

# AUTONOMOUS NAVIGATION AND OBSTACLE AVOIDANCE FOR UNMANNED SURFACE VEHICLES

Jacoby Larson\*, Michael Bruch, and John Ebken  
Space and Naval Warfare Systems Center, San Diego, 53560 Hull St., San Diego, CA 92152-5001

## ABSTRACT

The US Navy and other Department of Defense (DoD) and Department of Homeland Security (DHS) organizations are increasingly interested in the use of unmanned surface vehicles (USVs) for a variety of missions and applications. In order for USVs to fill these roles, they must be capable of a relatively high degree of autonomous navigation. Space and Naval Warfare Systems Center, San Diego is developing core technologies required for robust USV operation in a real-world environment, primarily focusing on autonomous navigation, obstacle avoidance, and path planning.

**KEYWORDS:** robotics, unmanned surface vehicle, USV, autonomous, waypoint navigation, obstacle avoidance, path planning

## 1. USV PLATFORM

The concept for the Space and Naval Warfare Systems Center, San Diego (SSC San Diego) USV was rapid production of a low-cost reliable platform to develop technology for transition to other unmanned assets and programs. The USV platform was chosen for its low cost, ease of systems integration, low maintenance, and similarity in performance characteristics to operational USVs. The platform itself is not the focus of the program but simply a convenient host for the core USV technologies being developed that will then be transitioned to other USV efforts in the Navy.

The platform selected for this development effort is a SEADOO Challenger 2000 sport boat with a jet drive, shown in Figure 1, configured as an unmanned vessel.



Figure 1. SSC San Diego USV operating autonomously in open water

Much of the technology and equipment that make up the basic components of the USV were transitioned directly from ongoing unmanned ground vehicle (UGV) programs at SSC San Diego.<sup>1</sup>

\*jacoby@spawar.navy.mil; phone 619-553-9775; fax 619-553-6188; [www.spawar.navy.mil/robots](http://www.spawar.navy.mil/robots)

## 2. AUTONOMY

SSC San Diego has been involved in the development of autonomous vehicles for over twenty-five years. Autonomous UGVs have become highly sophisticated and capable of robust navigation in very complex environments. Examples of this include the Defense Advanced Research Projects Agency (DARPA) Grand Challenge, the Army Research Laboratory (ARL) Demo III vehicle, and the National Aeronautics and Space Administration (NASA) Mars Rovers.

It is the intent of SSC San Diego to leverage technology developed in the UGV arena and adapt and apply the results in the USV domain. The rationale for this is not simply financial, but also because there are many corollaries between the UGV and USV environments. These include a mostly planar surface with potentially complex features such as hazardous terrain areas, pre-defined navigation ways and driving rules, dynamic obstacle environments, through-the-air sensing, etc. The SSC San Diego development approach for USV autonomous navigation capabilities is the same as for all of its unmanned vehicle programs. That is to develop basic, robust, transition-ready capabilities, and then build upon those capabilities with progressively more sophisticated and enabling functionality. The point is to transition and field technologies incrementally rather than spend years trying to develop the entire system all at once.

A simple example of this is shown in Figure 2, where the base functionality is waypoint navigation, which has been well demonstrated on the SSC San Diego USV and many other unmanned vehicles. Waypoint navigation without obstacle avoidance (OA), however, provides only limited capabilities to the warfighter in a real-world mission. In order to provide more functionality and reduce the reliance on operator oversight, a robust obstacle avoidance capability must be added. Once the obstacle avoidance piece has been sufficiently demonstrated, then more advanced behaviors can be added, such as autonomous recovery in the case of lost communications, target tracking and/or interception, etc. SSC San Diego is currently focused on the obstacle avoidance block which is the subject of the following sections.

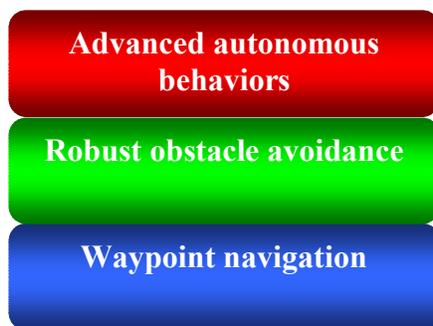


Figure 2. Basic capability blocks of an autonomous vehicle

### 2.1 Obstacle avoidance architecture

A two-tiered obstacle avoidance approach has been adopted, consisting of a near-field or reactive OA component and a far-field or deliberative OA component that operate simultaneously and in conjunction with one another. The primary function of the deliberative component is to continuously modify the existing waypoint route to plan around obstacles detected with the long-range sensors. The reactive OA component is responsible for avoiding obstacles in close proximity to the vehicle regardless of the vehicle's mode of operation or mission. Figure 3 attempts to depict this approach graphically. The deliberative OA component or path planning function attempts to modify the existing route from the outer horizon of the reactive component and beyond. Meanwhile, the reactive OA component is only concerned with obstacles between the reactive/deliberative boundary and the vehicle.

#### 2.1.1 Obstacle avoidance software architecture

To help set a frame of reference, the USV obstacle avoidance software architecture will be addressed, but only at a high level. Figure 4 is a basic block diagram of the OA software architecture as it exists on the USV. An attempt has been made to modularize the OA software into distinct functional components.

