. For an antenna matched to the seawater impedance, the total propagation loss is 130dB out to 90 meters. The near-field optimized propagation loss (at 3 meters for example) is approximately 60dB. Also, the antenna is not placed in direct contact with the seawater.

10MHz Underwater Link Budget Parameters: 100Meter		
	Parameter	Value
Link Parameters	Ambient System Temp Ts	290.00 K
	Tx power: Pt	33.50 dBm
	Antenna Gain Tx: Gt	3.00 dB
	Antenna Gain Rx: Gr	3.00 dB
	Distance: d	100.00 meters
	frequency: v	1.00E+07 Hz
	Far Field Attenuation: Lo_p	8.00 dB
	Near Field Attenuation: Lo_A	60.00 dB
	Far Field Diffraction Losses: Lo_D	68.00 dB
	Alamouti+Turbo Required: Eb/No	2.00 dB
	Noise Figure: NF	5.00 dB
	Bit rate: R	1.02E+07 bits/sec
Constants	constant: PI	3.1415
	speed of light: c	3.00E+08 m/sec
	Botlzmann's k	1.38E-23 W/K-Hz
	Thermal noise: kTo	-173.98 dBm/Hz
	Traditional Over Air Path Loss*	32.44 dB
	Estimated Underwater Path Loss	136.00
	Receiver Antenna Power	-96.50 dBm
	Rx Margin	0.37 dB
	Actual Receiver Eb/No	2.37 dB
	Required Min Rx Power for Demod.	-96.87 dBm

10MHz Underwater Link Budget Parameters: 1000 Meter		
	Parameter	Value
Link Parameters	Ambient System Temp Ts	290.00 K
	Tx power: Pt	33.50 dBm
	Antenna Gain Tx: Gt	3.00 dB
	Antenna Gain Rx: Gr	3.00 dB
	Distance: d	1000.00 meters
	frequency: v	1.00E+07 Hz
	Far Field Attenuation: Lo_p	8.00 dB
	Near Field Attenuation: Lo_A	60.00 dB
	Far Field Diffraction Losses: Lo_D	72.00 dB
	Alamouti+Turbo Required: Eb/No	2.00 dB
	Noise Figure: NF	5.00 dB
	Bit rate: R	4.10E+06 bits/sec
Constants	constant: PI	3.1415
	speed of light: c	3.00E+08 m/sec
	Botlzmann's k	1.38E-23 W/K-Hz
	Thermal noise: kTo	-173.98 dBm/Hz
	Traditional Over Air Path Loss*	52.44 dB
	Estimated Underwater Path Loss	140.00
	Receiver Antenna Power	-100.50 dBm
	Rx Margin	0.35 dB
	Actual Receiver Eb/No	2.35 dB
	Required Min Rx Power for Demod.	-100.85 dBm

Table 1: Underwater link budgets derived from models published in (a) 100 meter (b) 1000 meter

System Component	Qualitative Optimization Enabler
Small Form Factor Antenna	size, underwater mobile integration
High Gain Antenna	antenna array on buoy node and underwater vehicle
Antenna Array	MIMO space time coding plus beamforming
Precoding	beamforming; turbo equalization (see [17])
Turbo Encoding	high coding cain
RF-OFDM	flat fading, high spectral efficiency, 15KHz subcarrier spacing
Zero Carrier RF (1-20MHz IF)	software defined radio w/high resolution A/Ds, digital mixers

Table 2. Underwater RF data rate enablers

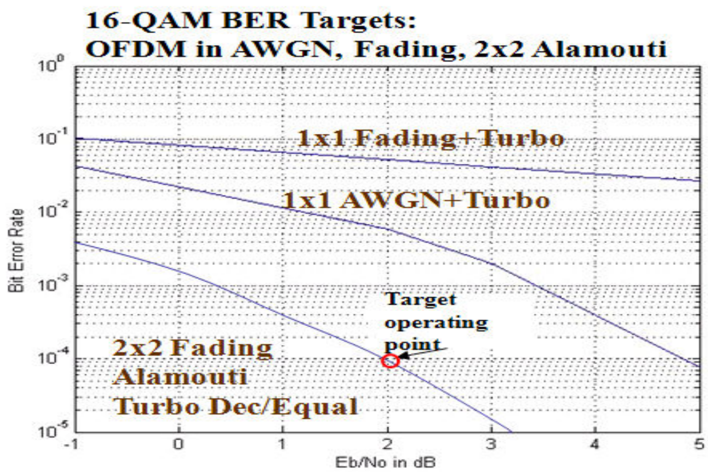

Figure 8: Matlab BER Analysis

The list of possible uses with AUVs is numerous[19]:

- Real-time control of AUVs from shore, submarines and surface vessels
- Wireless through-hull transfer of power and data
- High-speed transfer of data between AUVs and surface vessels
- Real-time transfer of sensor data from AUVs when submerged
- Communications between UUVs and subsea sensors
- AUV distributed navigation systems for shallow harbors and ports
- AUV docking systems
- Subsea navigation beacons; asset location, asset protection
- Subsea networks
- Data transmission from underwater sensors to surface or shore without surface repeaters
- Harvest data from submerged sensors via Unmanned Aerial Vehicles
- Heterogeneous communications; AUV to AUV, submarine to AUV, AUV to Unmanned Surface Vehicle (USV), AUV to Unmanned Aerial Vehicles
- Underwater navigation
- Underwater sensing

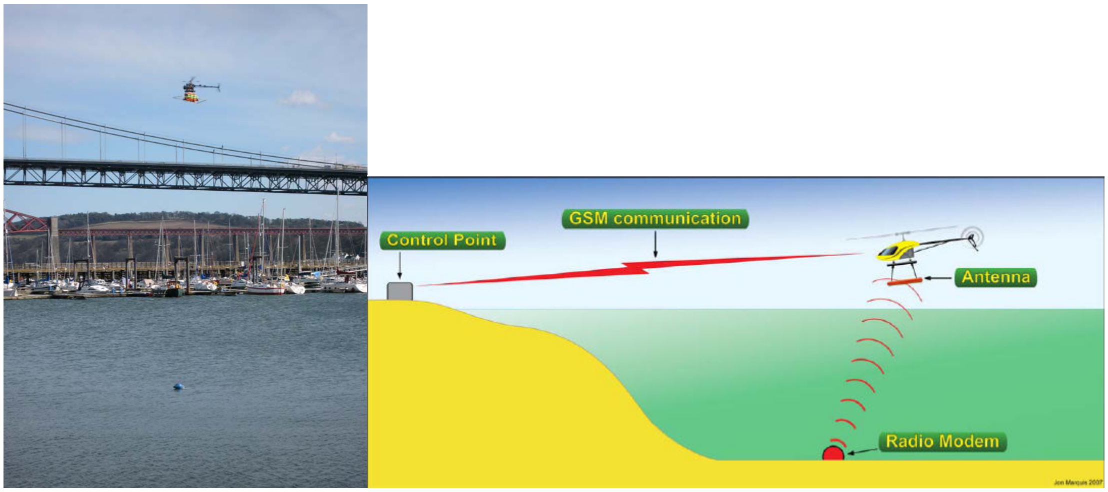

Figure 9: Wireless Fibre Systems demonstrates two way communications from underwater to an unmanned aerial vehicle.[20]

IV. AUTONOMY AND COOPERATIVE VEHICLES

Multi-AUV design allows for a much wider swath of the ocean to be observed and monitored. While, collaborative operations allows multiple aspects of a mission to be tackles with distributed systems. Both oceanographic and military missions are aided tremendously by the use of such an AUV network. The current paradigm for the use of AUVs is a platform centric sensing system. Currently, the US Navy's planned deployment of AUVs emphasizes single-vehicle operation through at least 2015 with uncertain capabilities for cooperating vehicles beyond that. However, to reduce the overall time and cost of acquiring data over large unstructured area, multiple vehicles must be used.

a. When Should We Cooperate? [21]

Biologists have observed switching between individual and cooperative behaviors in nature. For example, it is conjectured that fish in a group communicate and exchange information with others only when they are not confident with the information gathered individually [22]. Some recent work on exploration and exploitation [23] [24] and task allocation [25] [26] are also related to switching strategies. A cooperative network of sensing agents is expected to perform better than a single agent or a number of sensors that are fixed when accuracy and adaptiveness are concerned. However, a single agent has advantages when mobility and the cost of the system are concerned. In order to make the best use of resources, the exploring behavior for each sensing agent does not have to be fixed.

