

APPLYING UNMANNED GROUND VEHICLE TECHNOLOGIES TO UNMANNED SURFACE VEHICLES

John Ebken*, Mike Bruch and Jason Lum
SPAWAR Systems Center, San Diego, 53560 Hull St., San Diego, CA 92152-5001

ABSTRACT

Development of Unmanned Ground Vehicles (UGVs) has been ongoing for decades. Much of the technology developed for UGVs can be applied directly to Unmanned Surface Vehicles (USVs) with little or no modification. SPAWAR Systems Center San Diego (SSC San Diego) has successfully demonstrated this by transitioning technology (both hardware and software) from a man-portable UGV to a USV demonstrator platform. By transitioning technology already proven in a UGV, SSC San Diego was able to develop a working USV much more quickly than would have been otherwise possible.

The technologies ported from the UGV to the USV include: the software architecture and protocol, tele-operation, a Kalman filter for state estimates, waypoint navigation, the Operator Control Unit (OCU), miniature processors, Ethernet switches and a video CODEC board.

KEYWORDS: robotics, unmanned ground vehicle, UGV, unmanned surface vehicle, USV, autonomous, waypoint navigation

1. USV BACKGROUND

The concept for the SPAWAR Systems Center, San Diego (SSC San Diego) USV was rapid production of a low cost, reliable USV to be used for technology development for transition to other unmanned assets and programs. The base USV would be required to perform basic tele-operation functions and waypoint navigation with an on-board observer to intervene for avoidance maneuvers. In addition to the basic control requirements, the USV had to be designed with flexibility to easily incorporate additional technologies that SSC San Diego is addressing for USVs.

While reviewing the requirements that had been created for the USV it became apparent that a significant number of the attributes required for the effort were already available on a ground platform that had been developed by SSC San Diego.



Figure 1. MPRS URBOT w/ GPS Evaluation Antenna

The URBOT (shown in Figure 1) is a Man Portable Robotic System (MPRS) developed for the Office of the Secretary of Defense (OSD) Joint Robotics Program (JRP) to provide a small ground vehicle with remote sensing capability¹.

To accomplish the project goals a study was conducted on candidate production vessels in the 18-22 foot range. This size range was chosen because it provides the capability to perform test and evaluation with personnel on board, eliminating the requirement for support vessels during development of new capabilities, and because it also provides a modest payload increase over vessels 15 – 18 feet in length while still being easy to store and transport. A jet drive was chosen for the platform to reduce mechanical complexity, increase safety and to provide more reliable operation near the kelp beds located off of Point Loma in San Diego. Candidate platforms included Rigid Hull Inflatable Boats (RHIB) and recreational sport boats. The platforms were evaluated with cost, ease of integration of required equipment and supportability. Using these criteria, the list of candidate platforms was reduced to the sport boat category and a SEADOO Challenger 2000 was selected. The platform configured as an unmanned vessel is shown in Figure 2.



Figure 2- SSC-San Diego Unmanned Surface Vessel

Modifications required for operating the vessel remotely included:

- Addition of an isolated 24 VDC power system to power auxiliary equipment for unmanned operation as well as payloads.
- Integration of actuators for steering, throttle and reversing bucket. Electrical actuators were selected due to ease of integration.
- Water-tight equipment enclosures installed to house electronic equipment. Three standard Fiberglass NEMA 4 enclosures were installed beneath the rear seat.
- Fabricating and installing a stainless steel equipment tower to support cameras and radar.

Locations were identified in the tub of the hull which provided ample space to incorporate the required modifications without using any free deck space that could later be used to place payloads under evaluation or restrict movement of engineers during testing. Figure 3 provides an interior view with labels identifying locations where additional equipment has been housed to provide operation in an unmanned state.

The actuation architecture was designed so that the USV has three distinct modes of operation: 1) Fully manual (mechanically linked) control, 2) Fly-by-wire control, and 3) remote tele-operation or computer control. This range of control was accomplished by effectively splitting the original flexible control cables in two and mounting the actuator housing in the middle. Quick release pins were used to either connect the

two cables mechanically (for fully manual mode) or to connect the actuators to the flexible cables attached to the USV control surfaces (for fly-by-wire and computer control modes). When set for fly-by-wire mode the flexible cables connected to the boat helm are still connected to linear position sensors inside the actuator housing. These sensors are used to track the movements of the steering wheel, throttle and bucket levers and position the actuators accordingly.