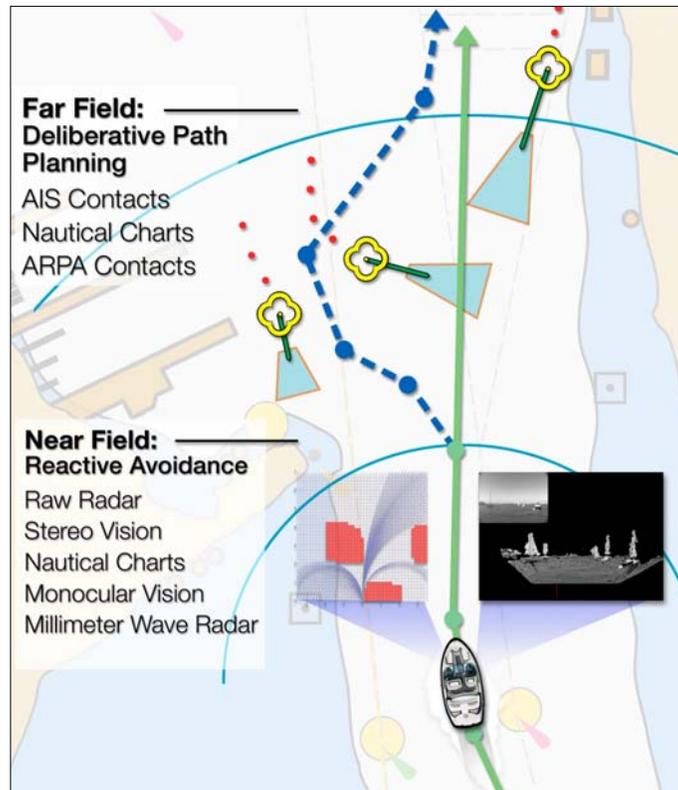


Figure 3. Graphical representation of the two-tiered OA architecture

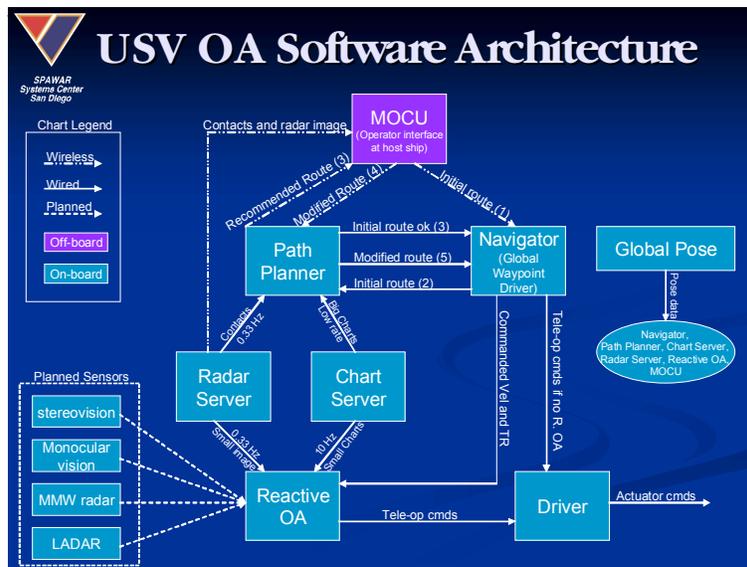


Figure 4. Block diagram of existing USV OA software architecture

The deliberative OA component consists primarily of the *path planner*, which interfaces directly with the *navigator*. The *path planner* receives data from both *chart server* and *radar server*. The reactive component intercepts the tele-operation or driving commands from the *navigator* and modifies them before forwarding them to the *driver* for execution at the actuators. The reactive component also interfaces with the *radar* and *chart server* but receives different types of data than is received by the *path planner*. The reactive component receives data from the other near-field sensors as well, such as the vision and ladar systems. It's worth noting that all of the interconnecting lines in the block

diagram are Joint Architecture for Unmanned Systems (JAUS) messages. In the future, the use of standard messages for the component interfaces should allow the unmanned systems community to exchange and share basic navigation components.

### **2.1.2 Deliberative obstacle avoidance**

The task of the deliberative obstacle avoidance component is to plan a path in the far-field that follows the original path as much as possible and avoids obstacles, both moving and stationary. It does this with the help of a *path planner* using a two-dimensional (2D) obstacle map. The obstacle map is essentially an occupancy grid<sup>2</sup>, which is created by dividing the environment into a discrete grid and assigning each cell location a value representing the probability of being occupied or not occupied by an obstacle. The map for the deliberative OA component is filled with stationary obstacles from the *chart server* and moving obstacles provided by the radar in the form of Automated Radar Plotting Aid (ARPA) contacts.

#### **2.1.2.1 Stationary obstacles**

Besides stationary obstacle data from the *chart server*, users may add their own obstacles to the map through operation or exclusion zones. Operation zones are sets of points that make up polygonal areas within which the USV should operate, and exclusion zones represent areas the USV should avoid.