We consider the mission of searching for a local minimum of an unknown scalar field with multiple sensing agents. The field is assumed to be corrupted by time-varying non- Gaussian noises. This is a challenging task, but it is closer to real world applications than the Gaussian noise assumption in most literature. A switching strategy for the networked sensing agents is developed so that each agent makes decision at every time step about whether to perform individual exploration or cooperative exploration. Inspired by the Razumikhin theorem [27] [28], which is widely used in justifying the stability of time-delay systems, we introduce sufficient conditions for the sensing agents to examine whether they will converge to a field minimum by conducting individual exploration. If the sufficient conditions are not violated, which means each agent can find the local minimum by itself, then an H_∞ filter is running and producing filtered measurements of the field, which are used to give estimates of the field gradients. The estimated gradients are utilized so that by following the opposite directions of the gradients, the agents may find a local minimum of the field [29]. If the switching conditions of convergence are violated, the agents switch to cooperative exploration to obtain more accurate estimates of the gradients. A cooperative H_∞ filter is constructed to give estimated field values and gradients at the formation center formed by the agents. During the cooperative exploration phase, if the agents detect that the signal-to-noise (SNR) ratio of the field increases by a sufficient amount, they switch back to individual exploration with the hope that convergence can be achieved. Our switching strategy in this paper is based on rigorous convergence analysis.

The H_∞ filter differs from the Kalman filter in that it does not require the knowledge of the noise properties except that the noises are assumed to have bounded power. Note that the Kalman filter assumes the noises to be Gaussian [30] [31] [32]. Therefore, the H_∞ filter is robust to time-varying non-Gaussian noises. An important constraint of the H_∞ filter is that the existence of the filter requires the fulfillment of a set of feasibility conditions, which further posts constraints on the exploration behaviors for the individual agents and the cooperative agent formation. Convergence analysis of H_∞ filters has been performed in [33] [34]. Based on these work, we develop sufficient conditions for the cooperative H_∞ filter to admit feasible solutions and convergence.

In order to test and demonstrate the switching strategy in real-world environment, we develop a multi-robot test-bed and implement the switching strategy on it. By conducting several trials, we compare the behavior of a group of robots associated with different parameters for exploration. Especially, we observe the effect of memory lengths on the switching behavior, which agrees with theoretical predictions. We justify the effects of different formation sizes and noise attenuation levels to the performance of the cooperative H_∞ filter that are utilized in the cooperative exploration phase to provide estimates of the field value and the gradient.

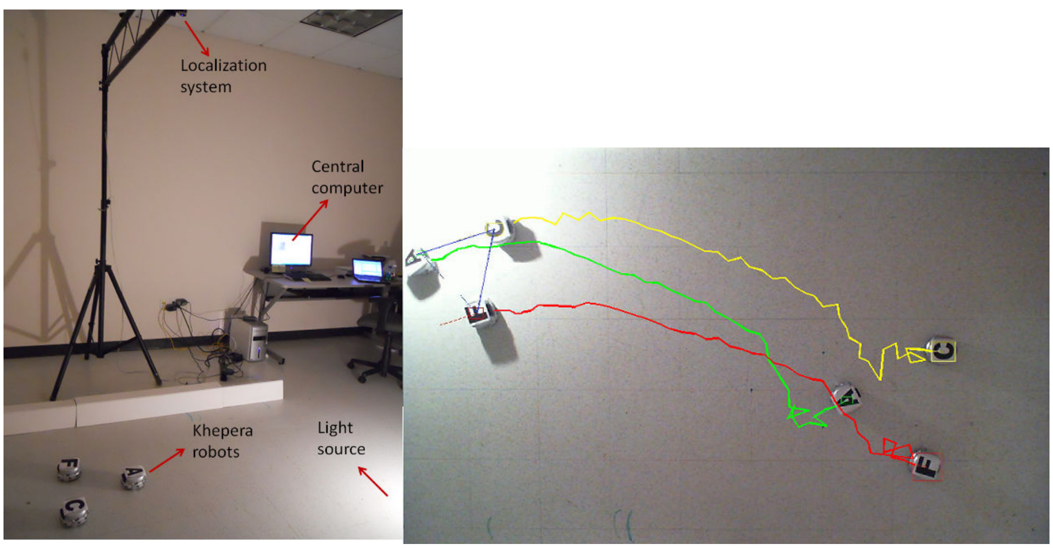

Figure 10: Experimental Validation of Independent vs. Cooperative Robots[35]

b. Behavioral Autonomy

i. Architecture

The control architecture, or high level controller, for a genuinely autonomous underwater vehicle should include a hybrid architecture as shown in figure 11.

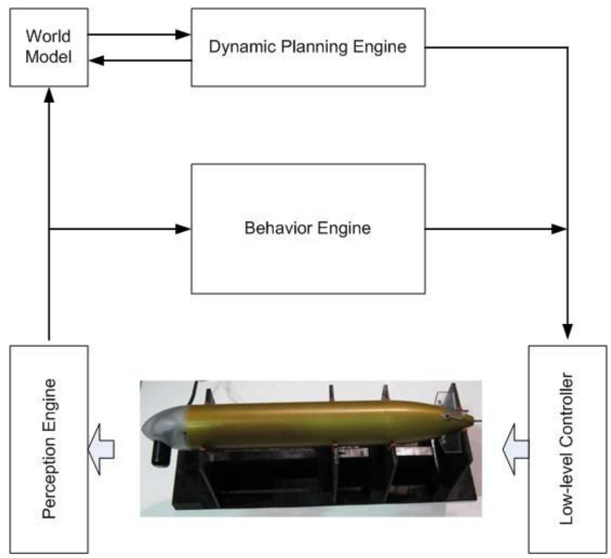

Figure 11: AUV Hybrid Behavioral Control

Broadly speaking, architectures for vehicle behavioral autonomy have included

- Hierarchical architectures, ranging from traditional nested control loops to the earlier forms of the Albus RCS architecture [36]
- Reactive architectures, exemplified by approaches as different as Brook's subsumption [37] and Arkin's AuRA [38].
- Hybrid Architectures, exhibiting some aspects of hierarchy for deliberative planning and control, along with a reactive layer for effective and robust real-time performance.

In truth, all but the purest of reactive architectures have always incorporated some deliberative aspects, and thus have always been hybrid in some sense. And as time progresses, even the most steadfast adherents of hierarchy have incorporated more reactive components [39]. Thus, it is generally possible to map most architectures onto the more general model of a deliberative component that hierarchically "instructs" a reactive component. There is no limit to the complexity of either component. For example, learning and adaptation can occur within both the deliberative components and the reactive components.

Deliberative components are based on planning and usually also on a world model. They allow reasoning and making predictions about the environment. Data flows from sensors to the world model (bottom-up), which is used to plan new actions to be undertaken by the actuators (top-down). When dealing with a highly dynamic environment, the delay in the response time is the main drawback. Reactive components, on the other hand, tend to operate in parallel or with efficient means of inhibiting one another through arbitration. Data also flows from sensors to the reactive components, but typically with little interpretation, and only with the most primitive of world models being formed.

Georgia Tech's research on robotic architectures pioneered the notion of deliberative/reactive systems through the design of the *Autonomous Robot Architecture* (*AuRA*), which is supported by the mission specification system *MissionLab*. *AuRA* was developed as a hybrid approach to robotic navigation. The highest level of *AuRA* is a *mission planner* concerned with establishing high-level goals for the unmanned system and the constraints within which it must operate. In earlier *AuRA*-based systems, the mission planner has acted primarily as an interface to a human commander, but more recently, automatic learning takes place at this level and at other levels in the architecture. The *spatial reasoner* uses cartographic knowledge stored in long-term memory to construct a sequence of navigational segments which the unmanned vehicle must execute in order to complete its mission. The *plan sequencer* translates each path leg generated by the spatial reasoner into a set of motor behaviors for execution. Finally, the *schema controller* is responsible for sending motor commands to the robot hardware (usually to an embedded controller).

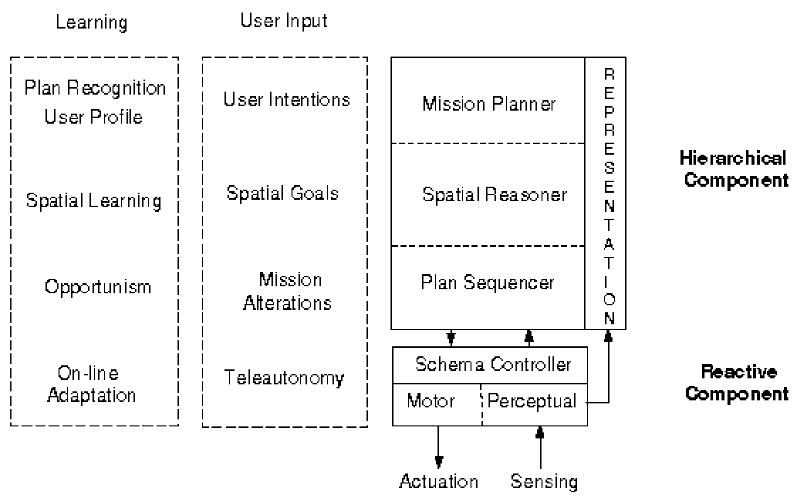

Figure 12: AuRA Robotic Architecture

As AuRA has evolved over the last two decades, other hybrid architectures have developed simultaneously. Most recently, the term "architecture" has become blurred with "operating system," "API," or "middleware," because software tools for particular architectural approaches have become extensive. In some cases, the mere existence of command/response protocols and supporting APIs for messaging has been defined as an "architecture." This is generally a positive development, as it establishes a common trend toward hybrid approaches, with supporting software and uniform standards for messaging. The *MissionLab* system has embraced this by supporting "late binding" of robot mission executables to the user's choice of robotic platforms. AuRA-based robots are still fully supported, but it is also possible to create executables that issue, for example, STANAG 4586 messages or commands specific to an ActivMedia Pioneer robot.