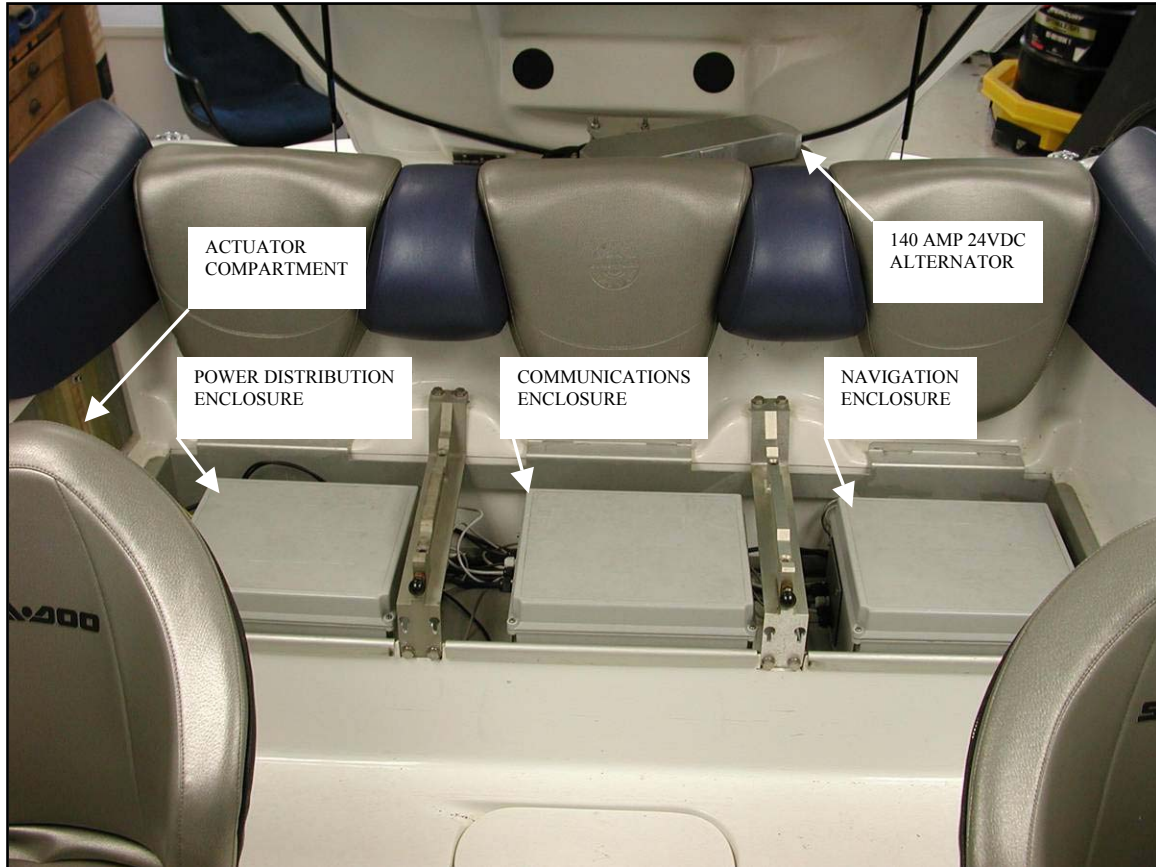


Figure 3. SSC San Diego USV interior modifications

2. UGV TECHNOLOGIES APPLIED TO THE USV

The following technologies, taken directly from existing ground vehicles at SSC San Diego, were utilized to decrease design and integration time of the JRP USV project.

2.1 Hardware

The control system for the USV is based on the hardware that had been proven to be reliable during the development and fielding of the URBOT. Packaging constraints onboard a small, man portable UGV dictate that size and weight of all components be optimized. The URBOT electronics box occupies a significant portion of the payload bay on the man portable vehicle and is quite densely packed. Components which generate large amounts of heat during operation, specifically motor controllers and RF amplifiers, are located outside the electronics box and dissipate heat directly to the aluminum structure for the vehicle. Power distribution inside the URBOT is handled by a power supply board that provides up to 12 separate devices with 5 or 12 volts DC. The individual power supply circuits on the board can be toggled on or off remotely by the operator. 10 Base-T ethernet is used on board the URBOT to link processor boards with communication, video and auxiliary sensor packages. Figure 4 is an interior view of the electronics box which provides vehicle control, precision navigation and IP based communications to the URBOT.