Planning around stationary obstacles is rather trivial as proven best-path search algorithms have been used to plan optimal paths around stationary objects for years. The underlying search technique for the USV's deliberative obstacle avoidance is the A\* (A Star) search algorithm. A\* was chosen because it can find an optimal solution in a short amount of time. Also, since A\* uses a cost analysis at each step, inserting an added cost for proximity to obstacles was a natural process. This cost allows the USV to set a safety barrier around obstacles, which can also be adjusted for different obstacles. This cost analysis also can be extended for other costs such as direction, shipping lanes, "soft" obstacles, route time, etc.

#### **2.1.2.2 Moving obstacles**

Path planning around moving obstacles, however, is not a trivial task. Canny and Reif<sup>3</sup> showed that motion planning for a point in a plane with bounded velocity in the presence of moving obstacles is Non-deterministic Polynomial-time hard (NP-hard) or, in other words, very difficult. No research has been found which produces optimal solutions with computations in near-real-time as required by a USV in motion. Aggarwal and Fujimura<sup>4</sup> show that a more optimal solution can be found by adding a third dimension of time and plotting the location of the moving obstacles along that three-dimensional (3D) structure, but it is not close to meeting the time constraints of a USV planning 200 or 300 yards ahead of its current position while traveling 20 knots. Fujimura and Samet<sup>5</sup> provide yet another solution representing obstacles in time through a quadtree-type hierarchical structure, but even they admit the solution is best with few moving obstacles. The requirements for this project were that the solution not necessarily be optimal, but could allow for many obstacles and be completed within seconds. To keep the amount of time to search for a valid path manageable, the *path planner* has translated the third dimension of time of a moving obstacle to a 2D projected area. This truncation from three dimensions to two means that a solution can be found within seconds, but there is no guarantee the solution will be optimal nor always completely acceptable. For this reason, there still needs to be an extremely fast reactive obstacle avoidance system. Still, the deliberative obstacle avoidance will increase the likelihood of avoiding collisions and make the job of the reactive component much easier, resulting in less violent obstacle avoidance maneuvers. This idea is similar to an algorithm used for UAV path planning by Rathbun, Kragelund, Pongpunwattana, and Capozzi.<sup>6</sup>

##### **2.1.2.2.1 Velocity obstacle**

To avoid moving obstacles and maintain the waypoint path set by the user, the *path planner* determines safe velocity ranges using the Velocity Obstacle method.<sup>7</sup> This algorithm transforms a moving obstacle into a stationary one by considering the relative velocity and trajectory of the USV with respect to the obstacle. After producing a collision area called the Velocity Obstacle, defined using the relative velocity vector, the algorithm returns a set of USV velocity vectors guaranteeing collision avoidance. This transformation and collision area detection, when applicable, reduces the complexity of the path planning among moving obstacles problem to linear time. This is used as a first pass to avoid moving obstacles. However, in the case that changing velocity doesn't avoid collisions, the *path planner* changes the path by creating projected obstacle areas for each obstacle and determining a safe alternative route using the A\* search.

### 2.1.2.2.2 Projected obstacle areas

To achieve a more realistic real-time planner, a projected obstacle area (POA) was created, which is the area a moving obstacle could occupy in the future: a translation from 3D to 2D. The possible area a moving obstacle could occupy during the entire planning stages of a path could fill up the map with too many obstacles and hinder the movement of the USV. Therefore, it was necessary to identify a particular moment where the moving obstacle would pose the greatest threat (when the obstacle is at its closest point to the USV). This greatest threat is found using the closest point of approach (CPA), which is the minimum distance between the two objects in time along their respective paths. Since a moving obstacle can pose a threat to the USV along multiple stretches of the path, it was necessary to calculate the CPA of every obstacle along each path segment. This results in a single POA for each obstacle for each path segment, similar to snapshots of the moving obstacles in time, thus providing an accurate representation of the areas the USV will want to avoid.

### 2.1.2.2.3 Rules of the road

Because the projected obstacle area is an estimate of the location of the obstacle in the future, and because moving objects in water very rarely continue along the same heading at the same velocity, the POAs are made to handle uncertainty. Each POA contains uncertainty values to modify the associated area, simulating a moving obstacle that might speed up, slow down, and/or change direction, as shown in Figure 5.

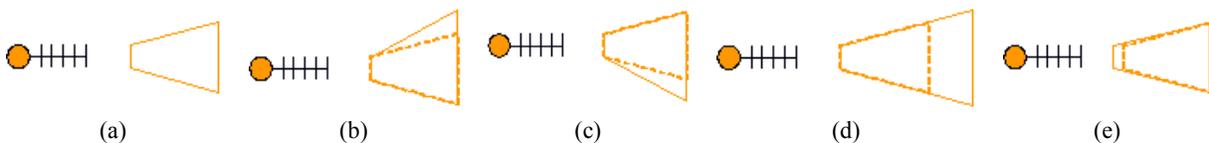


Figure 5. Projected obstacle areas display results of different uncertainty values (dashed lines represent the average POA and the solid lines represent the new POA. (a) An average POA (b) An increase in port angle (c) An increase in starboard angle (d) An increase in ahead distance (e) An increase in astern distance

Through the use of these uncertainty values, the deliberative obstacle avoidance component can mimic the rules of the road. Increasing the uncertainty angle on one side of a POA might mean that the lower cost path passes on the other side. The *path planner* addresses three rules of the road: 1) when meeting head on, pass port-to-port; 2) continue on course when meeting port-to-port or starboard-to-starboard; 3) when crossing, vessels found in your danger zone (on your right) have the right of way. An example of rule 1 is shown in Figure 6.