On a wider front, the open-source community is actively developing two cross-platform software projects that directly support code development for hybrid robotic architectures. One of these is ROS [*]. It is called a "Robotic Operating System" because it provides services such as

---

[*] www.ros.org

hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It evolved from an earlier cross-platform project, Player/Stage, which continues to be supported as a means of interfacing Player-compatible networked robots. Several dozen robotic platforms are currently compatible with ROS, but with minimal AUV applications thus far. Extending ROS for AUVs is a primary thrust of work to be described here.

A similar open-source effort, but targeted more directly at AUVs, is MOOS (Mission-Oriented Operating Suite). MOOS provides similar middleware capability as ROS, but is more strongly oriented toward an onboard publish/subscribe architecture implemented with its shared database *MOOSDB*. This is particularly suitable for the concurrent execution of control processes and sensory "instruments," taking advantage of the *interval programming* methods inherent within MOOS. While ROS supports a similar service-oriented architecture (SOA) through its *roscore* process, it is more general and provides services to multiple platforms with less rigorous scheduling and arbitration. A sibling project to MOOS called ivpHelm provides an ensemble of behaviors relevant to AUVs and USVs.

Recognizing the wealth of open-source software in both the ROS and MOOS communities, Georgia Tech is developing a strategy that combines the strengths of ROS, MOOS/ivpHelm, *MissionLab*, and other ongoing software projects. As shown in Figure 13, multiple vehicles can be configured and monitored from a workstation that is "fluent" with vehicle protocols and architectural capabilities. (Although a single centralized node is shown for simplicity, we do not preclude distributing this role across multiple control stations or even across some of the platforms themselves). While this figure illustrates the operational "run-time" mode of the system, this would typically be preceded by a configuration phase where the control station downloads behavioral specifications to each platform. These can take the form of executable programs or standard behavioral formats, such as the ivpHelm "bhv" file format. *MissionLab* is particularly suitable to this. Its native CDL (configuration description language) code generator can be replaced with other architectural bindings at run time, including bhv formats (already demonstrated with prototypes at Georgia Tech).

Some platforms may require no preconfiguration, and may in fact have no capability for onboard decision planning or the sequencing of behaviors. This is especially likely to be true for currently deployed military platforms conforming to JAUS or STANAG 4586 protocols. In such instances, the control station assumes the role of issuing messages at the appropriate time. But even fully autonomous platforms may allow (or require) redirection, thus the various links (ROS, JAUS, STANAG 4586) are shown for each platform in Figure 13. We also anticipate the introduction of more vehicle autonomy capabilities to JAUS and STANAG 4586 standards.

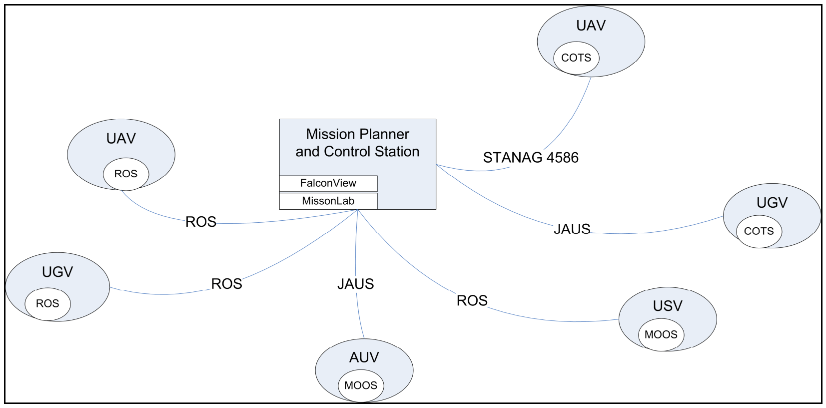

Figure 13: Open-source and vendor-specific protocols and architectures can co-exist in an effective heterogenous system architecture.

However, since most of the vehicles in Figure 13 could have been preconfigured to operate autonomously, the assumption is that the illustrated linkages are primarily for OPTIONAL monitoring and redirection. Additional discussion of some of the key components in the figure (FalconView and MissionLab's cfgEdit) is provided later, under the Software System section of the Yellowfin vehicle overview.

Within this multi-vehicle architecture, the *MOOSDB* mechanism for interprocess communication is replicated locally within MOOS nodes and is needed nowhere else. The similar ROS *roscore* process could reside either within each node or within the control station, if needed for services such as collaborative tasking and redirection. Although the publicly-distributed *roscore* currently is allowed to exist in only one location on a vehicle network, ongoing collaboration on the ARL MAST project has demonstrated a *roscore* that can be distributed.

ii. Dynamic Planning Engine [40]

Two challenges in fielding teams of mobile robots lie in determining the steps a robot should follow when executing a task (mission specification) and determining which robot should execute a given task (task allocation). Mission specification can be a time-consuming and complex process for users experienced in mission design, let alone for those not intimately familiar with the domain. In the case of multi-robot missions, the difficulty of mission specification is compounded as the process must be repeated multiple times, increasing the possibility for error in the design and increasing design time. For heterogeneous teams (i.e., robots that have significantly differing capabilities such as different sensor packages and/or different terrain capabilities such as aerial versus ground versus undersea unmanned vehicles), allocating the available robots to the appropriate tasks places an additional burden upon the operator. Task allocation becomes increasingly difficult as the number of robots increases or if the capabilities of the robots are not known accurately by the operator.

Mission Specification

Mission specification, as described in this work, is the process in which step-by-step instructions are generated to guide one or more robots to accomplish a set of tasks. An example of a mission generated for a multi-robot team may be reconnaissance of an unknown area. Such a mission is composed of many separate tasks. In this case, tasks may include patrolling a particular area or tracking targets discovered in the area. These tasks can be further broken down into the individual actions or behaviors that must be undertaken to achieve them. In order to fully specify a mission, therefore, one must detail: (1) the tasks to undertake; (2) the way to perform the tasks; and (3) any temporal constraints that may exist between the tasks or behaviors (i.e., the requirement of finding a target before tracking it). One method in which mission specification has been conducted in the past has been through traditional programming, where an expert explicitly programs the robots to perform the tasks in the proper order using languages such as C or LISP. Many systems attempt to automate the mission specification process through the use of planners [41 42]. These approaches, however, often result in mission plans that are difficult for a human operator to customize, reuse, or inspect. An alternative to these approaches is to present the user with a graphical mission specification interface with which reusable components at the action, task, or even mission level can be combined or modified to create the desired mission. MissionLab [43] is one such toolset used for multi-robot mission specification.

Multi-robot Task Allocation

Once a mission has been decomposed into its requisite tasks, the question of which robot should be responsible for executing each particular task still remains. Many techniques for multi-robot task allocation (MRTA) have been examined. Prior work includes approaches utilizing teammate modeling [44], distributed constraint matching , and others [45]. Gerkey and Matari´c [46] provide a thorough review of several MRTA frameworks. MRTA is often framed as an optimization problem in which some performance metric is minimized or maximized given a set of tasks and a set of available robots.

For many applications, however, the ordering of tasks, the availability of robots, and the suitability of a robot for a particular task are not known. Because of this, recent research has investigated various market-based approaches to team allocation with significant success [47 48].
In market-based task allocation, a series of task proposals corresponding to the tasks available to be allocated are generated and submitted to all available robots. Each proposal contains information pertaining to the type of task to be executed. The available team members accept or reject each proposal. If the proposal is accepted by a robot, it also submits a bid for that proposal. This bid provides a self-estimate of how suitable that robot is to perform the task described in the proposal. After a specified period of time, the task proposer evaluates all the submitted bids, and awards one or more of the bidders the contract. Once the contract has been awarded, the winning robots are now responsible for that task. This process is continued until all tasks have been contracted out. Market-based MRTA provides several advantages over other mechanisms for task allocation. For instance, market-based MRTA does not rely on a priori knowledge of tasks, task ordering, or robot availability to allocate robots. Because of this, it is highly robust to unanticipated failures and uncertainty in the environment. While the market-based MRTA does not guarantee optimal allocation of robots to tasks when task ordering is known a priori, it does provide provable bounds on this optimal allocation.

CBR Mission Specification and CNP Task Allocation

This work addresses the complexity of these two tasks by examining multiple ways by which mission specification in the form of a case-based reasoning (CBR) planner can be integrated with contract net protocol (CNP) based task allocation. We examine two alternative designs in which to combine these two tools within the MissionLab mission specification toolkit [49] . In the first design discussed, mission specification and task allocation are linked together during the process of mission generation. In the second design, mission specification instead assists creating structures that support task allocation while the mission is executing. Both approaches afford different strengths and weaknesses in terms of usability, design, and scalability.