During development of the URBOT, multiple processors were integrated into the design to distribute loading. The first URBOT was strictly tele-operated and was configured with two processors. One processor provided an interface between the operator and the motor controllers and was designated as the Driver processor. A second processor, referred to as the Observer, was housed in the sensor package (snout) of the vehicle and provided the interface for the camera and light circuitry. As waypoint functionality became a necessity, a third processor known as the Navigator was added. The Navigator processor receives waypoint positions from the OCU and interfaces with the Driver processor as an independent controller.



Figure 4-URBOT Electronics Box

This same processor architecture has been adopted on the USV. The Driver is located in the actuator housing and commands the actuator positions based on either the fly-by-wire sensor data or tele-operation commands. The Observer is located on the sensor tower and interfaces to the camera and video CODEC hardware. The Navigator is positioned underneath the rear seat and hosts the waypoint navigation software. Data transferred between boxes is handled using 10/100 Base-T ethernet in the same configuration as that found on the URBOT.

2.1.1 ipEngine

The ipEngine (shown in Figure 5) is a small single board computer (SBC) manufactured by BrightStar Engineering, Inc. It contains a Motorola PPC CPU running at 66 MHz with 16 MB DRAM, 4 MB Flash and 10 Base-T ethernet. A 16K gate FPGA, dual RS232 ports, and general purpose I/O are also resident on the package which make it well suited to applications in robotics applications. The ipEngine is running Bright Star Engineering's pKernel operating system.

A daughterboard was designed and fabricated by SSC San Diego which provides access to functions on the motherboard. In addition to the native features of the SBC, the daughterboard provides:

- Auxiliary power outputs: 5VDC, 3.3VDC @ 1.5A
- 16 Analog Inputs
- 4 Analog Outputs
- 2 PWM outputs (variable)
- 2 Additional RS232 Serial Port
- 2 TTL Level Serial Port
- 28 General Purpose I/O

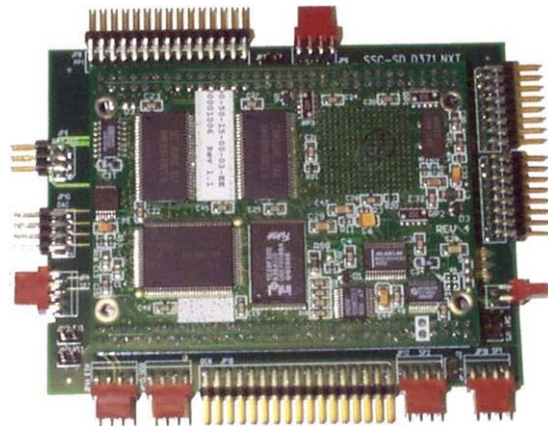


Figure 5- ipEngine mounted on SSC-SD Daughterboard
Package Dimensions: 4.2" X 2.2" X 0.6"

Even though the size and power limitations required for the URBOT were essentially eliminated on board the USV, it became apparent during the design phase of the USV that the ipEngine would ensure reduced integration time and assure system reliability. The ipEngine was originally used for all three processors (Driver, Observer and Navigator). Test and evaluation of the navigation system on board the URBOT, however, showed that processor loading on the ipEngine was near maximum under waypoint navigation. Code optimization provided minimal reduction in loading, so an alternate, more powerful SBC was sought that could be packaged in a similar physical envelope. The control and observation functionality on the vehicles are not processor intensive so the ipEngine remains an adequate solution for their needs.

2.1.2 686 Core

The 686 Core Module is manufactured by CompuLab, Inc in Haifa, Israel. The processor is a National Semiconductor Geode clocked at 266MHz with the Intel x86 architecture. The processor performs floating point arithmetic and contains the MMX instruction set.

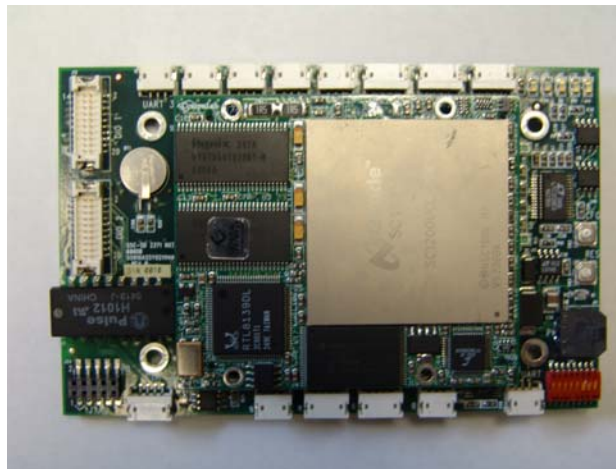


Figure 6- 686 CORE mounted on SSC-SD daughterboard
Package Dimensions: 4.4" X 2.8" X 0.5"

A daughterboard for the 686 core was designed to utilize the same mounting hole pattern as the ipEngine. The CompuLab 686 core module attached to a SSC San Diego daughterboard is shown in Figure 6.

