# Mission Planning in a Dynamic Ocean Sensorweb

**David R. Thompson**[1], **Steve Chien**[1], **Matthew Arrott**[2], **Arjuna Balasuriya**[3], **Yi Chao**[1], **Peggy Li**[1], **Michael Meisinger**[2], **Stephanie Petillo**[3], **Oscar Schofield**[4]

[1] Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr., Pasadena, CA 91109

[2] Calit2, University of California, San Diego, 9500 Gilman Drive #0436, La Jolla, CA 92093

[3] Laboratory for Autonomous Marine Sensing Systems, MIT, 77 Massachusetts Avenue, Cambridge, MA 02139

[4] Coastal Ocean Observation Lab, Rutgers University, 71 Dudley Road, New Brunswick, NJ 08901

## Abstract

The Ocean Observatories Initiative (OOI) will deploy a network of in-situ sensors, orbital assets and numerical models that synthesize collected data to forecast ocean conditions. We have implemented a prototype mission planning system for autonomous glider submersibles within this infrastructure. This system will be refined for the upcoming Observing System Simulation Experiment (OSSE) and eventual incorporation into the OOI. We describe our two-stage planning process involving abstract *cartographic* and more complete *detailed* representations. A new path planning algorithm exploits current forecasts to compute efficient paths between operator-specified waypoints. Experiments demonstrate significant improvements in path selection for a case study involving Regional Oceanographic Modeling System (ROMS) current forecast data.

## Introduction

Recent work in Earth-observing sensor networks demonstrates autonomous coordination of in-situ and orbital sensors that monitor, detect, and react to dynamic climactic phenomena in real time (Chien et al. 2005; Morris et al. 2008). In coming years the Ocean Observatories Initiative (OOI) will carry these concepts to the deep ocean (Given, Banahan, and Williams 2007). The OOI will incorporate platforms such as moorings and autonomous submersibles. It will synthesize these measurements with numerical models that incorporate collected data and generate forecasts for future conditions. A data distribution network will disseminate these forecasts to users around the world, while an ongoing science investigation will refine the core models and improve our understanding of physical processes in the dynamic ocean.

The OOI infrastructure will support underwater gliders, autonomous vehicles are which are well suited for long-term missions (Schofield et al. 2007). They locomote by changing buoyancy to generate vertical forces. Winged surfaces translate these depth oscillations into forward thrust. The result is a sawtooth trajectory that can propel the glider thousands of kilometers on a single battery charge. Slocum gliders, for instance, have participated in successful deployments ranging from whale vocalization studies to coastal physics and ecology (Dickey 2009). Such gliders will play a

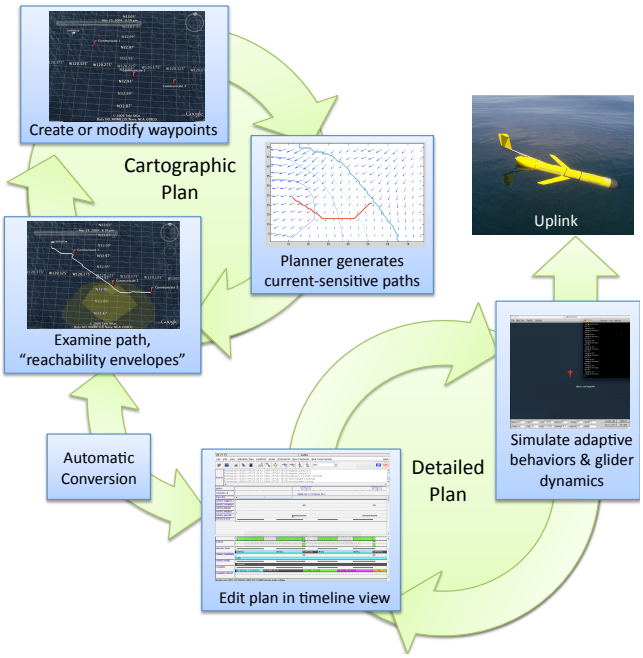


Figure 1: Planning workflow. The operator generates cartographic plans in a visual map interface. These are converted into detailed plans which can be edited in a timeline view and validated in a dynamic simulation. Glider image courtesy Rutgers University, map views courtesy Google, Inc.

significant role in the OOI by deploying instruments such as Conductivity, Temperature, and Depth (CTD) packages and advanced mission-specific sensors.

This work presents a strategy for Slocum Glider mission planning and execution developed as a technology integration prototype preceding the OOI construction. This system will support an Observing System Simulation Experiment (OSSE) to test cyberinfastructure capabilities using a simulated glider deployment to the mid-Atlantic bight. We will continue to refine the mission planning system to facilitate these experiments and eventual incorporation into the OOI Cyberinfastructure (Chave et al. 2009).

This problem area presents several new technological challenges. First, the gliders only provide sparse coverage of

the vast spatiotemporal volume under observation. Deployments to support numerical models must coordinating multiple assets to cover the most important areas (Dickey 2009). Second, downlinking data or receiving new commands requires that the glider stop other activities and surface to activate its satellite link. Missions typically limit gliders to sporadic communications at 3 to 8 hour intervals (Schofield et al. 2007). Between communications gliders must operate autonomously and respond appropriately to any unanticipated events and science data. Finally, planners must account for time-varying currents that can significantly affect navigation. These ocean currents can easily overwhelm the glider's own propulsion; accounting for current dynamics during path planning is essential for mission success and glider safety.