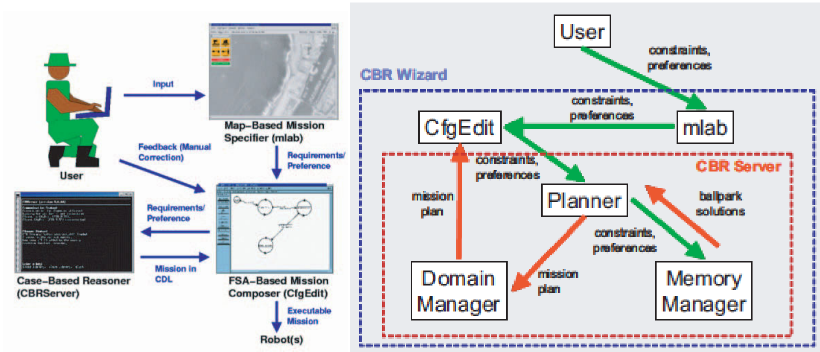

Figure 14: Overview of mission specification using the CBR Wizard. A user specifies the desired tasks using the map-based interface of MissionLab. CBRServer retrieves a mission based on the user's input. The mission is presented for user inspection, modification, and execution.

An experiment directed to a target interception scenario. In a target interception mission, a robot must overtake a mobile surface vessel launched from a nearby pier (figure 15). The vessel interception experiments used a highly heterogeneous team of autonomous underwater vehicles (AUV), unmanned aerial vehicles (UAV), and unmanned surface vehicles (USV). Each vehicle type had differing velocity and fuel consumption characteristics.

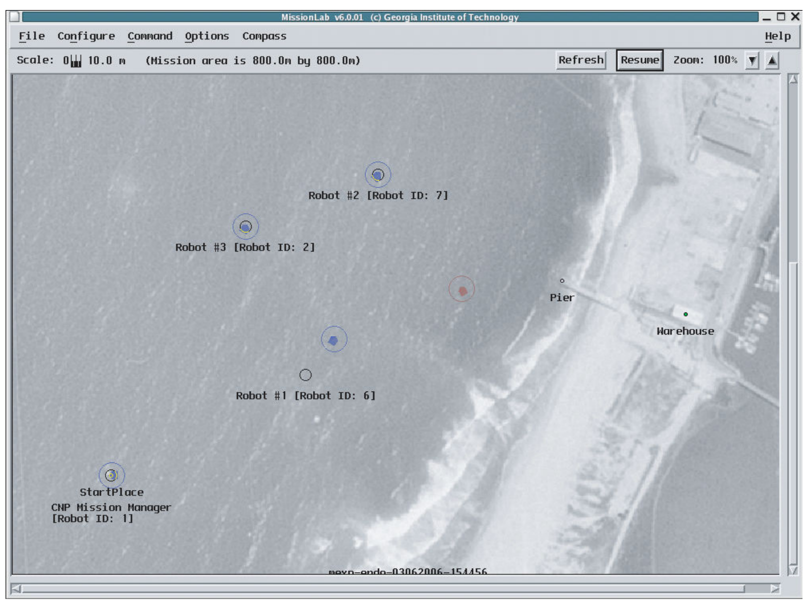

Figure 15: Mission area used for interception scenario. Enemy vessel originates from the location marked pier.

The team deployed in the mission area is responsible for tracking this object by following the transportation vehicle in a stealthy manner when possible. A robot that is stealthy in this scenario, means that the robot has low visibility (e.g. a UUV). The component tasks within this scenario are a series of tracking tasks based on the current mobility type (land, sea, or air) of the target, the target's current position, and stealth requirements. A team of four robots was used in this scenario: a UUV, UGV, UAV, and USV. Each vehicle has differing velocities, mobility, and stealth characteristics.

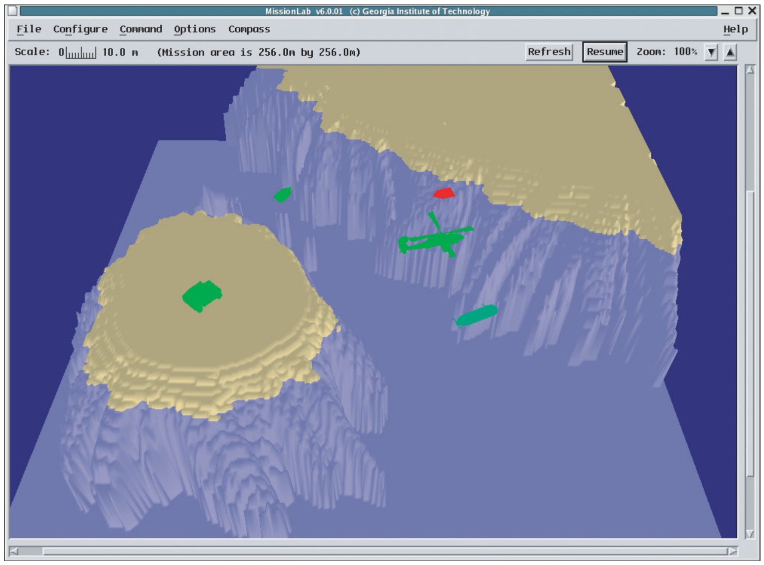

Figure 16: Mission area used for target tracking scenario. Target begins on the mainland depicted on the top left of the mission area, travels to the island, and then flies from the mission area.

Results can be seen in[50]. This research was applied to the Lockheed ICARUS project Figure 17.

Figure 17: Lockheed ICARUS Demonstration

II. OVERVIEW OF THE YELLOWFIN VEHICLE

The Yellowfin is a single man portable micro-UUV which weighs less than 17lbs. The mission for the Yellowfin is to conduct autonomous homogeneous collaborative operations. The system uses multi-agent collaborative technologies to autonomously control and coordinate multiple UUVs. The behavior modules were developed and implemented using the open source Mission Oriented Operating Suite (MOOS) architecture along with the Interval Programming (IvP) Helm software module. The MOOS architecture was coupled with MissionLab open source software tools developed by Georgia Tech. This suite of software was then used to develop and test behaviors for both single UUV autonomy and a team of autonomous collaborating UUVs. This

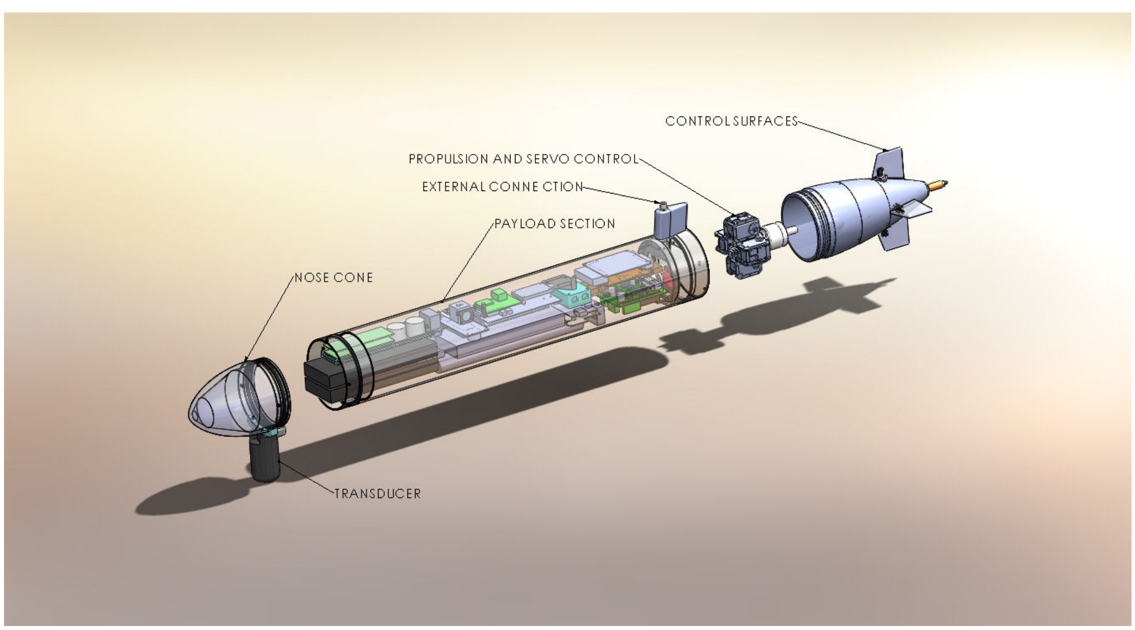

Figure 18: Yellowfin CAD Drawing

coupling of software allowed for the development of highly capable UUVs using behavior-based autonomy. By using open source and standards-based development in the design of Yellowfin, we have attained a modular and adaptable suite of platforms that support flexibility in mission execution. The design also decouples the vehicle autonomy logic from the mechanical control of the vehicle hardware, so that upgrades and adaptations are readily incorporated. Yellowfin is also designed to seamlessly incorporate different sensors, so that collaborative behaviors will enable coordination between the different vehicle sensors. The result is a standards-based system capable of supporting intelligent autonomy, from perception to situational understanding to automated responses. This section will show the initial design of Yellowfin.

a. Yellowfin Vehicle Design

A number of design challenges exist for the development of man portable micro-UUVs. Because space is at a premium all of the electronics for power, communications, sensors, computation, and actuation along with their packaging must be carefully considered. A constant tradeoff between cost and development time was mitigated with the use of commercially available off the shelf components when possible.

b. Design Approach

The intended design of Yellowfin is to supply a high baseline of integrated functionality while still being adaptable to a wide range of mission-specific requirements. This purpose has lead to a focus on designing Yellowfin to be modular and has also dictated some of the high-level design specifications. In order to keep cost and development time as low as possible while maintaining the quantity and quality of vehicle features along with system interoperability, the design of Yellowfin adopted the use of COTS (Commercial, off-the-shelf) components, open-source software, and industry standards whenever feasible. The vehicle itself was designed through an iterative process where one prototype was designed, built, and tested followed by several additional vehicles designed using the lessons learned from the first. The software design has also progressed in stages with the results of the implementation motivating additional cycles of design work.