The new daughterboard provides access to the following capabilities in conjunction with the SBC:

- PCI or ISA bus interface
- 3 USB ports
- I2C Bus
- Two 12-bit D-A output
- Five 12-bit A-D input
- Approximately 50 digital I/O
- FPGA (200,000 – 1,000,000 Logic Gates)
- Up to 8 configuration data pages stored on-board
- On board Voltage Regulator distributes 1.8, 3.3 and 5 VDC
- Stackable Configuration for additional expansion via the PCI bus
- 7 UARTs
- CAN bus
- 1 RS232 Serial Port
- Two 8-bit D-A output
- Seven 8-bit A-D input

The daughterboard provides an additional 4MB of SRAM and an 8051 microcontroller running at 20 MHz. The microcontroller provides two of the seven UARTs and the CAN bus interface. It is also used to preprocess some sensor data before sending it to the FPGA. The daughterboard can be used in a standalone fashion without the 686 core if desired.

2.1.3 VP604 Video CODEC

The VP604 is a custom, IP-based video and audio encoder/decoder produced by IndigoVision based in Edinburgh, UK. The VP604 is an enhanced version of a commercial product, the VP6000. The VP6000 is a single input transmitter/receiver that transmits, receives, encodes and decodes video, audio and data over an IP network. Enhancements to the product for robotic application included a significant reduction in size to fit into a small enclosure and the addition of 3 (providing 4 total) user selectable inputs or one output. Video data is encoded using the H.263 format providing video that can be transmitted at rates up to 30 frames per second. The user has the ability to reduce either the frame rate or bandwidth requirements during operation to optimize network and /or video performance.

automatic gain control for the audio interface from the board. The daughterboard is shown in Figure 7.

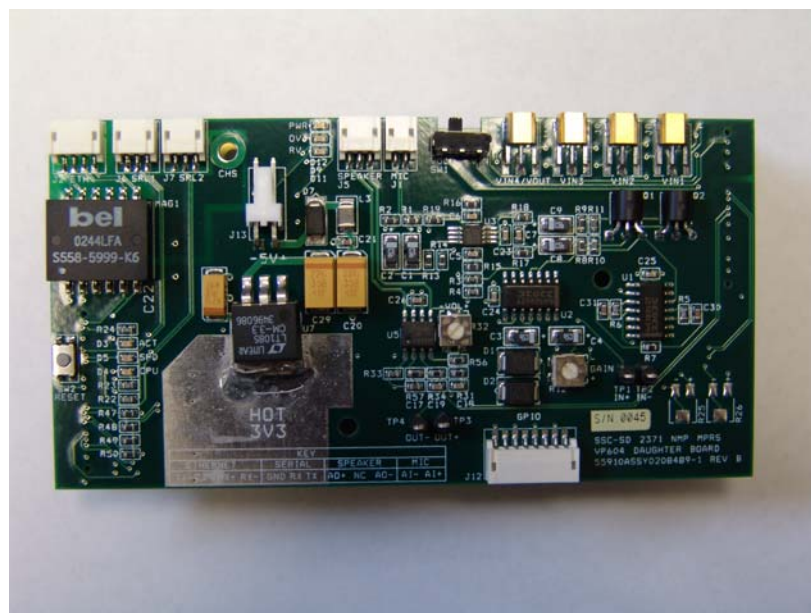


Figure 7. SSC-SD daughterboard (VP604 mounted beneath)
Package Dimensions: 4" X 2.2" X 0.5"

2.2 Software

A significant savings in time and cost was realized by re-use of software and architectures developed for UGV projects and programs at SSC San Diego. By reusing existing software tele-operation of the USV was accomplished in less than two weeks after completion of the hardware. Successful waypoint navigation was accomplished only a few weeks thereafter, following modification of the Kalman Filter and PID parameters for use in a marine environment.

2.2.1 Architecture

The Small Robot Technology (SMART) software architecture was developed for the URBOT UGV in October 1999³. It was designed to be compatible with the Multiple Resource Host Architecture (MRHA) that has been under development since the early 1990's at SSC San Diego⁴. In the SMART software all remote sensors are treated as independent domains. Capabilities or functions on the specific sensors are referred to as "agents". When a sensor becomes available its agents register with the controllers and their capabilities become available to the operator.

