Planning Efficient Paths through Dynamic Flow Fields in Real World Domains

Christopher Tomaszewski, The Robotics Institute, Carnegie Mellon University

This research addresses the problem of planning efficient paths for agents through flow fields in small real-world domains where vehicle dynamics and environmental uncertainty can significantly affect the optimality of a path. The objective of this project is to develop a method that can account for vehicle dynamics during motion planning in an efficient and reusable manner that does not incur a prohibitive computational cost when replanning. In particular, this work examines the feasibility of using forward expansion techniques to generate a reachable interface, and then level set methods to evolve this interface forward over time. To validate the effectiveness of the planner developed, it will be deployed on the Cooperative Robotic Watercraft platform developed by the Robotics Institute, equipped with a Nortek AD2CP acoustic doppler current profiler, and the results will be compared against the performance of current state of the art planning techniques.

1. INTRODUCTION

The problem of planning efficient routes for agents is of great importance across domains in mobile robotics, where energy is at a premium. In domains involving movement through fluids, which themselves may be moving, such as flight through various wind patterns or navigation through river currents, it is understandably desirable for algorithms to harness the available kinetic energy in the environment to decrease a robots energy consumption. Successful implementation of such techniques in turn facilitates new robotic applications involving extended deployment range and duration (e.g. autonomous environmental monitoring of long stretches of river).

This particular domain has motivated much research in the past and many planning algorithms have been adapted or developed to solve this problem. In their work, Garau, Alvarez, and Oliver assume constant thrust navigation and use the A* algorithm to compute the optimal paths for a set of different current and eddy distributions [Garau et al. 2005]. Warren has proposed a method based on artificial potential fields, which is less susceptible to local minima but requires most of the global workspace to be known and does not account for variabilities in the flow [Warren 1990]. A method using the forward evolution of level sets that can produce time optimal paths assuming a constant vehicle velocity through time dependent flow fields was developed by Lolla et al [Lolla et al. 2012]. In their work on autonomous underwater gliders, Davis, Naomi, and Fratantoni develop a variational calculus approach for path planning through time invariant velocity fields of comparable magnitude to the operating velocities of long range gliders [Davis et al. 2009].

Many of the proposed algorithms make simplifying assumptions regarding the vehicle dynamics and oftentimes operate with at least some model for the environment. Although this makes for efficient performance in simulation, in reality measuring the rate of a moving fluid while moving through that fluid is difficult and developing a detailed environmental model beforehand is impractical and not robust to unexpected events. Further difficulty comes from the inherent uncertainty of state estimation, as even small differences in a robots perceived state can yield very different optimal paths to the goal state, as evident in Figure 1 below. This figure shows the time-optimal paths for a boat travelling at various fixed velocities upstream against a parabolic current distribution in a river.

In order to limit the scope of the experiments to be done, this research focuses on planning paths through river domains for use with small autonomous airboats. This relative size of the domains under scrutiny with respect to the vehicles is such that

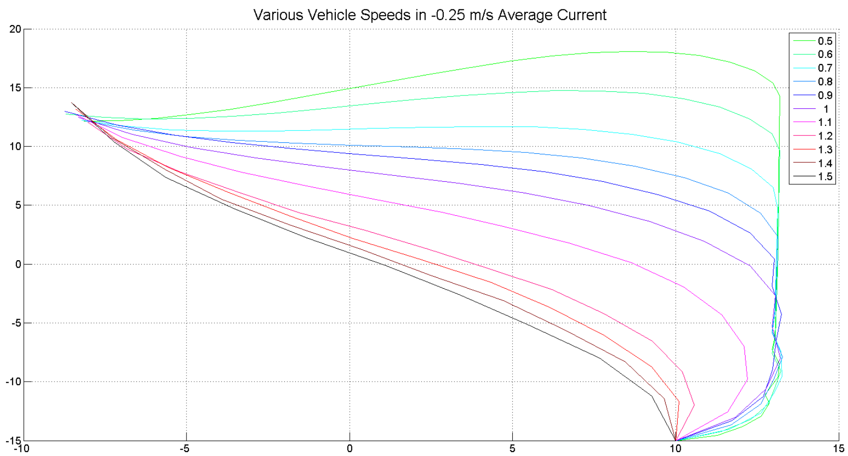

Fig. 1. Optimal Constant Velocity Paths to Upstream Waypoint Assuming Parabolic Current Distribution

the dynamics of the agents have significant bearing on the optimal path through the current field. A more detailed description of the airboat platform and domain representation is discussed in the following section.