We begin by describing our general approach to scheduling and persecution. Mission planners use a two-stage process: they generate a coarse cartographic plan and then refine it into a detailed executable plan which respects vehicle dynamics. Later sections discuss each stage in greater detail. We present a new path planning algorithm that leverages model forecasts and computes optimal glider paths through time-varying current fields. Tests demonstrate significant improvements in path quality over interpolated waypoint methods currently in use. Finally, we describe progress on a planning and execution architecture for these glider mission plans, including a shoreside simulation of vehicle dynamics and an onboard architecture for robust execution.

## General Approach

The proposed planning strategy uses two levels of granularity called the *cartographic* and *detailed* representations (Figure 1).

The cartographic plan is a sequence of mutually-exclusive mission activities, each of which occupies a specific time interval and exercises exclusive control over the glider. In the first planning stage operators use a map-based interface to modify the cartographic representation and design timed paths between waypoint locations. This stage emphasizes science-relevant factors such as vehicle location, ocean current forecasts and overlays of remote sensing data. It abstracts from vehicle dynamics and related control parameters that can generally be determined automatically. An interface displays a map of the mission area along with the positions of waypoints and activities, and overlays of relevant model forecasts (Figure 2). We anticipate that users will interact mostly with this simplified cartographic plan, deferring computationally expensive and time-consuming validation until after the basic path has been established. During the cartographic phase a path planner determines waypoints' reachability and efficient connecting paths based on a simplified motion model and 48-hour current forecasts.

After operators are satisfied with the initial plan waypoint locations are fixed and an automatic translation expands the sequence into a complete *detailed* plan. The detailed plan adds specificity in the form of lower-level behaviors, several of which may be active simultaneously. The detailed planning phase tracks depletable resources and maximizes energy or time efficiency by altering factors such as glider

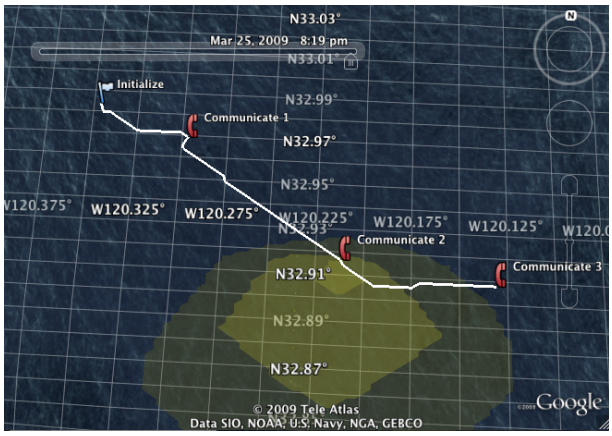


Figure 2: Cartographic plan with three waypoints (red) and optimal interpolated paths (thick white lines). Yellow polygons show current-based reachability envelopes for each hour after the second surfacing. Image courtesy Google Earth, Google, Inc.

pump displacement and depth. The two representations are not isomorphic - each cartographic plan corresponds to many possible detailed plans. Operators visualize and adjust the detailed plan in a timeline interface that tracks glider state variables, resources, and active behaviors (Figure 3).

Finally, operators can validate the detailed plan using a full dynamic simulation that considers the effect of adaptive control on vehicle performance. A copy of the onboard execution system runs the entire plan in a three-dimensional "virtual ocean." In our execution architecture an executive process commands the vehicle through a behavior-based control methodology using the MOOS-IvP architecture (Eickstedt et al. 2007). The executive activates and manages these low-level behaviors in the appropriate sequence while monitoring environmental conditions and vehicle safety. Our simulation currently uses a generic AUV control strategy with a simplified dynamic model but will eventually account for both vehicle dynamics and glider-specific adaptive behaviors. These control behaviors will respond realistically to currents that vary at different depths.

The dynamic simulation may suggest adjustments to specific activities. If significant changes are required operators can return to the cartographic representation in order to revise the basic path. When the simulation results are satisfactory the plan is ready for uplink to the glider. Onboard the vehicle, an identical copy of the executive and MOOS-IvP behavior stack can execute the final plan.

## The Cartographic Plan

This section details the initial planning step in which operators determine glider paths and surfacing locations. Operators examine the cartographic plan, along with remote sensing and forecast data, in an interface based on the Google

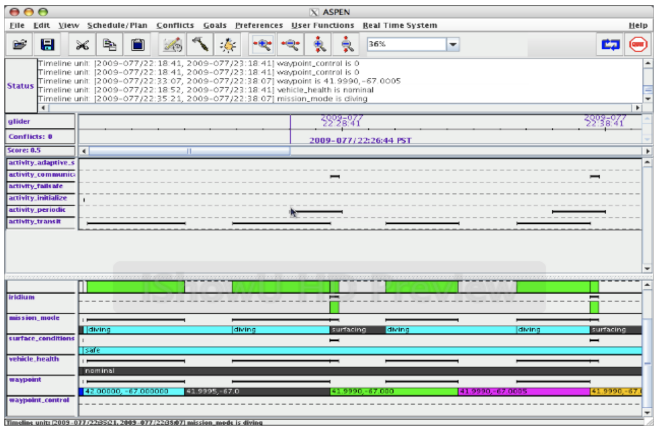


Figure 3: ASPEN timeline interface for the detailed planning step. Horizontal bars represent quantities such as resource use and vehicle state, as well as the activities in progress at each mission timestep.