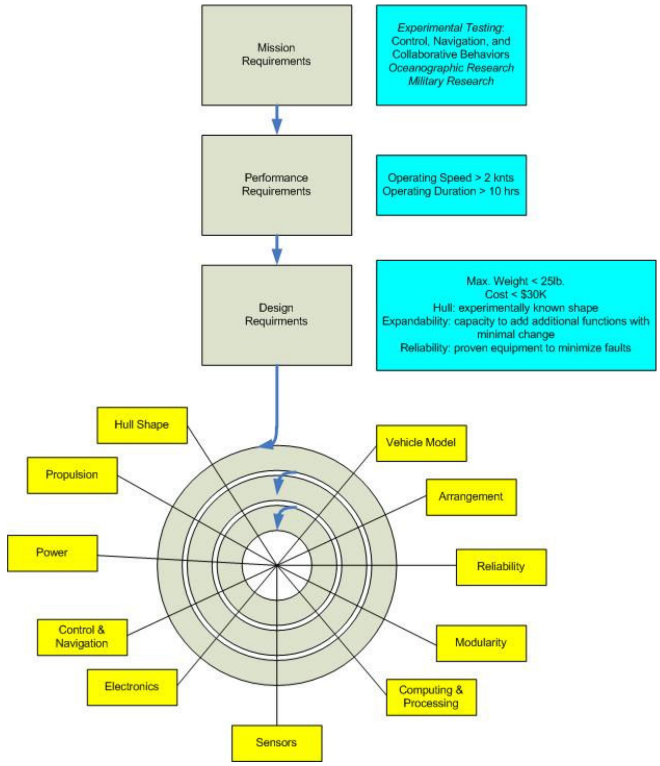

Figure 19: Yellowfin CAD Drawing

c. Design Goals

The requirements can be categorized into two primary components: [51]

Mission requirements

These are specific to the mission Yellowfin will perform. While Yellowfin should be capable of performing a diverse category of missions, there is still a core set of features, which are common requirements to most UUV missions.

- Control, navigation, and collaborative behaviors
- Validation through real and simulated testing
- Suitable for multiple applications, including oceanographic and military research

Vehicle requirements

These are specific to the Yellowfin vehicle. Particular values for performance specifications were determined by looking at the requirements for some specific applications and also chosen to meet or exceed the capabilities of platforms similar to Yellowfin.

- Operating speed > 2 knots
- Operating duration > 10 hours
- Max weight < 17 lb
- Cost < $30K

- Able to accommodate various sensor and communication payloads

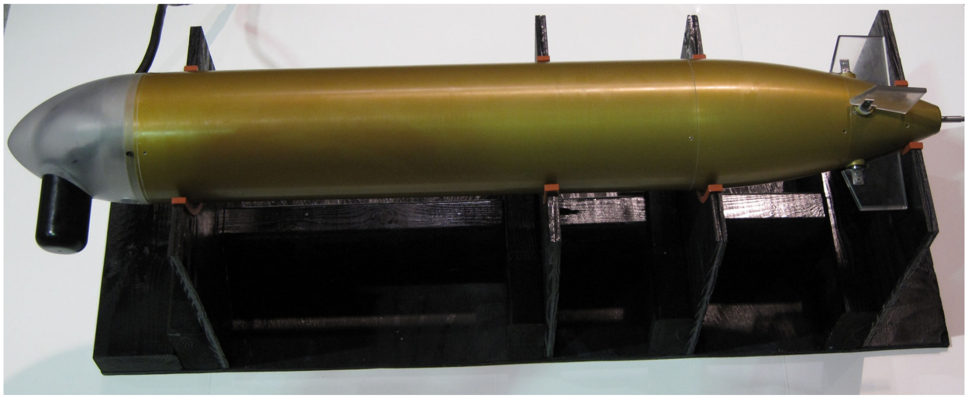

Figure 20: Fully Assembled Yellowfin

Dimensions/Constants			
Parameter	Value	Unit	Description
L	889	mm	Total length
D	123.8	mm	Diameter
W	7.711	kg	Weight
U	2.572	m/s	Design velocity
U_{max}	5.144	m/s	Max velocity
a	137.2	mm	Nose length
b	502.9	mm	Body length
c	241.3	mm	Tail length
S_{fin}	2149	mm^2	Fin area
$X_{finpost}$	21.4	mm	Fin moment arm
γ	10	°	Max fin angle
c_d	0.14	--	Drag Coefficent
ρ	1000	kg/m^3	Density water at 5C
A_f	0.0120	m^2	Frontal Area

Drag Force			
Fd	22.31	N	Drag Force @ 10knots
	5.02	lbf	
	5.58	N	Drag Force @ 5knots
	1.25	lbf	

Figure 21: Yellowfin Physical Dimensions

d. Mechanical System

Yellowfin's hull was chosen to take on a Myring shape, which is often used by UUVs[52]. Since leaks are a serious risk for UUVs, care was taken to minimize the number of hull seals. Each of the separations of the hull are sealed using two o-rings and the propeller uses a drive shaft with a stainless steel spring inside a rubber sleeve. A pressure fitting is used to equalize the pressure when sealing and unsealing the hull and can also be used to test for leaks by pressurizing the hull. The nose cone was designed to be replaceable to support various payloads such as sensors and communication hardware. This is possible by having the nose cone sealed or flooded. The hull also has a removable mast with electrical connectors. These connectors can be used for mounting antennas, which need to be held out of water, such as the GPS antenna.

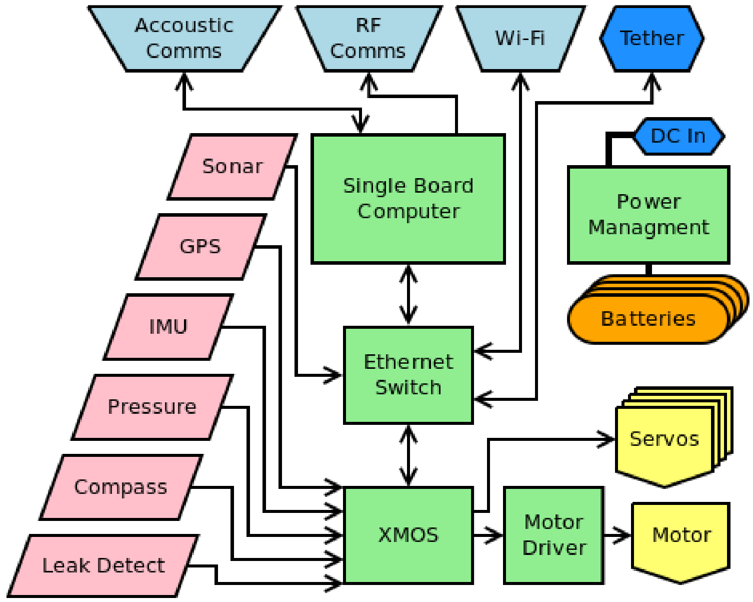

Figure 22: Yellowfin's Major Electrical Components

e. Electrical System

The electrical design for Yellowfin has gone through several revisions. The current revision reflects the lessons learned from the previous revisions and provides all of the core functionality needed for Yellowfin to operate successfully. Yellowfin uses as many COTS electrical components as possible rather than custom designed electronics. This choice was made at the expense of using additional space for some of the components; however, it allowed the first revisions of Yellowfin to be created much faster and at a lower price than would have been possible otherwise. Future revisions may include more custom parts, but the benefits of any redesign will always be weighed against the costs, both in time and money.

Communications

Yellowfin's wireless communications components are shown at the top of Figure 22 in light blue. Acoustic communication using the WHOI micro-modem, which is used throughout UUV research, allows Yellowfin to communicate underwater over fairly long distances at very low bandwidth.[53] Wi-Fi provides a high bandwidth data link when Yellowfin is at the surface of the water, but does not work underwater. Radio frequency (RF) communications falls between the other two in terms of bandwidth and works both above and below water. Yellowfin is also equipped with an optional Ethernet tether, shown in dark blue in Figure 22, which has been used for initial in-water drive testing and is used to provide a high-bandwidth connection to Yellowfin when it is docked.