2. METHODOLOGY

2.1. Cooperative Robotic Watercraft Platform

The testing platform chosen for this research is the Lutra 1.0 series of the Cooperative Robotic Watercraft system developed at the Robotics Institute of Carnegie Mellon University. The platform, pictured in figure 2, is comprised of a vaccuum-formed ABS hull filled with expanding polyurethane foam, a panning ducted fan assembly for propulsion, and an electronics compartment, which houses a smartphone, a custom embedded board, and batteries. The characteristics of the system make it an ideal candidate for the deployment of the efficient algorithms this research seeks to produce; as a low-cost autonomous monitoring system, the CRW platform carries a limited energy supply, making it desirable to optimize navigation and sampling procedures as to maximize the information that can be gathered before recharging is necessary. A more detailed treatment of the platform can be found in our previous publications [Scerri et al. 2012; Valada et al. 2012b; Valada et al. 2012a].

In order to take into account the vehicle dynamics, it was necessary to develop a rough dynamic model for the system. The airboats were measured under controlled conditions in the lab and deployed at Panther Hollow Lake in order to characterize the system parameters. The resulting dynamic model is of the form show below:

$$m\ddot{\vec{x}} = -\vec{b}\dot{\vec{x}}^2 + \vec{F}$$
$$J\ddot{\theta} = -b_\theta\dot{\theta}^2 + ||F||L\sin\phi$$

where m is the mass of the vehicle, $\vec{b}$ is the vector of linear drag coefficients, $\vec{F}$ is the force vector applied by the propulsion assembly, J is the moment of inertia of the vehicle, b_θ is the torsional drag coefficient, L is the moment arm of the vehicle, and ϕ is the angle of the propulsion assembly. The empirically derived values for these parameters are provided within the included simulator code.