Earth software package (Google Earth Software Package 2009). A simple "timeline slider" shows vehicle progress over time and the geographic location of mission activities and waypoints (Figure 2). The operator seeds path planning by selecting locations of waypoint goals and communications surfacings. A typical plan might include anywhere from zero to five such intermediate destinations in a three-hour communications cycle. The path planner computes travel times between these waypoints. It provides the interpolating path that arrives as early as possible at each operator-specified location, with an accuracy determined by the resolution of the planning grid.

Ocean currents change on an hourly basis, and current-sensitive path planning will be important for mission safety and science return. Fortunately ocean simulator models can predict currents on scales and accuracies that support glider path planning. These models exploit surface measurements, shore-based radar and remote sensing data to generate forecasts up to 24 or 48 hours in advance with kilometer-scale spatial resolution at up to 30 depth levels (Li et al. 2006; Wang et al. 2008). Our path planner uses travel time estimates based on current forecasts and a sawtooth glider trajectory with fixed minimum and maximum depths. Here we assume constant glider propulsion and ignore factors like custom flight parameters, bathymetry, salinity, and uncertainty in current predictions. The following sections will describe this calculation in greater detail.

More generally the path planner determines whether a sequence of waypoint locations and times is achievable by *any* path. It computes *reachability envelopes*: polygons showing the area that is reachable by the glider at future times (Figure 2). These isocontours summarize current constraints on feasible paths. Operators can shift waypoints within the reachability envelopes to provide additional margins of error or to maximize the travel distance; the system responds by updating the cartographic plan with the optimal interpolation.

## Path Planning: Background

Here we describe the planning algorithm that computes optimal glider paths in time-varying current fields. This system uses it to interpolate between waypoint locations but the algorithm itself is more general and permits destinations defined by arbitrary, and possibly disjoint, spatiotemporal volumes.

Several existing path planning algorithms account for current fields. Kruger et al. treat path planning as a nonlinear optimization problem with an objective function based on travel time and energy (Kruger et al. 2007). Pêtrès et al. demonstrate a general path planning approach for generating smooth trajectories against a static current field (Pêtrès et al. April 2007). Soulignac et al. describe a "sliding wavefront" algorithm that propagates costs within a polygonal lattice to account for strong but static current effects (Soulignac, Taillibert, and Rueher 2008). Others permit currents to vary over time, such as the trajectory optimization of Zhang et al (Zhang et al. 2008). Later work by Soulignac et al. allows for delayed departures, propagating path cost expressed as a function of departure time (Soulignac, Talilibert, and Rueher 2009).

In general these algorithms begin with fixed departure and goal locations, and choose a path to optimize an objective based on trip duration or energy. Our path planner deals with the more general case where the destination is defined in both space and time. Guaranteeing arrival time is important for glider operations since currents can change on an hourly basis. Moreover, scientists studying transient phenomena like salinity, temperature, or algal blooms (Leonard et al. 2007) will almost certainly wish to specify measurement times. Finally, timed destinations are important for coordinating multiple assets such as orbital overpasses or multiple gliders. Traditional algorithms that define destinations only in terms of a physical point location are inadequate since gliders that arrive early may not be able to hold position against currents.

We employ a novel "earliest valid arrival" criterion for path planning. Here the best path minimizes travel time to the goal region *provided that* the vehicle can then hold position until entering the destination volume. This ensures the glider will be present at the desired spatial and temporal location. This objective is amenable to an efficient wavefront propagation algorithm which finds the best existing solution up to the accuracy of path discretization.

## Path Planning: Algorithm

We discretize the glider's environment using a three-dimensional grid with two spatial dimensions indexed by $x$ and $y$, and one temporal dimension indexed by a *time interval* $\tau$. Each node $\mathcal{N}$ corresponds to a rectangular volume with an associated current prediction $\mathcal{C}$. The glider travels a path $\mathcal{P}$ given by a sequence of intermediate segments $\mathcal{P} = \{\mathcal{P}_i\}_{i=1}^{n}$. Valid segments travel between adjacent nodes in the 3D spatiotemporal grid.

Segment endpoints take specific continuous time values in their nodes' intervals. Therefore the glider is confined to the physical locations represented by grid points but can

move continuously along the temporal dimension. For example, an endpoint in $\mathcal{N}_k$ could have index $(x_k, y_k, t_k)$, for $t_k \in \tau_k$. We explicitly permit "position holding" activities that pause at the same physical location but move temporally from one time interval to the next. Figure 4 illustrates a simple four-segment path through the grid. The glider begins in the lower center and travels a path to the goal location represented by the orange node at top. The final leg of the trip consists of a position holding action; the glider waits at the same position until it enters the time interval of the destination node.