Sensors

The main sensors are represented in pink on the left of Figure 22. Yellowfin has many of the most common navigation sensors, including a GPS, IMU, compass, and pressure sensor. These are used for low-level motion control and also for high-level localization. Yellowfin also includes moisture sensors for leak detection. The sonar in use for Yellowfin development is a BlueView Forward Imaging Sonar. In contrast with traditional sonar where a single beam is mechanically rotated, imaging sonar implement multi-beam sensors that form several small acoustical beams at once. Due to the nature of the imaging sonar, it makes it ideal for use on moving platforms, as movement of traditional sonar during its operation can cause data errors. Other sensor payloads can be easily substituted for this sonar using different nose cones.

Actuators

The actuators used for controlling the motion of Yellowfin are shown in yellow on the lower right of Figure 22. The drive motor, which spins the propeller, is a standard DC brushless motor powered by a motor driver, another COTS component. The four fins are controlled using digital servos. These actuators were chosen instead of more common varieties of linear actuators because field-testing revealed that the common varieties suffered from limited control precision and other associated mechanical problems.

Power

A simplified representation of the power system is shown in the middle right of Figure 22, consisting of a set of batteries, the power management system, and a DC power input connection, which can be used to charge the batteries without unsealing the hull. Yellowfin uses several high-capacity Li-ion batteries to achieve the desired operating duration, and additional batteries can be easily added for increased operating time. The power from the batteries is regulated and distributed to the other electrical components using several power management boards.

Computation

Processing on the submarine is split into two sections: the low-level processor, which manages all hardware and software integral to sensors and actuators, and the high-level processor, which manages the autonomy and collaborative behaviors of the submarine. This decoupling between high-level and low-level control was created in order to reduce the difficulty of modifying the more complex high-level software and to make that software platform-independent. It also reduces the risk of the less stable high-level software interfering with safety-critical low-level software. The low-level and high-level processors need to be able to communicate in a reasonably fast manner (high bandwidth, low latency) to exchange data. The low-level controller requires a large number of general-purpose digital I/O pins, which are used simultaneously and must still have enough processing resources remaining to respond to the high-level SBC, which itself must have enough processing power for high-level autonomy and data processing such as from the sonar.

The low-level processing is provided by an XMOS[*] controller which takes care of all of the platform-specific functionality of Yellowfin, including sampling the navigation and health sensors and generating correct control values for the fins and propeller. The XMOS microcontroller is a multi-threaded, event-driven, real-time microcontroller specifically designed to handle large amounts of digital I/O. It does efficient I/O using hardware interrupts generated by ports, which can be set to interrupt on a timer or when a specific state is reached. Each thread has its own dedicated set of registers, which allows preemptive multi-threading that adds very little overhead. Each processor core can run 8 threads simultaneously and multiple cores are connected on the same die using a switch-based channel communication. These features make the XMOS microcontroller suitable to process the types of data used by Yellowfin's low-level software and do it in a way that is potentially much more efficient than a standard design which typically involves a

[*] http://www.xmos.com/

medium sized micro controller for computation and communication and a separate FPGA which offloads the intensive task of sampling multiple digital I/O simultaneously. A Pico-ITX SBC performs the high-level processing. This form factor allows for a sufficiently powerful general processor with adequate standard interfacing options such as Ethernet, USB, and WiFi. The SBC can be upgraded or replaced easily, as the form factor is an industry standard, to conform to mission requirements.

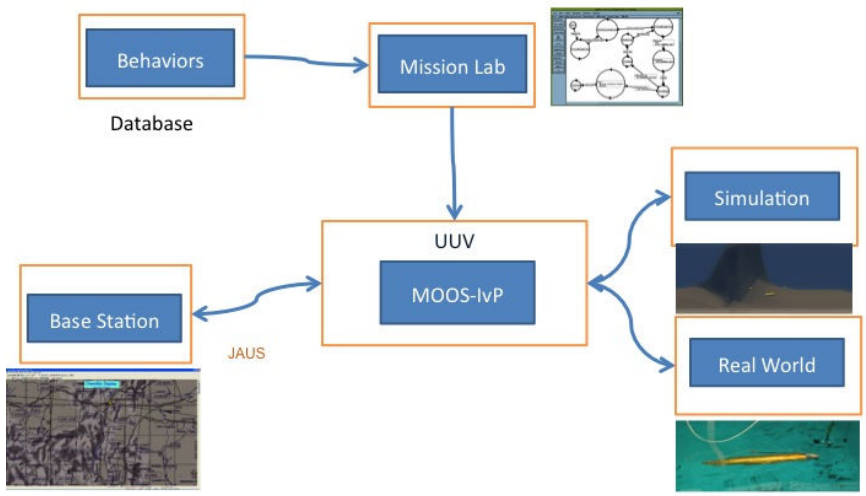

Figure 23: Yellowfin's Software For Mission Execution

f. Software System

The Yellowfin project has a complete software package from pre-mission planning to mission execution, as can be seen in figure 23. As described below, the pre-mission planning is performed with Mission Lab by organizing, from a database of available behaviors, the necessary mission behaviors. Mission execution on the vehicle is performed by the MOOS-IvP suite of applications for high-level autonomy and the XMOS low level controller. [545] Command and control of one or more vehicles is performed by a base station using FalconView. A mission can be executed in the Yellowfin simulator or on the actual vehicle.

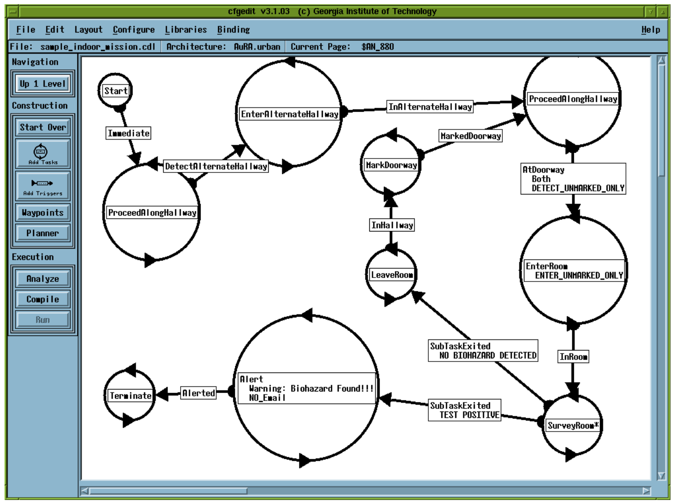

Figure 24: Mission Lab's CfgEdit

Mission Lab

Yellowfin utilizes Mission Lab for pre-mission planning. Mission Lab was created at the Georgia Institute of Technology for the purposes of organizing and executing behavior based architectures. In particular, Yellowfin has a database of behaviors which can be organized for a particular mission based on requirements through the use of a GUI application, as seen in figure 24, named cfgEdit. cfgEdit then translates the organized behaviors into mission files which Yellowfin's autonomy then executes during mission deployment.

FalconView Base Station

Command and control of one or more UUVs is performed through a base station using FalconView.* FalconView is widely used by the United States Department of Defense for its aircraft mission planning and mapping capabilities. FalconView provides for application extensions through a plug-in framework. The UUVs in this system communicate to a base station server through a WHOI acoustic modem and the FalconView application plug-in to display vehicle position and telemetry information in real time, using JAUS messages. The base station can also be used to send JAUS messages to the vehicles such as waypoint and mission based commands.

Vehicle Software Architecture

The high-level software architecture and autonomy for Yellowfin is provided by MOOS and the MOOS-IvP packages, respectively. MOOS is a C++ cross platform middle ware, created and maintained by the Oxford Mobile Robotics Group for robotics research. It enables cross process communication through a central database based on the publish-subscribe architecture. MOOS-IvP, which is maintained by MIT's Laboratory for Autonomous Marine Sensing Systems (LAMSS), is a collection of MOOS-based applications designed for maritime autonomy. MOOS-IvP includes applications for communication, simulation, data acquisition, pre-mission planning, and post-mission analysis. In collaboration with the developers of MOOS-IvP, Yellowfin leverages their software to reduce development time and increase the platform's capability.

---

* http://www.falconview.org

Acoustic Communication Software

As the Department of Defense has mandated that all unmanned systems will use the Joint Architecture for Unmanned Systems (JAUS) message protocol, Yellowfin has elected to use an open source implementation of JAUS called OpenJAUS.* OpenJAUS provides a library of message routines that code and decode a variety of JAUS message types including system health, UUV pose, mission directives, and sensor data communication. The Yellowfin software also includes a library of routines that encode and decode the JAUS messages into and out of the NMEA 0183 standard protocol utilized by the WHOI acoustic modems. A MOOS-based application called pJAUSCodec has been created to translate and relay the JAUS-encoded acoustic messages to and from variables in the MOOS database. Messages traveling to and from Yellowfin's WHOI acoustic modem are handled by the MOOS-IvP application pAcommsHandler. It includes such capabilities as message queuing while allowing access to the lower level WHOI firmware capabilities such as pinging for range through the use of MOOS database variables.

High-level Autonomy

IvP Helm is a behavior-based architecture that uses multi-objective optimization which allows for the coordination of multiple competing behaviors. Included with IvP Helm are 17 behaviors and the ability to create new ones with the use of an included toolbox. Additionally, behaviors may be clustered according to different mission modes which will make the entire cooperative system flexible to changing mission dynamics. Pre-mission planning for Yellowfin is performed by cfgedit from MissionLab which configures the appropriate mission behaviors. IvP-Helm then executes these behaviors at run-time.