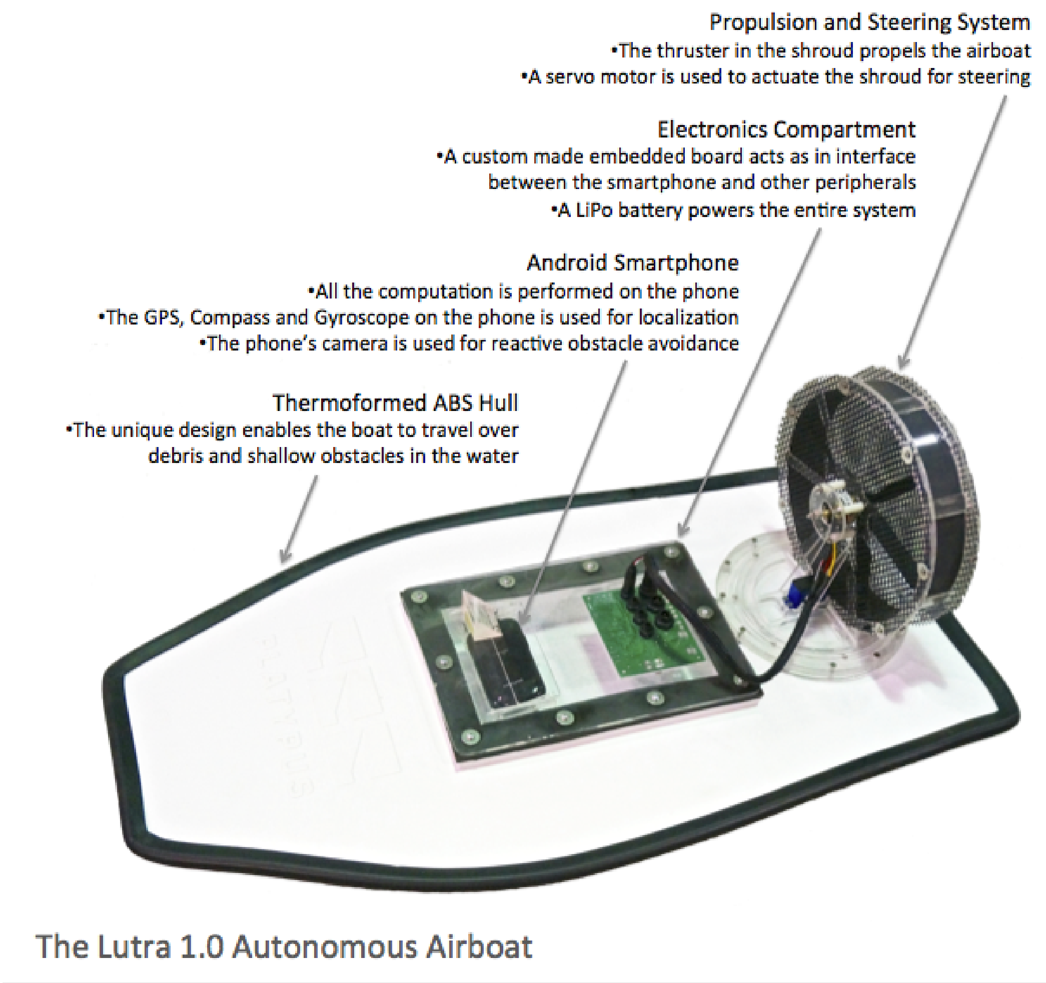

Fig. 2. A Complete CRW Airboat

Figure 3 shows the results of using these dynamics equations to simulate the motion of an airboat under constant thrust at various angles for a period of 5 seconds. The green vectors represent the resultant velocity of the boat at the end of the simulation period, while the red vectors indicate the heading of the boat at the same time.

In figure 4, we present a curve depicting the relationship between the static thrust provided by the propulsion assembly and the current draw measured from the battery. Although we do not model energy usage in our experiments, the nonlinear relationship we see here raises several interesting questions for future work regarding how to best vary the vehicle's thrust for optimal battery life. We can imagine that it may be advantageous to thrust harder as the boat passes through regions with unhelpful currents, or cut power and float when the current is moving towards the goal. Some of these scenarios will be tested during field tests this summer.

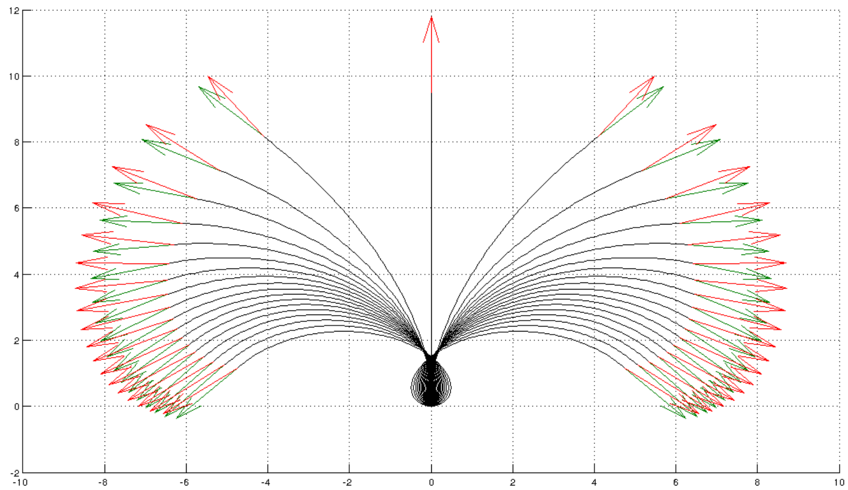

Fig. 3. 5 Second Forward Expansion Using Dynamic Model and Constant Control Signals