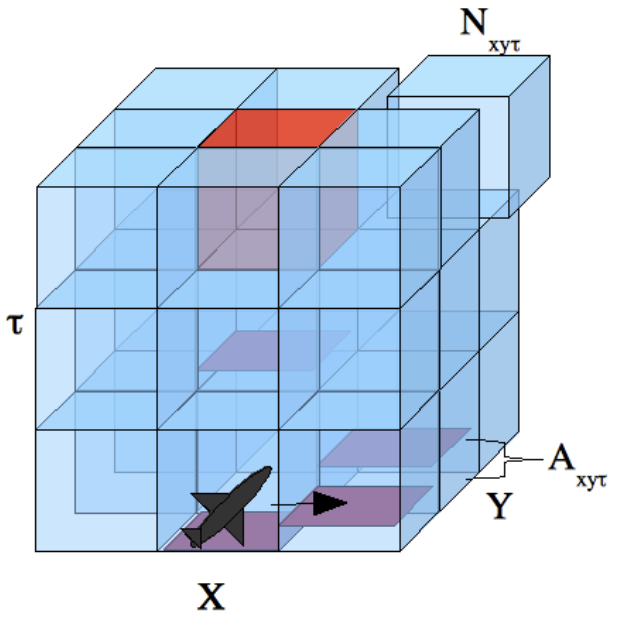


Figure 4: The 3D spatiotemporal grid. The glider begins in the lower center and travels a path that visits 4 nodes before arriving at the goal location (orange node at top). The red squares show $\mathcal{A}_{xyt}$: the best arrival time for each node. For clarity we omit arrival times for those nodes not participating in the optimal path.

The path planning algorithm computes $\mathcal{A} \in \mathbb{R}$ for each node; this is defined as the earliest possible time of arrival to location $(x, y)$ for the "hold position" action ending in time interval $\tau$. Often node a node directly following another at the same physical location will take its predecessor's value of $\mathcal{A}$ due to position-holding. For example, in Figure 4 the final segment consists of a hold position action. Physical locations have not changed between the final two nodes so their earliest arrival times are identical. We record only the single best arrival time for each node in the grid.

Our path planning strategy begins with the start position and searches for reachable neighbors in the grid. It expands a path to these children, and continues recursively expanding new nodes until it has accounted for all nodes in the grid (Algorithm 1). We retain paths corresponding to the earliest possible arrival time for each child, which requires storing a single candidate "best parent" for every node. Whenever we discover a new path to a node that arrives earlier than the current one, we overwrite the old entries for that node's parent and time of arrival $\mathcal{A}_{xy\tau}$. We expand new nodes in order of increasing arrival times which requires maintaining a sorted queue tantamount to a wavefront expansion (Soulignac, Taillibert, and Rueher 2008) or Dijkstra method. After completely expanding all the nodes in the grid one can trace a valid path from any goal node back to the start using the recorded parent nodes.

**Input**: Nodes $\mathcal{N}$, Current predictions $C$, Start location $\mathcal{N}_{\text{start}}$
**Output**: Grids of optimal parents $\mathcal{M}$ and expected arrivals

Initialize $\mathcal{A}_{\text{start}} \leftarrow 0$, all other $\mathcal{A} \leftarrow \inf$
Initialize wavefront queue $Q = \{\mathcal{N}_{\text{start}}\}$
**while** $Q$ *not empty* **do**
  find "parent node" $\mathcal{N}_p \in Q$ at $x_p, y_p, \tau_p$ minimizing $\mathcal{A}_p$
  $Q \leftarrow Q \setminus \mathcal{N}_p$
  define "hold position destination" $\mathcal{N}_h$ at $(x_p, y_p, \tau_p + 1)$
  **if** $\mathcal{N}_h$ *reachable from* $\mathcal{N}_p$ **then**
    $\mathcal{A}_h \leftarrow \mathcal{A}_p$
    $\mathcal{M}_h \leftarrow \mathcal{N}_p$
    $Q \leftarrow Q \cup \mathcal{N}_h$
  **foreach** $(x_c, y_c)$ *neighboring* $\mathcal{N}_p$ **do**
    **if** $(x_c, y_c)$ *reachable from* $\mathcal{N}_p$ **then**
      **if** $\mathcal{A}_p \notin \tau_p$ **then**
        $t_{\text{depart}} \leftarrow \min t : t \in \tau_p$
      **else**
        $t_{\text{depart}} \leftarrow \mathcal{A}_p$
      $t_{\text{arrival}} \leftarrow t_{\text{depart}} +$ travel time (from $\mathcal{C}$)
      $\tau_c = \tau : t_{\text{arrival}} \in \tau$
      "child" node $\mathcal{N}_c$ at $(x_c, y_c, \tau_c)$
      **if** $t_{\text{arrival}} < \mathcal{A}_c$ **then**
        $\mathcal{A}_c \leftarrow t_{\text{arrival}}$
        $\mathcal{M}_c \leftarrow \mathcal{N}_p$
        $Q \leftarrow Q \cup \mathcal{N}_c$
**return** $\mathcal{A}, \mathcal{M}$

**Algorithm 1**: Computation of arrival times and parent nodes satisfying the "earliest valid arrival" criterion. $Q$ is a priority queue of unexpanded nodes maintained in order of increasing arrival times. Subscripts $p, c, h$ indicate data associated with the current parent, child, and station-keeping destination nodes.

The retained paths optimize an "earliest valid arrival" criterion. The optimal path to a node reaches its physical destination as soon as possible provided that the vehicle can then hold position until the appropriate node's time interval. This allows operators to specify goals in terms of both physical locations and time intervals. A consequence of early arrival is preference for short and energy-efficient paths, as well as travel during early periods when current forecasts are most accurate and the most time remains to compensate for unforeseen developments.