The major benefit afforded by adopting the SMART architecture is the dynamic resource discovery where sensors and capabilities register with all controllers as they become available. This provides the basis for interoperable systems and the ability to use an omni-controller type device. Additionally, the SMART Architecture is easily adapted to embedded processors and supports rapid prototyping through standard HW/SW interfaces.

The architecture has been used on multiple platforms and has been ported to a variety of operating systems including pKernel, WIN32 and Linux.

2.2.2 Tele-Operation

Tele-operation mode provides the operator remote control of the platform throttle and steering. Tele-operation instructions received from the OCU are transmitted to the Driver CPU on-board the platform which then sends required messages to the appropriate motor controllers to execute commands. Many vehicle control devices were evaluated during test and evaluation of the URBOT. When deployed in the field the URBOT OCU is carried in a backpack worn by the operator. Commands to the OCU are input by depressing push buttons on a pendant carried in one hand. When operating the URBOT from a PC a joystick is used. This provides the operator a simple, effective user interface for precise vehicle control. Operator feedback is provided through video from the on-board camera, diagnostic messages programmed into the vehicle, and navigation sensors on the vehicle.

For the USV a laptop with a joystick interface was selected for commonality of control between vehicles. The joystick interface can be tailored during system set-up to suit the operator's personal preferences. Initially, vessel speed was adjusted by increasing the joystick forward displacement from the neutral axis and turns are initiated by left or right displacement of the joystick. This mode is typically found on video game systems. Reverse direction of the vessel was caused by increasing the reverse displacement of the joystick. However, since USV velocity is closely coupled to small changes in joystick displacement, it may prove to be advantageous to assign the vessel speed function to a rheostat on the joystick.

2.2.3 Kalman Filter

To aid in semi-autonomous navigation an Extended Kalman Filter was developed for the URBOT to fuse sensor data and provide an optimal state estimate⁵. The filter has nine sensor measurement inputs and seven vehicle state outputs. During development of the filter two assumptions were made about the vehicle dynamics:

1. The vehicle translates only along its longitudinal axis
2. The vehicle rotates only on its vertical axis

A low dynamics assumption was also made which alleviated the requirement for acceleration states. A complete description of the URBOT Kalman Filter is contained in Reference 5.

The original Kalman Filter developed for the URBOT was modified and adapted for use on the USV. Modifications include the elimination of the vehicle odometry as a sensor input and tuning the covariance matrix to match the USV sensors. It was found that the Kalman Filter provides good approximations of the vessel state under a wide range of dynamic conditions. The assumptions about the vehicle dynamics described above do not pose as much of a limitation as might be expected. This is primarily due to the fact that when the vehicle is moving at a speed of several knots or more it is primarily translating along its longitudinal axis. In addition, the covariance matrix values are dynamically adjusted so that at very low speeds the GPS position and compass heading inputs are given greater weights and the GPS heading input weight is decreased. This has the effect of tracking position while maintaining the actual vehicle bearing. Currently, there are not separate vehicle states in the filter for vehicle bearing and course.

2.2.4 Waypoint Navigation

The waypoint navigation process developed for the URBOT was adopted entirely for use on the USV with great success. The waypoint navigation process includes receiving and parsing the path (route) message sent from the OCU, determining the vehicle's current position, calculating the current desired heading and velocity and executing the Proportional-Integral-Derivative (PID) controller to obtain that heading and velocity.

The process employed is most commonly referred to as the *follow-the-carrot/goal* method. A goal point is calculated on the intended path at a pre-defined look-ahead distance. The heading error is the difference between the heading to the goal point and the robot's current heading. Both the URBOT and USV implementation utilize a PID controller fed by the heading error (Fig. 8).

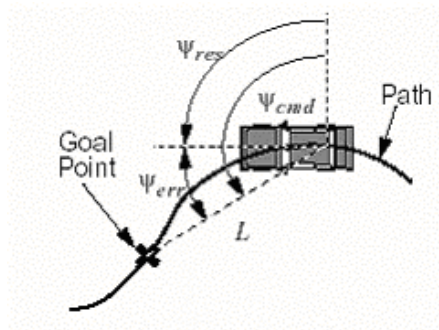


Figure 8. Depiction of heading error between robot's heading and heading to the goal point (Kelly, 1997)

The waypoint navigation scheme developed for the URBOT provided an immediate capability for use on the USV. However, since the velocities and vehicle dynamics on the USV are magnitudes larger than those encountered with the URBOT, PID values and look ahead distances required adjustment. It has been found that a dynamic table is required to establish proper values for the PID and look ahead distances on the USV to provide precision control in the various conditions encountered at sea. Tests have shown that typical values of track error for the USV are within 10 meters even at speeds of 25-30kts.