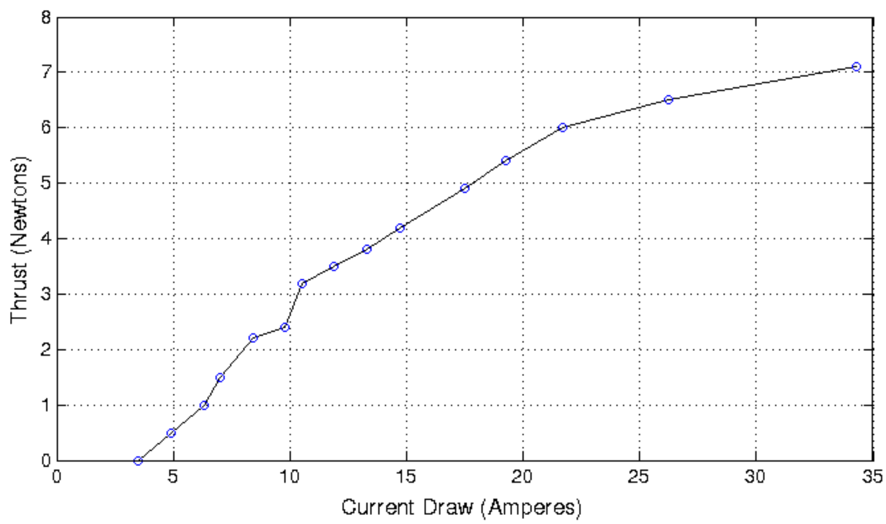

Fig. 4. Thrust Output vs. Current Draw of CRW Platform

2.2. Level Set Evolution

The level set method described by Lolla et al. in their paper is a two phase algorithm where the first part involves propagating an interface (denoted as the zero level set of the signed distance function ϕ) forward in time, and the second part entails tracking a particle back from the goal state along the normal vectors to the intermediate interfaces [Lolla et al. 2012]. The interface being tracked actually represents the farthest set of points reachable by the vehicle at any point in time, under some fairly strict assumptions. The process modeling the evolution of this interface is constructed as a

Hamilton Jacobi partial differential equation, which is then solved discretely over a grid. The general form of this PDE is given below.

$$
\begin{aligned}
0 =& \frac{\delta\phi(\vec{x},t)}{\delta t} && (1)\\
&+ F|\nabla\phi(\vec{x},t)| && (2)\\
&+ V(\vec{x},t)\cdot\nabla\phi(\vec{x},t) && (3)\\
&- b(\vec{x},t)\kappa(\vec{x},t)|\nabla\phi(\vec{x},t)| && (4)\\
&+ \lambda(\vec{x},t)\phi(\vec{x},t) && (5)\\
&+ H(\vec{x},t,\phi(\vec{x},t),\nabla\phi(\vec{x},t)) && (6)
\end{aligned}
$$

Lines 1 and 6 make up the general form of the hamilton jacobian partial differential equation, however, these general form equations are typically more difficult to solve. Therefore, there are a few terms that are sometimes added, which are specifically constrained and can be more easily solved. In their algorithm, Lolla et al use only the terms on lines 1, 2, and 3 to represent the farther points reachable by some vehicle in a given amount of time. Line 2 represents motion normal to the level set at a constant speed F, while line 3 represents motion due to the external velocity field. In using this representation, Lolla et al make the assumption that the vehicle is always traveling at speed F normal to the level sets. Furthermore, their method relies on having exact knowledge of the velocity field V and how it will change over time. Clearly these are not the best assumptions to make for real world implementation, which further motivates our objective of producing an algorithm that calculates paths more true to the dynamics of the vehicle.