Algorithms 1 and 2 detail the basic procedure. Algorithm

1 uses current predictions to produce a grid of arrival times $\mathcal{A}$ and path parents $\mathcal{M}$. It expands nodes in turn for activities corresponding to station holding and physical motion. For station keeping, the previous arrival times are preserved; for physical motion arrival times are computed using the earliest possible departure from the parent node. Note the use of common subscripts $p, c, h$ to indicate data associated with parent, child, and station-keeping destination nodes. After best parents have been computed for all reachable nodes, Algorithm 2 uses this information to generate paths to any desired destination node. A grid of approximately $100 \times 100 \times 100$ nodes requires about one minute of computation time on a modern desktop processor.

**Input**: Parents $\mathcal{M}$, Goals $\{\mathcal{N}_{\mathrm{g}}\}_{g=1}^{m}$, Start location $\mathcal{N}_{\mathrm{start}}$
**Output**: Path $\mathcal{P} = \{P_i\}_{i=1}^{n}$
$\mathcal{P} \leftarrow \{\mathcal{N}_{\mathrm{end}}\}$ for $\mathcal{N}_{\mathrm{end}}$ minimizing $\mathcal{A}_{\mathrm{end}}$
**while** $\mathcal{N}_{\mathrm{start}} \notin \mathcal{P}$ **do**
  $\mathcal{N}_{\mathrm{end}} \leftarrow M_{\mathrm{end}}$
  $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{N}_{\mathrm{end}}$
**return** $\mathcal{P}$

**Algorithm 2**: Computation of the optimal path from the arrays of parent nodes and arrival times.

Note that this algorithm retains only a single parent for each node and a single arrival time for each time interval. This means that it expands *only* the path corresponding to the earliest possible arrival at a particular node. In theory, discretization effects could make this suboptimal for the same reason that early arrival might fail more generally. For example, in the presence of extremely strong currents (when holding position is impossible) postponing arrival might be the only way to push future path segments into time intervals where currents are more favorable. However, simplicity of implementation and computational efficiency both favor the use of a single arrival time for each node. The designer should provide sufficiently fine temporal discretization to retain all the relevant paths. This is not difficult in practice, and temporal discretization effects were not apparent in any of the simulations we considered.

## Path Planning: Demonstration

We evaluated the path planning algorithm using a 48-hour forecast from the Regional Ocean Modeling System, or ROMS (Li et al. 2006; Wang et al. 2008). At this stage the motion model presumes depth-variable currents and a sawtooth trajectory between fixed minimum and maximum depths. We aggregate forecasted currents at all depths into an effective current velocity experienced by the glider passing through the node. In order to estimate travel time between neighboring nodes we compute a control propulsion that combines with the prevailing current to yield velocity vectors in the desired direction of travel. Specifically we define velocity 2-vectors corresponding to control propulsion $v_{control}$, a constant local current with velocity $C_{xy\tau}$, and a resulting net velocity $v_{actual}$. The net velocity has magnitude $\lambda$ and follows the desired direction of travel $d$. We have the relation:

$$v_{actual} = v_{control} + C_{xy\tau} = \lambda d \quad (1)$$

We compute a control to maximize the travel speed subject to a maximum propulsion magnitude $m$:

$$\lambda = \max |v_{control} + C_{xy\tau}|_2 \text{ S.T. } |v_{control}|_2 \leq m \quad (2)$$

A path between adjacent nodes is considered impossible if the glider would require longer than a single time interval to travel between them.

Figures 5 and 6 show two time steps drawn from a simulation in which the glider travels from lower right to upper left. The background vector field shows the predicted currents at both time steps. Colored blue and green lines show isocontours for the arrival time values $\mathcal{A}$. In Figure 5 the glider has progressed part way along the optimal path. Note that the glider's start location has become unreachable due to a surge in currents in the East side of the simulation area.

Figure 6 shows the solution path at the final simulation time step in which the glider has reached the goal location. A small area in the center of the map is still unreachable due to the exceptionally strong currents in this area. The glider cannot legally enter these gridsquares from an adjacent upstream location because the currents would push it beyond to the southwest before the end of that time interval.

We evaluated the path planner using a series of randomized trials. A virtual ocean testbed used a ROMS forecast for a 48-hour interval in March 2008. The simulation area covered a region spanning approximately 150km off the coast of Los Angeles, California. The simulation grid squares measured 500m along physical axes and 1 hour in the temporal dimension. Figures 5 and 6 demonstrate that the currents fluctuated significantly during this period; there were some regions of near calm and others where the current magnitude was greater than the propulsive force of the glider. We used a glider still-water travel speed of $0.3$ meters per second.

We ran a series of trials comparing the globally-optimal wavefront planning strategy to an alternative greedy approach using local current information. Each trial chose a random start location from the simulation region and a random destination within 10km of the start. We then computed connecting paths using both strategies. The wavefront-based path used algorithms 1 and 2 to satisfy the "earliest valid arrival" criterion. The local greedy path always traveled toward the neighboring grid square that was reachable and physically as close as possible to the goal. A planning run was deemed successful if it found a solution that reached the goal before the 48 hour forecast had elapsed.