Velocity control of the USV has been implemented with a PID loop using the error between the commanded velocity of the planned route and the velocity state from the Kalman Filter. Typical velocity errors are less than 1kt.

2.3 Command and Control

When the waypoint navigation functionality was added to the URBOT development of a new map-based OCU was required. The operator now needed the ability to plot the vehicle course on a map or chart and be able to monitor the vehicle's progress along its intended path during execution. This was a significant departure from the simple backpack controller originally developed for the URBOT.

2.3.1 Multi Robot Operator Control Unit

The Multi-robot Operator Control Unit (MOCU) was specifically designed to augment the waypoint navigation capabilities of the URBOT. The Windows 2000/XP based software package was designed to control multiple autonomous vehicles from one operator station. Only one vehicle can be tele-operated at a time but multiple vehicles can be controlled and monitored simultaneously under waypoint navigation.

Using GEO-referenced TIFF aerial photographs as the map, MOCU originally provided the operator the ability to:

- Tele-operate the URBOT using a joystick or other controller
- Plan and create waypoint routes with point and click inputs on the map image
- Monitor the active vehicle operational parameters
- Suspend or stop any or all active routes
- View and control video feeds originating on board the vehicle

Figure 9 shows an early version of MOCU that was used to control the URBOT and the USV with a geo-referenced bitmap. The erratic line shown on the screen is GPS noise encountered during the test run. The straight line segments are the intended path and the remaining line is the actual path taken by the vehicle.

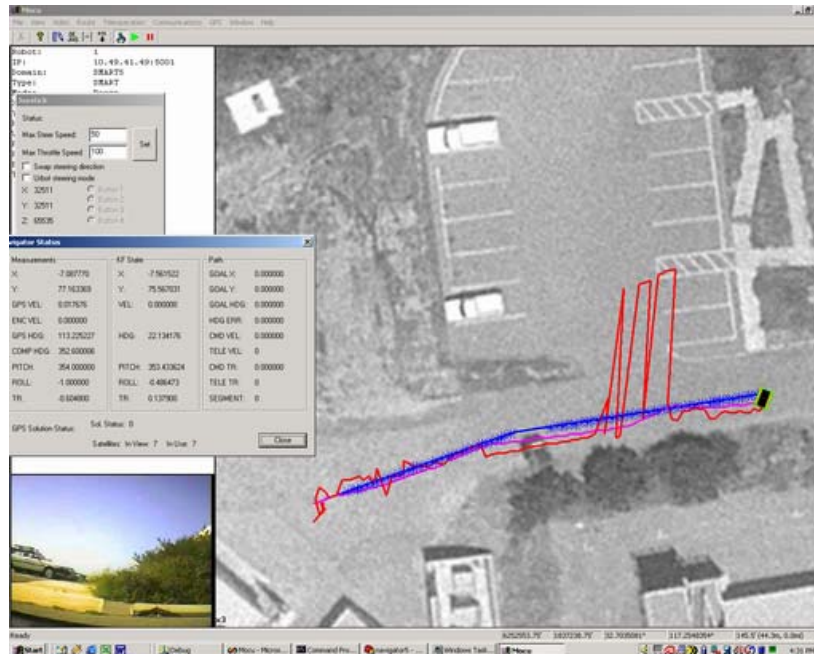


Figure 9. Early Generation MOCU Screenshot

For USV control a charting engine developed by SPAWAR Systems Center Charleston was integrated into MOCU to provide the ability to display and manipulate commonly used nautical electronic charts (DNC, S-57). Figure 10 is an image of MOCU configured for use with the USV.