Low-Level to High-Level Connection

Communication between the low-level XMOS controller and the MOOS database hosted on the single board computer (SBC) is conducted via Ethernet with a MOOS application created for Yellowfin called pyMoosServer. Upon being activated the XMOS controller sends a packet to pyMoosServer indicating which variables it wants to receive and which variables it wants to publish to the database. pyMoosServer then listens for strings containing variables and their values from the XMOS controller posting them to the MOOS database upon receipt. When the pyMoosServer receives an update to a subscribed variable it sends strings containing the variable name and value to the XMOS controller.

Low-level Controller

The software written for the low-level XMOS controller was made as modular as possible, which was made fairly simple due to XMOS's concurrency architecture. The software was written in a combination of XC, a language similar to C but with extensions which make use of the XMOS's special features, as well as in standard C and C++, which has made it much easier to port existing code to the XMOS. Yellowfin has separate modules to handle Ethernet communication, sampling each of sensors, sensor fusion, motion control, and health monitoring.

* http://www.openjaus.com

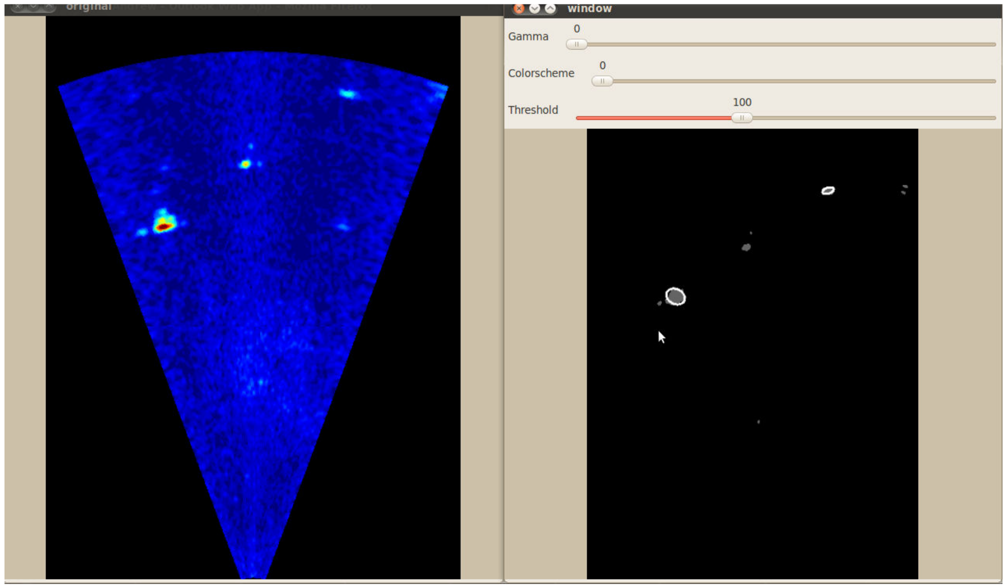

Figure 25: Yellowfin's Sonar GUI Tracking Buoys

Sonar Software

The sonar is crucial for determining range and bearing to detected features for Yellowfin's autonomy. Without this capability, the Yellowfin would not be able to perform target tracking nor simultaneous localization and mapping (SLAM). Yellowfin's sonar software provides a base level of manipulation and processing which includes the ability to stream and process live data as well as record sonar data for later analysis. Using OpenCV, an open-source computer vision library originally developed by Intel, feature extraction is implemented using computer vision techniques such as contour matching.*

* http://opencv.willowgarage.com

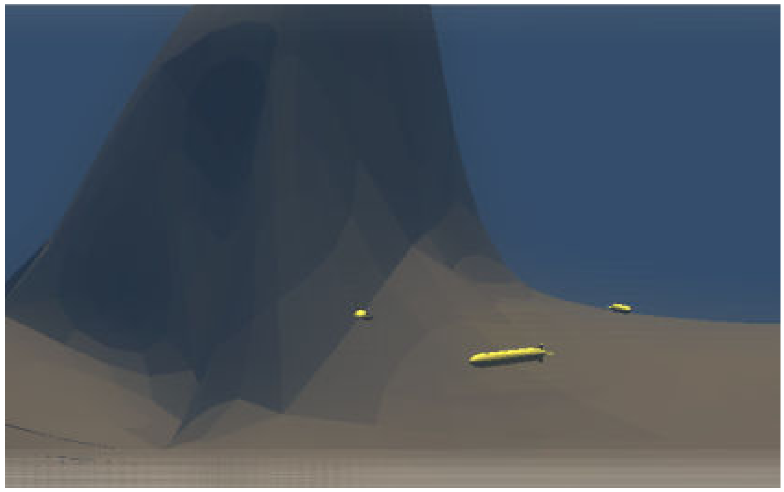

Figure 26: Yellowfin Simulator

G. YELLOWFIN SIMULATOR DESIGN

The Yellowfin simulator was designed for the purpose of developing autonomous behaviors and functionality in a fully realized 3D environment, as seen in figure 26. A key aspect of this was the ability to model the Yellowfin vehicle within the simulator and allow the Yellowfin software package to operate in a manner similar to that in which the vehicle will be deployed. Developing a simulator also provides the ability to develop multi-vehicle behaviors without requiring the cost of deploying multiple vehicles.

Visualization Software

The main visualization technology driving Yellowfin's simulator is the use of Blender, an open source cross-platform toolset for creating 3D worlds.* Blender provides a programming interface using the Python language as well as several graphical software development tools. A 3D model of the Yellowfin vehicle was imported into the simulation using Blender's 3D modeling tools.

Bathymetry Data

In order to simulate true deployment conditions, an underwater Bathometry can be imported into Yellowfin's simulator using GeoTIFF files of any location of interest. GeoTIFF is a non-proprietary standard for TIFF image files that contain embedded geographical or georeferencing data for use in constructing real world geographical images. These GeoTIFF files can be constructed from various sources including satellite imagery or elevation models.

Sensor Data Simulation

Yellowfin's simulator has the ability to simulate sensor data from the 3D environment. Ray tracing is utilized to create a sonar image of the simulated sonar's field of view. This allows for the testing of Yellowfin's target-tracking and simultaneous localization and mapping (SLAM) capabilities. These are two active research areas and contribute to Yellowfin's overall autonomy.

Simulator to Vehicle Interface

A MOOS database interface has been created for the Blender-based simulation in order to provide a gateway for the testing of Yellowfin's autonomy software. MOOS-IvP provides simulation tools, such as iMarineSim, to exercise a vehicle's autonomy. However, the provided simula-

* http://www.blender.org

tor does not include an attractive 3D visualization with perception sensor data such as sonar data from the simulated sea floor as provided by a GeoTIFF. Connecting Yellowfin's simulator to the MOOS Database allows the Yellowfin autonomy software to receive the exact same input the physical robot would receive during a mission and respond accordingly. This provides the ability to develop working autonomous behaviors in the simulator before actual deployment.

V. CONCLUSION

This paper provided an overview of the AUV research that is being conducted at the Georgia Institute of Technology. To summarize, GITs main research effort include underwater vehicle sensing, underwater communications, and underwater vehicle autonomy including heterogeneous multi-vehicle collaboration. In addition, the Yellowfin man portable autonomous underwater vehicle was presented. It has been designed to be deployed in both scientific and military missions. Thus, its payload is adaptable to mission requirements. The design process emphasized low cost, high functionality, and interoperability by designing and implementing with off the shelf parts, open-source software, and industry standards. The software of Yellowfin allows for pre-mission planning through mission execution of a single unit or a collaborative team of vehicles with robustness for different missions and real-time situational awareness. The Yellowfin simulator enables the ability to test the autonomy software in various life-like situations. Yellowfin's small size and robust software make it ideal for littoral missions and research into homogeneous collaborative behaviors.

REFERENCES

[1] D. R. Blidberg and J. Jalbert. *Mission and System Sensors*, pages 185–220. TSI Press, 1995.

[2] J. Leonard, A. Bennett, C. Smith, and H. Feder. Autonomous underwater vehicle navigation. *MIT Marine Robotics Laboratory Technical Memorandum*, 1998.

[3] R. Babb. Auv navigation for underwater scientific surveys. *Sea Technology*, 31(12), Dec 1990.

[4] J.S. Stambaugh and R.B. Thibault. Navigation requirements for autonomous underwater vehicles. *Navigation*, 39(1):79–92, Spring 1992.

[5] B.L. Hutchison and D.E. Shirley. A precision long range navigation and guidance system for an auv. In *OCEANS '90. 'Engineering in the Ocean Environment'. Conference Proceedings*, pages 594–599, 1990.

[6] Absorption coefficients obtained from the National Physical Laboratory website: http://resource.npl.co.uk/acoustics/techguides/seaabsorption/ using the default temperature of 8°C and a depth of 50 m.