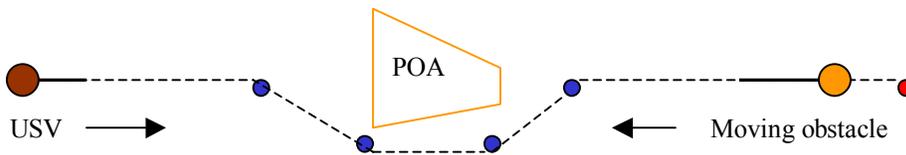


Figure 6. The projected obstacle area of the moving obstacle with an increased starboard uncertainty angle forces the USV to plan around the obstacle port-to-port.

### 2.1.2.2.4 Multiple iterations

The POA of a moving obstacle is calculated from the current path of the USV and the time taken to traverse that path. As that path changes, there is a need to update the POA and recalculate. For this reason, more than one iteration of the planner could be needed to account for updated POAs and changes to the path. Usually the program requires no more than 2-3 iterations to return a successful obstacle avoidance route.

### 2.1.2.3 MOCU interaction

The autonomy of the USV is still in its early stages, which requires that the user stays in the control loop at all times in the deliberative obstacle avoidance process. Any route change made by the *path planner* has to be accepted or rejected by the user. For this to occur, SSC San Diego's own Multi-robot Operator Control Unit (MOCU) needs to be able to communicate with the obstacle avoidance component and receive updated routes from it. Currently the process starts with the user creating a route for the USV to follow. This route is sent to the USV's *navigator* and immediately

executed. The *path planner* then begins to plan an obstacle-free path. If any significant part of the route has changed, the *path planner* notifies the user of an alternative path (Figure 7), giving up all control to the user and setting its planning on hold until a response is received. That path is displayed simultaneously with the current route in MOCU and the user can accept, accept and edit, or reject that path. If the user accepts the new path, those changes are sent to the USV's *navigator* as the current route. If the user accepts the path but makes a few changes, that path is sent back to the *path planner* to be planned again, which then is returned to the user for acceptance. If the user rejects the path, the *path planner* repeats its planning process.

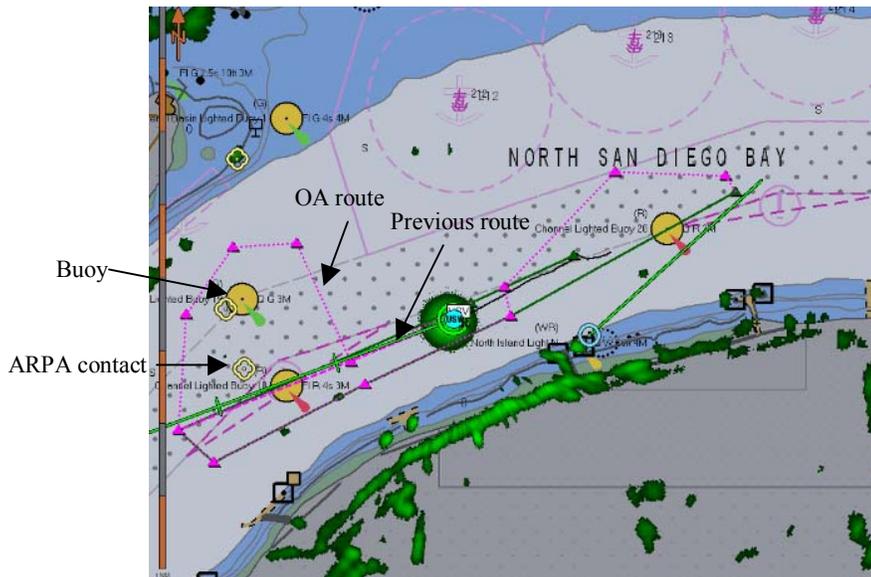


Figure 7. Screen capture of MOCU with an obstacle avoidance route displayed in pink dashed lines. The previous route is displayed as solid green lines. This particular OA route avoids the buoys (solid yellow circles) and ARPA contacts (yellow quatrefoil).

#### 2.1.2.4 Maintain route

When a user specifies a set of waypoints for the USV to follow, it is assumed they have a specific purpose for creating each one of those waypoints. Therefore, every time the *path planner* runs, it attempts to maintain as much of that user-defined route as possible. During each iteration, the deliberative obstacle avoidance component plans from the original route. If the USV has already deviated from the original route to avoid obstacles, then the planner attempts to return to the original route as soon as feasible. This removes any obsolete paths caused by volatile moving obstacles or unreliable ARPA contacts. An example is demonstrated in Figure 8.

#### 2.1.3 Reactive obstacle avoidance

The deliberative OA component or *path planner* attempts to modify the route so that it is as clear of obstacles as possible. The deliberative OA component does not guarantee obstacle avoidance, however, for multiple reasons: 1) the USV may inadvertently deviate from the planned path if GPS is jammed or the Inertial Navigation Unit (INU) drifts; 2) the long-range sensors are not capable of detecting small low-profile obstacles such as very small personal boats; 3) the deliberative OA component is only useful if the USV is in waypoint navigation mode. In a manual or tele-operation mode, auto-heading mode, target-tracking mode, etc., there is no preplanned path to modify. Because of these limitations, a reactive component has been included in the overall OA architecture.