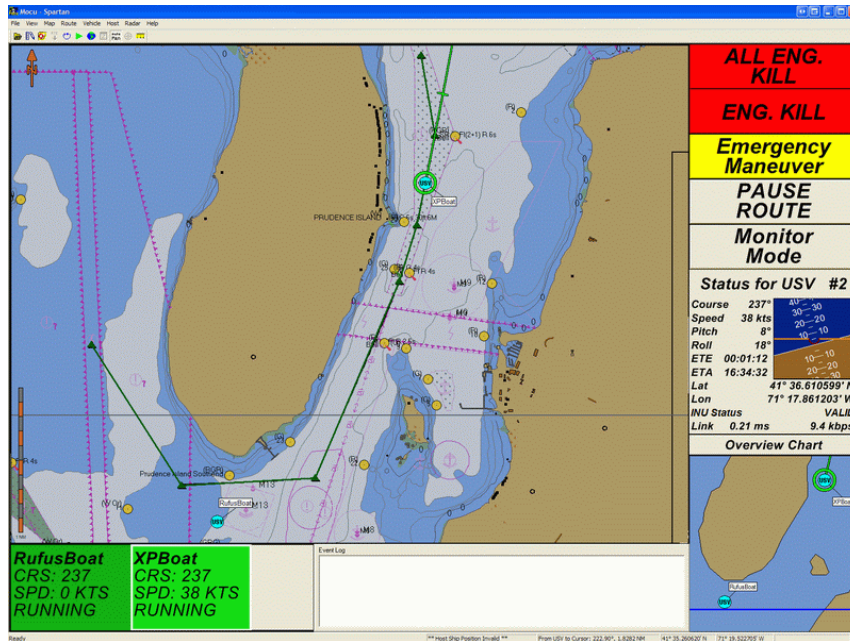


Figure 10. Screenshot of MOCU controlling two USVs

MOCU utilizes the dynamic registration feature of the SMART architecture to configure its display and interfaces depending on the type of vehicles that register. If the vehicle is selected by the operator, MOCU automatically configures its display to provide the user with a vehicle specific interface. This allows MOCU to control both the ground and water-based vehicles in a manner most appropriate for each domain.

The MOCU software has also been adopted by the Spartan ACTD, a USV development program. The Spartan program has facilitated the development of more USV specific features in MOCU including radar image and track overlay, standard military symbology, multiple map displays, overview charts, controller hand-off and more.

2.3.2 Joint Architecture for of Unmanned Systems (JAUS)

The JAUS architecture is an OSD JRP initiative to develop a common, domain level architecture into consumer, military and industrial unmanned systems. The Society of Automotive Engineers has voted to establish an unmanned systems standard based on the JAUS architecture. Adoption of the JAUS interface for robotic systems is increasing steadily. To meet the requirements of the Spartan ACTD SSC San Diego developed a series of new JAUS messages, including radar data transport and dynamic (on-route) re-configuration of the waypoint route.

A JAUS interface has already been developed for use with the URBOT utilizing the Spartan JAUS messages. Since the control system for the USV utilizes the same base code as that developed for the URBOT, adaptation of the JAUS interface for the platform should be accomplished with minimal effort. This will result in a JAUS-compliant USV by the Spring of 2005.

3. OBSTACLE AVOIDANCE

An absolute necessity for development of true autonomy in unmanned vehicles is reliable obstacle avoidance (OA). Development of methods and algorithms that will generate consistent behaviors when negotiating obstacles is required. Research efforts currently being conducted for SSC San Diego UGV systems have been performed with consideration for immediate transition to the USV².

The primary OA sensor on the USV is a digital marine radar system produce by Xenex Inc. The radar provides both the raw radar image and the contact track data over an IP connection. The raw radar image is

very similar to obstacle occupancy grid maps typically used in UGV OA systems. It is SSC San Diego's intention to primarily use the raw radar image for obstacle avoidance and add the ARPA contact data to the obstacle map. This is a unique approach and will be necessary because of the unreliable nature of the ARPA data and because ARPA only tracks moving objects.

3.1 Algorithms

The obstacle avoidance algorithm developed for the URBOT is loosely based on the CMU Morphin algorithm². The Morphin algorithm is a behavior based navigation architecture where various behaviors vote on a discrete set of actions. As applied on the URBOT a number steering of arcs are projected in front of the vehicle over the obstacle map (Fig. 11).