Table 1 shows simulation results for different start/goal separation distances: less than $5km$ (561 trials), 5 to $10km$ (1651 trials), and greater than $10km$ (683 trials). Often the destination location was blocked by strong currents and neither method found a valid path. At other times both methods succeeded. Occasionally the destination was only accessible by exploiting time-varying currents; the wavefront path planner succeeded while the greedy approach failed. The path planner never underperformed the greedy method with respect to either path validity or arrival time. The top two rows of the table show the fraction of successful runs.

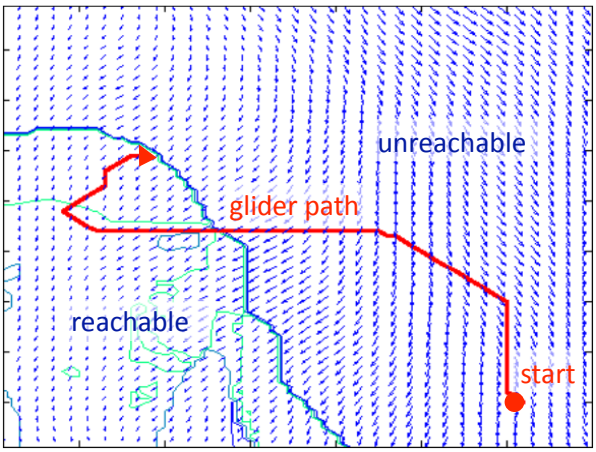


Figure 5: Midway through a simulated run the glider has planned a winding path to exploit currents. A red line shows the glider's path, while the background vector field shows current predictions at this timestep. Blue and green isocontours show reachability at representative times after the simulation start.

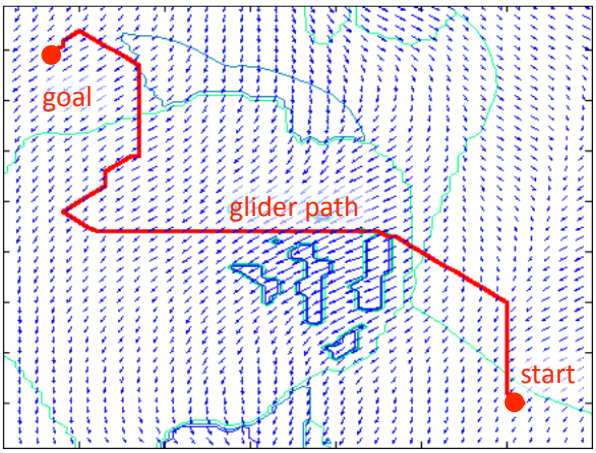


Figure 6: At the end of the simulation currents have changed significantly while the glider has reached the goal location.

| | **0-5km** | **5-10km** | **>10km** |
|---|---|---|---|
| **Wavefront** | 53.3% | 50.7% | 46.0% |
| **Greedy** | 33.7% | 33.1% | 26.1% |
| **Time difference** | $0.09h$ | $0.38h$ | $0.88h$ |

Table 1: Percentage of trials for which greedy and global planning methods find valid paths. Columns show performance based on distance between the random start and goal locations. All differences are significant to p < 1e-9. The bottom row shows the average hours of improvement in arrival time due to wavefront path planning.

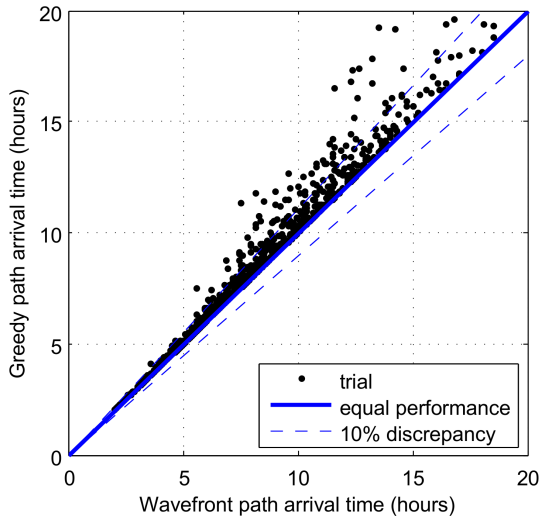


Figure 7: Path length (in hours) for trials in which both greedy and wavefront planning strategies return valid paths. This chart includes only those trials for which both methods find successful paths.

The wavefront method succeeds more often than the greedy approach for all path lengths (significant to p < 1e-9 for Fisher's exact test).

Naturally paths that optimize the "earliest valid arrival" criterion reach the physical goal location faster than the greedy paths. The final row of the table in Figure 1 shows the difference in travel times (in hours) for trials when both methods find a valid plan. The mean over all trials is shown, with standard deviations in parenthesis. Here the wavefront method outperforms; it can make opportunistic detours to exploit currents and arrive faster. Figure 7 shows a visual comparison of travel times. The arrival time improvement due to wavefront planning is more pronounced for longer paths; wavefront paths arrive strictly earlier for all trials lasting longer than 16 hours. Note that this chart understates the wavefront method's benefit somewhat since it excludes the trials for which the greedy approach fails to find a valid path.

These simulations corroborate experience that currents dominate the glider trajectory. They underscore the importance of considering currents in mission planning, and suggest that the "earliest valid arrival" criterion could provide significant benefits.