The reactive OA component is responsible for avoiding obstacles that come in close proximity to the USV, regardless of the mode of operation or current mission assignment. It does so by intercepting any tele-operation or driving commands before they reach the actuators and modifying them in real time so as to avoid collisions.

The SSC San Diego implementation of reactive OA is a behavior-based common world-model approach. That is to say, all of the near-field sensors are fused into a common local world-model, and individual behaviors vote on specific navigation solutions within that model. For instance, the obstacle avoidance behavior votes for actions that avoid or turn

away from potential hazards while the path-following behavior votes for actions that will keep the vehicle on the planned path.

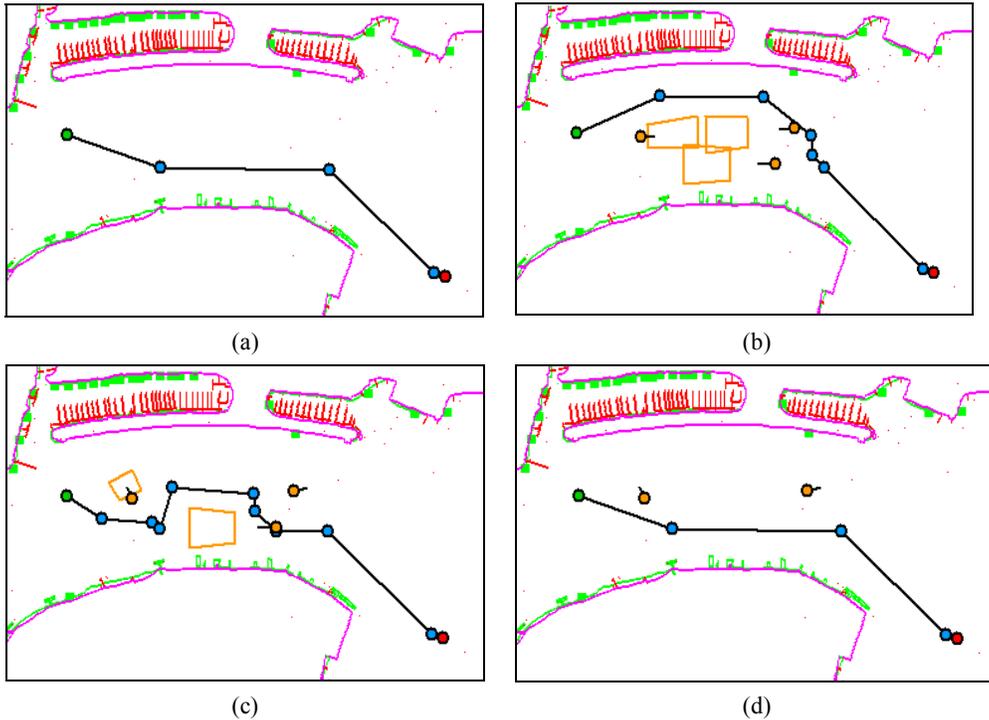


Figure 8. Maintaining the original route (a) The original route (b) OA route planned around 3 moving obstacles (c) OA route looks more like the original route when the headings of two of the moving obstacles have changed (d) The OA route completely returns to the original route when the changes from c are kept and the third moving obstacle is removed

This approach is not novel but has a long history of applications in real-world systems (including the Mars Rovers) and has its lineage back to the Carnegie Mellon University Morphin algorithm<sup>8</sup> and Distributed Architecture for Mobile Navigation (DAMN). As applied here, a number of arcs are projected in front of the vehicle over the local world-model obstacle map (Figure 9). The number of arcs considered is a function of the map size and grid spacing, with the arcs spaced such that one arc passes through each of the outer cells. This approach guarantees that each cell in the grid is covered by at least one arc so that all navigable paths are considered. Each of the arcs is related to the vehicle velocity and turn rate by

$$R = \frac{V}{\dot{\theta}}$$

where R is the radius of the arc, V is the vehicle velocity, and  $\dot{\theta}$  is the vehicle turn rate.

For the OA behavior, each arc is given a weight or vote based on the distance the robot could travel along that arc before it encountered an obstacle.<sup>9</sup> The longer arcs are weighted more favorably than the shorter arcs or arcs that are blocked near the robot. The votes are scaled from 0 to -1 so that they can be combined with votes from other navigation behaviors by the arbiter (Figure 11, top right).

One common navigation behavior is waypoint navigation or route following. The waypoint navigation function produces a commanded turn rate and velocity based on the heading error between the robot's current heading and the heading toward the goal point along the path (Figure 10). At each cycle these turn-rate and velocity commands from the waypoint navigation function are converted into an arc. To form votes for the entire array of arcs, the primary arc is

given the maximum value and votes for arcs on either side of it are linearly decreased until they reach a value of zero (Figure 11, top left). The path following behavior does not vote against any arcs.