[7] Lee Freitag, Matthew Grund, Sandipa Singh, James Partan, Peter Koski, and Keenan Ball, "The WHOI Micro-Modem: An Acoustic Communications and Navigation System for Multiple Platforms", Woods Hole Oceanographic Institution.

[8] Lee Freitag, "FHFSK Coding and Modulation Specification", Woods Hole Oceanographic Institution, December 12, 2005.

[9] Naveel Riz, Mohammad Ghavami, "Performance Comparison of Ultra-Wideband Time-Hopping, DSSS and OFDM Multiple Access Schemes for Wireless Sensor Networks," Proceed. of the WCNC.

[10] Joseph Nguimbis, Shijie Cheng, "Coupling Unit Topology for Optimal Signaling Through The Low-Voltage Power-line Communication Network," IEEE Transactions on Power Delivery, Vol. 19, No. 3, July 2004.

[11] Walter Y. Chen, "DSL: Simulation Techniques and Standards Development for Digital Subscriber Line Systems," Macmillan Technical Publishing, 1998.

[12] Ahmed I. Al-Shamma'a, Andrew Shaw, Saher SAman, "Propagation of Electromagnetic Waves at MHz Frequency Through Seawater, "IEEE Transactions on Antennas and Propagation, Vol. 52, No. 11, 2004.

[13] A. Shaw, A.I.-Shamma'a, S.R. Wylie, and D. Toal, "Experimental Investigations of Electromagnetic Wave Propagation in Seawater," Proceeding of the 36th European Microwave Conference, September 2006, Manchester U.K.

[14] Ralf Koetter, Andrew Singer, Michael Tuchler, "Turbo Equalization: An Iterative Equalization and Decoding Technique for Coded Data Transmission," IEEE Signal Processing Magazine, January 2004.

[15] Brian Kelley, "Jointly Optimized Software Radio for Low Power 4G Systems," Invited Talk, Proceedings of the IEEE Asilomar Conference on Systems, Signals, and Computers, 2007.

[16] Brian Kelley, "Software Defined Radio for Advanced Gigabit Cellular Systems," 2009 CRC handbook.

[17] 3GPP TS 36.212: 3GPP TS 36.211 v1.0.0, 3GPP TSPG RAN: Mutiplexing and Channel Coding (Release 8).

[18] Siavash M. Alamouti, "Simple Transmit Diversity Technique for Wireless Communications," IEEE Journal on Selected Areas in Communications, Vol. 16. No 8, October 1998.

[19] M. Rhodes. Underwater electromagnetic propagation: re-evaluating wireless capabilities. Hydro International, 10(10):28–31, 2006.

[20] WFS, Trial Summary, Flying Trial at Port Edgar

[21] W. Wu and Fumin Zhang, **"**A Switching Strategy for Robust Cooperative Exploration," in *Proc.49th IEEE Conference on Decision and Control (CDC 2010)*, accepted, 2010.

[22] C. Torney, Z. Neufeld, and L. Couzin, "Context-dependent interaction leads to emergent search behavior in social aggregates," Proceedings of the National Academy of Sciences, vol. 106, no. 52, pp. 22 055–22 060, 2009.

[23] C. Baldassano and N. E. Leonard, "Explore vs. exploit: Task allocation for multi-robot foraging," 2009.

[24] D. Baronov and J. Baillieul, "Search decisions for teams of automata," in Proc. of the 47st IEEE Conf. on Decision and Control, Cancun, Mexico, 2008, pp. 1133–1137.

[25] K. Lerman, C. Jones, Galstyan, and M. J. Mataric, "Analysis of dynamic task allocation in multi-robot systems," International Journal of Robotics Research, vol. 25, no. 3, pp. 225–241, 2006.

[26] J. L. Ny, M. M. Zavlan, and G. J. Pappas, "Resource allocation for signal detection with active sensors," in Proc. of the 48st IEEE Conf. on Decision and Control, Shanghai, China, 2009, pp. 8561–8566.

[27] K. Gu, V. L. Kharitonov, and J. Chen, Stability of Time-Delay Systems. Boston: Birkhauser, 2003.

[28] S. Elaydi and S. Zhang, "Stability and periodicity of difference equations with finite delay," Funkcialaj Ekvacioj, vol. 37, pp. 401– 413, 1994.

[29] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in Proc. of 41st IEEE Conf. on Decision and Control. IEEE, 2002, pp. 113–117.

[30] D. Simon, Optimal State Estimation. Hoboken, New Jersey: Wiley-Interscience, 2006.

[31] K. M. Nagpal and P. P. Khargonekar, "Filtering and smoothing in a H setting," IEEE Transactions on Automatic Control, vol. 36, pp. 152–166, 1991.

[32] I. Yaesh and Y. Theodor, "H¥ optimal estimation: a turtorial," in Proc. of the 31st IEEE Conf. on Decision and Control, Tucson, Arizona, 1992, pp. 2278–2286.

[33] P. Bolzern, P. Colaneri, and G. D. Nicolao, "Transient and asymptotic analysis of discrete-time H¥ filters," European Journal of Control, vol. 3, pp. 317–324, 1997.

[34] P. Bolzern and M. Maroni, "New conditions for the convergence of H¥ filters and predictors," IEEE Transactions on Automatic Control, vol. 44, pp. 1564–1568, 1999.

[35] W. Wu and Fumin Zhang, **"**Experimental Validation of Source Seeking with Switching Strategy" in *Proc.49th IEEE Conference on Decision and Control (CDC 2010)*, accepted, 2010.

[36] Albus, J. S. A Reference Model Architecture for Intelligent Systems Design. In Antsaklis, P.J., Passino, K.M. (Eds.) (1993) An Introduction to Intelligent and Autonomous Control. Kluwer Academic Publishers, 1993, Chapter 2, pp27-56. ISBN 0-7923-9267-1)

[37] Brooks, R. (1986). "A robust layered control system for a mobile robot". Robotics and Automation, IEEE Journal of Robotics and Automation 2 (1): 14–23. doi:10.1109/JRA.1986.1087032

[38] R.C. Arkin and T. Balch, "AuRA: Principles and practice in review." Journal of Experimental & Theoretical Artificial Intelligence, Vol. 2, No. 2, 1997, pp. 175-189.

[39] Albus, Albus, J.S. (2000). "4-D/RCS reference model architecture for unmanned ground vehicles". Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on Robotics and Automation.

[40] P. Ulam, Y. Endo, A. R. Wagner, and R. C. Arkin, "Integrated Mission Specification and Task Allocation for Robot Teams - Design and Implementation," Proc. IEEE Int'l Conf. Robotics and Automation, 2007, pp. 4428-4435.

[41] C. Le Pape, "A combination of centralized and distributed methods for multi-agent planning and scheduling," in *Proceedings of ICRA*, 1990, pp. 488–493.

[42] S. Botelho and R. Alami, "Plan-based multi-robot cooperation," in *Proceedings of the Dagstuhl workshop on plan-based control of robotic agents*, 2001.

[43] D. MacKenzie, R. Arkin, and J. Cameron, "Multiagent mission specification and execution," *Autonomous Robots*, vol. 4, no. 1, pp. 29–52, 1997.

[44] L. Parker, "ALLIANCE: An architecture for fault-tolerant multi-robot cooperation," *IEEE Transactions of Robotics and Automation*, vol. 14, no. 2, pp. 220–240, 1998.

[45] S. Botelho and R. Alami, "M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proceedings of ICRA*, 1999, pp. 1234–1239.

[46] B. Gerkey and M. Mararic, "A formal analysis and taxonomy of task allocation in multi-robot systems," *International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[47] R. Zlot, A. Stentz, and M. Dias, "Multi-robot exploration controlled by a market economy," in *Proceedings of ICRA*, 2002, pp. 3016–3023.

[48] M. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," Carnegie Mellon University, Tech. Rep. CMU-RI-TR-05-13, 2005.

[49] *MissionLab: User manual for MissionLab version 5.0*, Georgia Tech Mobile Robot Laboratory, 2002.

[50] P. Ulam, Y. Endo, A. Wagner, and R. Arkin, "Integrated mission specification and task allocation for robot teams - testing and evaluation," Georgia Institute of Technology, Tech. Rep. GIT-GVU-07-02, 2007.

[51] D. Furey, et al, "AUVSI/ONR Engineering Primer Document for the Autonomous Underwater Vehicle (AUV) Team Competition".

[52] B. Jun, J. Park, F. Lee, P. Lee, C. Lee, K. Kim. Y. Lim and J. Oh, "Development of the AUV 'ISiMI' and a free running test in an Ocean Engineering Basin." *Ocean Engineering,* Vol. 36, 2009 , pp. 2-14.

[53] M. R. Benjamin, J. J. Leonard, H. Schmidt, and P.M. Newman, "An overview of moos-ivp and a brief users guide to the ivp helm autonomy software." MIT, Tech. Rep. MIT_CSAIL-TR-2009-028, 2009.

[54] T. Bean, J. Canning, G. Beidler, M. O'Rourke and DB Edwards, "Designing and implementing collaborative behaviors for autonomous underwater vehicles." *Proceedings of UUST07*, pp. 19-22.