## The Detailed Plan

After the operators find a satisfactory path they convert the activity sequence to a detailed plan that contains all the information required for execution onboard the vehicle. The detailed plan adds additional fidelity by tracking energy resources, vehicle and instrument health. In addition it specifies control parameters such as the depth of each pump cycle and the pitch of glider control surfaces. It permits simultaneous activities and fine control of specific payload instru-

| Model Element | Cartographic | Detailed |
|---|---|---|
| Goto Waypoint Activity | | |
| destination | x | x |
| fallback destination | x | x |
| start time, duration | x | x |
| energy required | | x |
| depth | | x |
| pitch | | x |
| Communicate Activity | | |
| location | x | |
| start time, duration | x | x |
| Initialize Activity | | |
| location | x | x |
| start time, duration | x | x |
| Loiter Activity | | |
| start time, duration | | x |
| Sample Activity | | |
| start time, duration | | x |
| Failsafe Activity | | |
| start time, duration | | x |
| Vehicle Health | | x |
| Payload Health | | x |
| Battery Energy | | x |

Figure 8: Model elements in Cartographic and Detailed plans.

ments. Energy use is calculated using user-specified glider flight parameters and the nominal travel time estimates provided by the path planner.

Examples of activities include:

- **Initialize**: Reset glider systems and behaviors to begin the new plan.
- **Go to waypoint**: Travel toward a specified latitude and longitude position. Alternatively, travel toward a fallback waypoint after failing a start condition such as a desired position, keepout zone, sensor reading or glider state.
- **Communicate**: Rise to the surface and activate the satellite link to receive new instructions from the shore. Note that communications surfacings are given a location in the Cartographic view so that they appear in the map visualization. These locations are stripped from the detailed plan.
- **Loiter**: Drift passively on the current.
- **Sample**: Activate the CTD instrument package to record measurements.
- **Failsafe**: When a serious hardware fault is detected, surface to contact shore while permanently ceding vehicle control to the survival behaviors.

Neither the cartographic path planner nor the detailed planning stage explicitly models execution uncertainty. However, conditional "go to waypoint" activities do add robustness in the form of "keepout zones" or fallback destinations for timed waypoint sequences. The onboard executive continually reestimates vehicle position during plan execution. If unanticipated currents push a glider off-course then it chooses the appropriate result for these conditional activities and thus directs the glider to safe areas. Note that position is not tracked explicitly in the detailed plan so the plan remains formally *valid* regardless of the glider's physical position. This formulation places path design entirely in the cartographic planner. It frees the detailed planning step (and eventually the onboard executive) from considering complex spatial dependencies like path connectivity. In any case the onboard executive lacks up-to-date forecasts from shore so it will not have any new information that would justify more elaborate path revisions.

Our scheduling engine for manipulating detailed plans is the ASPEN / CASPER software suite (Chien et al. 2000). The ASPEN planner uses a formal model of glider state and resources to validate the activity sequence and optimize schedules. The model tracks depletable resources such as battery energy, availability of the vehicle helm and communications equipment, and vehicle state variables such as the health of the system and instruments. A timeline-based user interface allows operators to alter any aspect of the plan and immediately see its impact on the mission. (Figure 3). Any resource conflicts that appear (for example, a depleted battery) will be flagged at this stage. The operator can respond by reparameterizing the existing paths with alternative behaviors, or returning to the cartographic interface to edit plan waypoint locations.

CASPER is the real-time counterpart of ASPEN; it executes plans and monitors the system state to ensure consistency with the schedule. Our onboard architecture uses CASPER as an executive to command a behavior-based control system. The CASPER executive activates, manages and concludes each of the activities in the sequence, decomposing them into parameters for the low-level behaviors.

The behavior-based control utilizes the MOOS-IvP system developed at MIT CSAIL (Benjamin et al. 2009; Eickstedt et al. 2007). MOOS-IvP provides a subscription-based message passing service for interprocess communication onboard the vehicle. Its software modules include processes for state estimation and telemetry. Additionally, a helm process supplies behaviors for basic functions such as surfacing, forward travel, and waypoint following. The helm mediates between active behaviors by optimizing a piecewise linear objective function; it produces vehicle control commands that are passed to hardware-specific glider processes. During execution the CASPER executive modulates the helm behaviors, activating or deactivating them and updating their parameters in accordance with the schedule and vehicle state.

Operators can validate the detailed plan by simulating the entire mission within a copy of the MOOS-IvP environment. Here a simulator module takes the place of the physical glider. Operators can manually vary environmental conditions during execution; for example, one could see how additional unforeseen currents might affect the resulting glider path. The execution stage currently uses simplified AUV behaviors and dynamics, but it will eventually incorporate glider-specific models.

## Future Work

Future work will automate the feedback from the dynamic simulation to earlier planning steps. Currently any anomalies detected in dynamic simulations must be corrected manually in the plan. For example, travel times may differ from expectations due to current prediction errors or simplifications in the cartographic motion model. Future work will develop a protocol to propagate these conditions back to earlier planning stages. An off-nominal plan segment defines new constraints in the form of exceptions to path segment time estimates. These exceptions can alter travel time calculations for the next round of path planning.