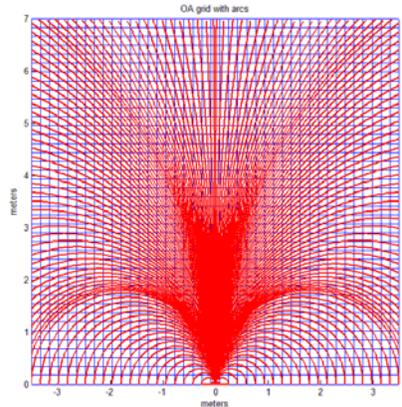


Figure 9. Obstacle grid with arcs shown

Other arc-voting behaviors have also been added to help navigate around obstacles. These include a free-space behavior that votes (from 1 to 0) for large continuous sections of arcs that are unblocked, essentially open areas in the obstacle map (Figure 11, bottom left). This behavior also votes for open arcs that fall between obstacles so that the robot won't always favor going around groups of obstacles.

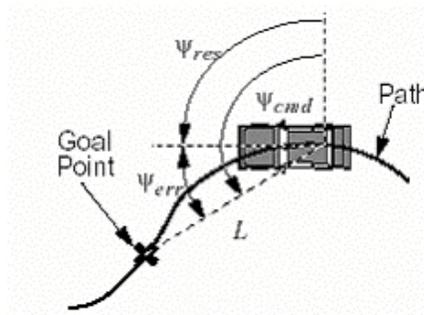


Figure 10. Depiction of heading error between robot's heading and heading to the goal point.<sup>10</sup>

Weighting factors are applied to the votes from each of the behaviors prior to combining them so that some behaviors contribute more to the final outcome than others. In the current implementation the OA behavior is weighted almost three times heavier than any of the other behaviors. The combined votes are shown in the bottom right plot in Figure 11.

### 3. SENSORS

As with any unmanned vehicle attempting to navigate in a complex environment, good sensor data is critical, and getting good data is often the most difficult part of the project. The marine environment poses many challenges including waves, spray, and a chaotic obstacle setting. There are, however, some advantages to the marine environment including well charted operating areas, absence of negative obstacles (holes or cliffs), a mostly planar surface (except for the waves), no vegetation, etc. It's important that the sensors are selected to make the most of the environmental advantages and to provide the best data possible in the challenging realms. For the purposes of this paper, the sensors have been divided into two categories based on which OA component they support.

### 3.1 Deliberative OA sensors

The sensors for the deliberative OA component need to provide data about obstacles in the far-field (e.g., >200-300 yards) and provide state information (position, course, and speed) for the moving obstacles. These are the sensors that are most commonly used in the commercial marine industry and include nautical charts and X-band radar.

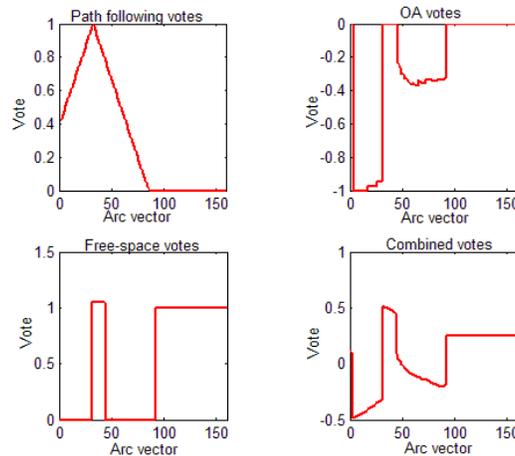


Figure 11. Example votes from behaviors

#### 3.1.1 Digital Nautical Charts

The National Geospatial-Intelligence Agency's (NGA) Digital Nautical Charts (DNC) are, according to the NGA's website, "an unclassified, vector-based, digital database containing maritime significant features essential for safe marine navigation".<sup>11</sup> A *chart server* was created to parse through the hierarchy of thousands of files and convert all the locations of permanent stationary obstacles (coastlines, piers, buoys, shallow water, etc.) from latitude/longitude points to occupancy grid pixels in the obstacle map. The *chart server* sorts the DNC data into four main categories of obstacles: above water, on the water's surface, below water, and land. Only those features with attributes of vertical clearance, horizontal clearance, or depth clearance that constitute a threat to the USV are added to the obstacle map. The data in the DNCs is encoded in a Vector Product Format (VPF), signifying that the data can be represented at any resolution without losing quality. This will become very important as the USV tackles missions ranging in hundreds of nautical miles but still avoiding obstacles in channels no wider than 50 yards.

Once the obstacle map has been created at a resolution beneficial for planning an obstacle free route, it is delivered to the deliberative OA component. This map is dynamically updated whenever the USV moves close to the edge of the map, when it is close to the edge of the DNC libraries' boundaries, or when a new route extends beyond the edge of the map. Currently the maps are created at a scale of 1000 pixels wide by 1000 pixels high, with each pixel representing 10 meters by 10 meters. This resolution was chosen based on the size of the USV, the size of most relevant obstacles, and the intent to keep a safe zone around the USV of at least 50 meters.

#### 3.1.2 Radar contacts

The radar system on the SSC San Diego USV is a standard marine radar (Furuno) with a third-party PC controller. The controller, developed by Xenex Innovations Ltd., provides a digital networked interface for the radar. The Xenex system provides an API to access the radar data and controls as well as an Advanced Radar Plotting Aid (ARPA) Software Development Kit (SDK), which provides algorithms to automatically acquire and track up to 100 contacts.