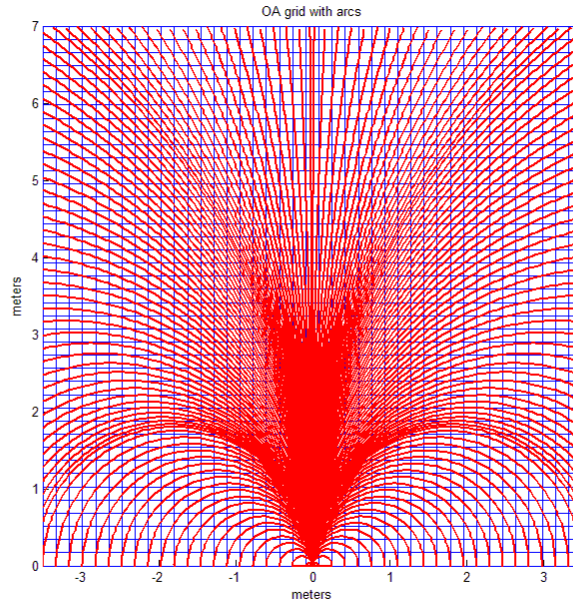


Figure 11. OA map and steering arcs

The number of arcs considered is a function of the map size and grid spacing. The arcs are spaced such that one arc passes through each of the outer cells. This guarantees that each cell in the grid is covered by at least one arc. Behaviors generate votes for arcs which best accomplish that behavior's goal. For instance, the obstacle avoidance behavior assigns a vote for each arc based on the distance the vehicle can travel along that arc until it is blocked by an obstacle. Votes from the various behaviors (OA, path following, target tracking, etc.) are combined in an arbiter which selects the most appropriate arc. The arc is then converted to a vehicle velocity and turn rate. This type of navigation system has been used in a variety of robotic systems including the NASA Mars Rovers.

The URBOT has a maximum speed of 4 mph and has a zero radius turn. The USV will operate at much higher speeds, cannot stop instantly and has a large turn radius (at speed). These differences will require that the obstacle map be much larger (on the order of 1km) but the cell size can also be increased by the same order of magnitude. This results in a computational problem of the same order of magnitude currently being used on the URBOT.

3.2 Stereo Vision/Laser Range Finders

A marine radar is the primary OA sensor on the USV but it has a minimum range of approximately 100m. This limitation severely limits the development of a robust OA system. To augment the radar SSC San Diego began investigating alternative sensors. Stereo vision has been used successfully on UGVs for many years and is currently being implemented on the MPRS URBOT. Scanning laser range finders are also in

wide use on many autonomous UGV systems. Both of these sensors have relatively limited ranges but provide high fidelity data and would augment the radar well. SSC San Diego is working in conjunction with a team from NASA's Jet Propulsion Lab (JPL) in Pasadena, CA to develop a stereo vision capability for use on the USV. Recent advances in processor performance and vision algorithms have made near real time detection possible. Coupling data from the laser and vision sensors with the radar return will enhance OA capabilities at close range.

4. CONCLUSIONS

By adopting a well developed architecture and adapting software and hardware created for Unmanned Ground vehicles, a significant reduction in construction and integration resources can be realized. In utilizing the ground vehicle technologies already available at SSC San Diego in the USV implementation an operational capability was achieved in a manner of weeks. Software development costs were minimal since the software for the control system had already been tested and implemented on the URBOT. Integration tests were mainly concerned with proper actuator response to operator input. Design and fabrication of mechanical components became the driving factor in the cost of platform development.

Robotic development and integration time can be dramatically reduced by adapting robust technologies created for different families of vehicles. Typically, the expected vehicle response to a given stimulus is independent of the medium in which the vehicle is operating. Since UGVs and USVs both operate in an essentially two dimensional world (they cannot submerge or fly) the control system rules are nearly identical, only the magnitude and direction of the response to a given situation can vary significantly. Realizing this characteristic early in the design of new vehicle types may lead to adaptation of existing vehicle systems which will reduce vehicle development time.

References

1. Bruch, M.H., Laird, R.T., and H.R. Everett, "Challenges for deploying man-portable robots into hostile environments," SPIE Proc. 4195: Mobile Robots XV, Boston, MA, November 5-8, 2000
2. Bruch, M.H., Lum, J., Yee, S., Tran, N., "Advances in Autonomy for Small UGVs", SPIE Defense and Security Symposium 2005, Orlando, FL, March 28 – April 1, 2005
3. R.T. Laird, M.H. Bruch, M.B. West, D.A. Ciccimaro and H.R. Everett, "Issues in Vehicle Tele-operation for Tunnel and Sewer Reconnaissance", Proceedings, Workshop 7, Vehicle Tele-operation Interfaces, IEEE International Conferences on Robotics and Automation, ICRA2000, San Francisco, CA, 28 April
4. Everett, H.R., R.T. Laird, G. Gilbreath, T.A. Heath-Pastore, R.S. Inderieden, K. Grant, and D.M. Jaffee, "Multiple Resource Host Architecture for the Mobile Detection Assessment Response System," *Technical Document 3026*, Space and Naval Warfare Systems Center, San Diego, CA; August 1998.
5. Bruch, M.H., G.A. Gilbreath, J.W. Muelhauser and J.Q. Lum, "Accurate Waypoint Navigation using Non-Differential GPS", AUVSI Unmanned Systems 2002, Lake Buena Vista, FL, July 9-11, 2002