The methods described in this work are part of an ongoing effort to design and test a CyberInfastructure (CI) to support OOI operations in the next decade (Chave et al. 2009). We have implemented the prototype planning system and will continue to develop and refine it during the coming year. The system will support the upcoming Observing System Simulation Experiment (OSSE) to test CyberInfastructure capabilities with a simulated glider deployment. The experiment will use historical forecasts for which followup measurements are available. This provides a ground-truth standard to compare different mission planning methods.

Our immediate goal is to facilitate the OSSE experiment. To this end we will incorporate a glider model into the dynamic simulation, along with glider-specific adaptive behaviors. We will develop a translation mechanism that will convert glider control parameters from MOOS-IvP behaviors into native commands for an unmodified glider platform. This will allow physical gliders to operate from the current-sensitive plans generated in advance by the simulation. Later stages of development will port the execution architectures to the physical glider, permitting true autonomous plan persecution and monitoring within the OOI network.

## Acknowledgments

The research described in this paper was sponsored by the National Science Foundation Ocean Observatories Initiative, managed at the University of California, San Diego. Experiments were carried out at the Jet Propulsion Laboratory, California Institute of Technology. 

## References

Benjamin, M. R.; Leonard, J. J.; Schmidt, H.; and Newman, P. M. 2009. An Overview of MOOS-IvP and a Brief Users Guide to the IvP Helm Autonomy Software. *MIT CSAIL Tech. Report, MIT-CSAIL-TR-2009-028*.

Chave, A.; Arrott, M.; Farcas, C.; Farcas, E.; Krueger, I.; Meisinger, M.; Orcutt, J.; Vernon, F.; Peach, C.; Schofield, O.; and Kleinert, J. 2009. Cyberinfrastructure for the US Ocean Observatories Initiative: Enabling Interactive Observation in the Ocean. In *Proceedings of OCEANS*.

Chien, S.; Rabideau, G.; Knight, R.; Sherwood, R.; Engelhardt, B.; Mutz, D.; Estlin, T.; Smith, B.; Fisher, F.; Barrett, T.; Stebbins, G.; and Tran, D. 2000. ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling. *Intl. Conf. on Space Operations (SpaceOps)*.

Chien, S.; Sherwood, R.; Tran, D.; Cichy, B.; Rabideau, G.; Castano, R.; Davies, A.; Mandl, D.; Frye, S.; Trout, B.; et al. 2005. Using autonomy flight software to improve science return on Earth Observing One. *Journal of Aerospace Computing, Information, and Communication* 2(4):196–216.

Dickey, T. 2009. Progress in Multi-disciplinary Sensing of the 4-Dimensional Ocean. In *Proceedings of SPIE*.

Eickstedt, D.; Benjamin, M.; Wang, D.; Curcio, J.; and Schmidt, H. 2007. Behavior Based Adaptive Control for Autonomous Oceanographic Sampling. *ICRA*.

Given, H.; Banahan, S.; and Williams, S. 2007. The US National Science Foundation's Ocean Observatories Initiative at Preliminary Design. *Oceans 2007* 1–8.

Google Earth Software Package. 2009. *Google, Inc. http://earth.google.com*.

Kruger, D.; Stolkin, R.; Blum, A.; and Briganti, J. 2007. Optimal auv path planning for extended missions in complex, fast-flowing estuarine environments. *International Conference on Robotics and Automation*.

Leonard, N. E.; Paley, D. A.; Lekien, F.; Sepulchre, R.; Fratantoni, D. M.; ; and Davis, R. 2007. Collective motion, sensor networks, and ocean sampling. *Proc. IEEE* 95(1).

Li, P.; Chao, Y.; Vu, Q.; Li, Z.; Farrara, J.; Zhang, H.; and Wang, X. 2006. Ourocean: An integrated solution to ocean monitoring and forecasting. *OCEANS* 1–6.

Morris, R.; Oza, N.; Keely, L.; and Kurklu, E. 2008. Learning to Improve Earth Observation Flight Planning. *AAAI Conference on Artificial Intelligence*.

Pêtrès, C.; Pailhas, Y.; Patrón, P.; Petillot, Y.; Evans, J.; and Lane, D. April 2007. Path planning for autonomous underwater vehicles. *IEEE Transactions on Robotics* 23(2):331—341.

Schofield, O.; Kohut, J.; Aragon, D.; Creed, L.; Graver, J.; Haldeman, C.; Kerfoot, J.; Roarty, H.; Jones, C.; Webb, D.; and Glenn, S. 2007. Slocum gliders: Robust and ready. *Journal of Field Robotics* 24:6:473–485.

Soulignac, M.; Taillibert, P.; and Rueher, M. 2008. Adapting the wavefront expansion in presence of strong currents. *International Conference on Robotics and Automation*.

Soulignac, M.; Talilibert, P.; and Rueher, M. 2009. Time-minimal path planning in dynamic current fields. *International Conference on Robotics and Automation*.

Wang, X.; Chao, Y.; Dong, C.; Farrara, J.; Li, Z.; McWilliams, J. C.; Paduan, J. D.; and Rosenfeld, L. K. 2008. *Modeling tides in Monterey Bay, California*. Elsevier.

Zhang, W.; Inanc, T.; Ober-Blöbaum, S.; and Mardesn, J. E. 2008. Optimal trajectory generation for a glider in time-varying 2d ocean flows b-spline model. *International Conference on Robotics and Automation*.