SSC San Diego has invested a significant amount of effort in characterizing the performance of the radar's ARPA function as this is the primary means of detecting uncharted obstacles in the far-field. As with any sensor in the real world there are many idiosyncrasies with the ARPA function. One significant problem with the radar is that it tends to classify noise from the shoreline return as contacts which are often shown to be moving at a significant velocity and in the direction of the USV. These false contacts are obviously detrimental to the successful operation of the *path planner*.

To mitigate this problem, the on-board nautical *chart server* can be used to calculate polygons that follow the shoreline and structures along the shoreline. The radar contacts are compared with these polygons and those that fall inside a polygon are rejected and deleted from the radar's list. An example of this is shown in Figure 12, where the blue circles and green leader lines indicate the current list of ARPA contacts.

Another challenge with the radar is that for a small, highly maneuverable boat, the turn-rate can approximate that of the radar itself. During high turn-rate maneuvers, the radar is either turning much faster or slower than normal, relative to earth, and the data is therefore skewed. When this occurs most often the contacts are lost until the USV returns to a relatively straight trajectory. Multiple approaches to mitigate these effects are currently under investigation.

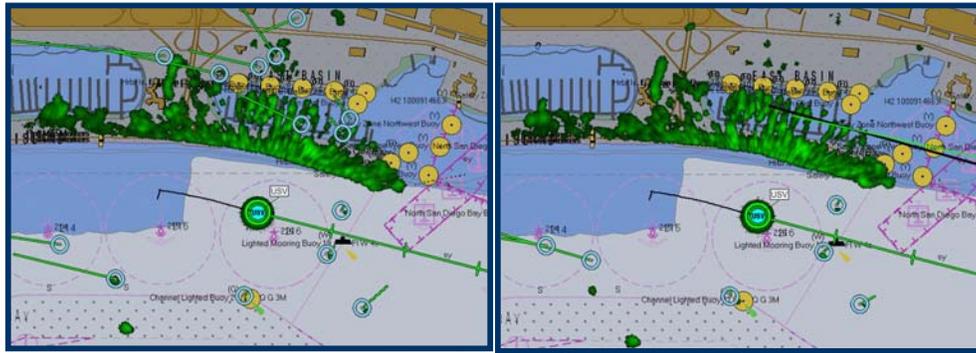


Figure 12. Radar contacts with no filtering (left), with shoreline filtering (right)

### 3.2 Reactive OA sensors

The sensors for the reactive OA component need to provide high-resolution data about obstacles in close proximity to the USV (e.g., <200-300 yards) and at a much higher rate than the deliberative sensors. Some of these sensors are typically not found in the commercial marine industry but many have been used extensively in UGV programs.

#### 3.2.1 Stereo vision

Stereo vision sensors have been commonplace on autonomous unmanned ground vehicles for many years and are the primary obstacle detection sensors on the NASA Mars Rovers. SSC San Diego has been working with the NASA Jet Propulsion Laboratory (JPL) for a number of years to transition technology to its UGV programs. That work is now being extended to the USV domain with very promising initial results. The stereo vision system provides high-resolution 3D data about the near-field environment, which can be converted into a 2D obstacle map and fused with data from the other reactive sensors.

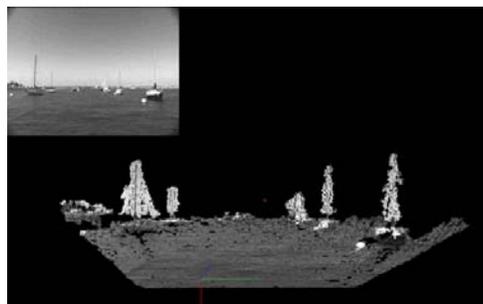


Figure 13. A 3D point representation of the scene in the upper left corner produced by the NASA JPL stereo vision system

#### 3.2.2 Monocular vision

Stereo vision is capable of providing very high quality 3D data but also has the disadvantage of requiring precise calibration every time the cameras are mounted. There is also the risk that the cameras may move relative to one another slightly which will affect the calibration and result in erroneous data. Because of some of the potential pitfalls of stereo vision, SSC San Diego is also investigating a monocular vision solution. The concept is that in any given frame of video, the horizon line is detected and obstacles are located on the water portion of the image. Since the distance to the horizon is known (from the nautical charts), and the height and properties of the camera are known, a rough estimate of

range can be calculated for each obstacle given the number of pixels below the horizon line an obstacle appears in an image.

SSC San Diego has developed the basic algorithms for detecting the horizon line under varying conditions (in the presence of city skyline, for example) as well as the algorithms for segmenting obstacles on the water. The initial results of this work are very promising and will be published in a future paper.

### 3.2.3 Digital Nautical Charts

The *chart server* also provides the reactive obstacle avoidance component a small portion of the larger deliberative planning map. The larger map is sectioned off, rotated so the heading of the robot is up, and delivered to the reactive obstacle avoidance component at a rate of 10 hertz. Although the usefulness of the chart data depends on the accuracy of the USV position estimate, it still makes sense to take every advantage of the nautical charts as much as possible.

### 3.2.2 Radar images

Like the nautical charts, the radar is used again in the reactive component, but this time it's the raw radar return data that is used. The radar return is, in essence, a ready-made obstacle map. In an ideal radar image, only obstacles on the water or the shoreline show returns. The radar data is converted from the polar scan line format to the Cartesian obstacle grid representation and fused with the other sensor data. Of course, the data is never ideal and often contains noise. At first inspection, it also appears that the radar returns no useful data within approximately 100 yards of the USV as it is a solid disk of noise. Upon further investigation, however, SSC San Diego has determined that both the image noise and the center disk noise can be filtered out, yielding useful obstacle data within the 100 yard radius, which is critical for the reactive OA component.

## 4. FUTURE WORK

### 4.1 Path planner

The Velocity Obstacle method produces both safe velocities and trajectories that avoid moving obstacles. Currently, the USV *path planner* only uses the velocity output in its calculations. A further investigation of safe trajectories could enhance obstacle avoidance.

Currently the human involvement in the process of accepting or rejecting the OA route can cause latency. This time delay could result in collisions as new threats appear but the user hasn't yet released control. In the future, as the deliberative obstacle avoidance component proves more reliable, the communication between the *navigator* and *path planner* will be automated (with user oversight of course). The *navigator* would accept every new path, given that each path is checked for meeting certain safety requirements. This will give the *path planner* more time to plan more paths, correct any outdated routes caused by noisy radar data or ARPA contacts that have changed their heading, and give a more accurate response to current environmental threats.

### 4.2 Sensors

SSC San Diego has plans to incorporate additional sensors for both the deliberative and reactive OA components. Additional sensors for the deliberative component will include an Automatic Identification System (AIS) receiver and more detailed bathymetric data. The AIS system receives position, speed and course data broadcasts from other marine vessels with compatible systems. This will provide very accurate data on at least some of the other contacts in the environment. The current nautical charts have only sporadic depth data. The addition of more detailed bathymetric data will complement the DNC data well.

Additional sensors for the reactive OA component will likely include millimeter wave (MMW) radar and ladar systems. SSC San Diego has conducted promising preliminary tests with a scanning ladar system and found that it did not report any return from the water's surface and only detected obstacles on the water.

## 5. CONCLUSION

An autonomous navigation and OA architecture for USVs has been developed by SSC San Diego that supports both deliberative and reactive OA and accommodates a variety of sensors with both long and short fields of view. This autonomy architecture has been developed under the JAUS framework for future interoperability and portability. Many of the software components of that architecture have been implemented and are being actively tested on the USV.

A quick and capable *path planner* has been developed to divert the route of a USV away from threats of stationary and moving obstacles in the far-field. On average, the *path planner* can successfully plan around potential collisions with more than 10 moving obstacles in less than 3 seconds. SSC San Diego will continue to make this *path planner* more robust to handle a variety of situations and obstacles in the aim of autonomy. The reactive OA software is being transitioned from existing ground vehicle programs and will be tested on the water in the near future.

The combination of a flexible and portable architecture, in depth sensor characterization and analysis, robust planning algorithms, and many hours of on-the-water testing will produce a reliable, transition-ready autonomous navigation and OA capability for use on many USVs throughout the Joint Forces and Department of Homeland Security.

## REFERENCES

1. J. Ebken, M. Bruch, J. Lum, "Applying unmanned ground vehicle technologies to unmanned surface vehicles", Proceedings of SPIE Unmanned Ground Vehicle Technology VII, **vol. 5804**, pp. 585-596, Orlando, FL, 2005
2. A. Elfes, "Using occupancy grids for mobile robot perception and navigation", Computer, **vol. 22**, pp. 46-57, 1989
3. J. Canny and J. Reif, "New lower bound techniques for robot motion planning problem", Proceedings of 28<sup>th</sup> Annual IEEE Symp. on Foundation of Computer Science, pp. 49-60, Los Angeles, CA 1987
4. N. Aggarwal and K. Fujimura, "Motion planning amidst planar moving obstacles", Proceedings of IEEE International Conference on Robotics and Automation, **vol. 3**, pp. 2153-2158, San Diego, CA, 1994
5. K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles", IEEE Transactions on Robotics and Automation, **vol. 5**, pp. 61-69, 1989
6. D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, "An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments", Proceedings of the 21<sup>st</sup> Digital Avionics Systems Conference, **vol. 2**, pp. 8D2-1 - 8D2-12, Irvine, CA, 2002
7. P. Fiorini and Z. Schiller, "Motion planning in dynamic environments using the relative velocity paradigm", Proceedings of IEEE International Conference on Robotics and Automation, **vol. 1**, pp. 560-565, Atlanta, GA, 1993
8. R. Simmons, L. Henriksen, L. Chrisman, and G. Whelan, "Obstacle avoidance and safeguarding for a lunar rover", Proceedings from the AIAA Forum on Advanced Developments in Space Robotics, Madison, WI, 1998
9. M. Bruch, J. Lum, S. Yee, and N. Tran, "Advances in Autonomy for Small UGVs", Proceedings of SPIE Unmanned Ground Vehicle Technology VII, **vol. 5804**, pp. 532-541, Orlando, FL, 2005
10. L. Matthies, A. Kelly, T. Litwin, G. Tharp, "Obstacle detection for unmanned ground vehicles: a progress report", Robotics Research: The Seventh International Symposium, Springer-Verlag, 1996
11. National Geospatial-Intelligence Agency, <http://www.nga.mil/portal/site/dnc/>