After the interface has evolved far enough that the goal state is within the signed distance function, the forward evolution terminates and a particle is traced back along the normals to the intermediate instances of the tracked interface. Lolla et al prove in their work that if a vehicle always travels at speed F in a normal direction to the set of its currently reachable states, and the external velocity field $V(\vec{x},t)$ is known perfectly for all times, then the path computed by this algorithm is time-optimal.

In our level set experiments, we made use of Ian Mitchell's Matlab Level Sets Toolbox, which implements several partial differential equation solvers [Mitchell 2008]. For consistency we chose to use a single domain for all of our simulations - a river 30 meters across, with a current distribution modelled after fully developed pipe flow. Figure 5 illustrates this distribution - essentially the speed of the current is twice the average velocity at the center of the river and then drops off to zero as you get closer to the banks. For our purposes, we included a two meter buffer by each bank that the boat should avoid. Initially the method of forbidden regions described by Lolla et all in their work was implemented, where the external velocity field is zeroed out and the normal velocity of the vehicle is held at zero within these regions. Unfortunately when such a discontinuous change in velocities was introduced, we experienced numerical instabilities while solving the partial differential equations; after the level set evolution reached the discontinuity, the interface would sometimes develop unstable distortions that would magnify during propagation and produce inaccurate paths.

In order to solve this problem, we treat the external velocity field as a cost function and exploit its lower dimensionality to eliminate the discontinuities while retaining the guarantee that the optimal path will never lie within the forbidden regions. Since our domain description assumes the river has the same parabolic current distribution along its course, we can define the cost function to be proportional to the velocity of the river. When the goal state is upstream from the vehicle, the cost function is postive because the agent must fight against it, and when the goal is downstream the cost

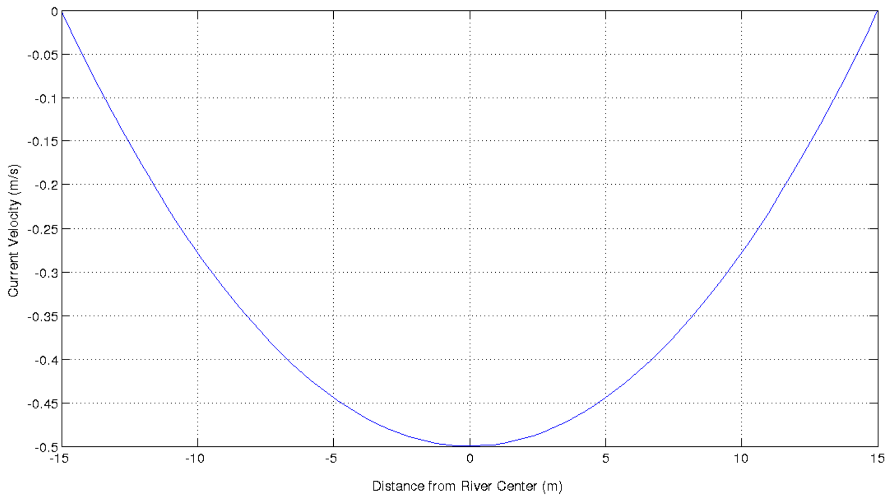

Fig. 5. Parabolic Current Distribution with -0.25 m/s Average Speed

function is negative because the boat is moved towards the goal by the current. Using the definition, we can exploit the lower dimensionality of the cost function (i.e. the fact that for a given X coordinate, the cost function does not vary as you move up and down stream) and apply the theorems described by Vernaza and Lee [Vernaza and Lee 2012]. They state that when the cost function does not vary in some dimension, then the optimal path to the goal will never move away from the goal in that dimension. Therefore, it must be that in our particular domain, the boat will never head downstream when the goal is upstream and vice versa. If we consider the forbidden regions again, and simply clamp the value of the external velocity field in these regions to the value at the boundary of the region, our cost function no longer varies in the X direction in these regions and therefore the optimal path will never head away from the goal along the X axis in these regions as well. If we define the goal state as always lying outside of a forbidden region, this guarantees that the optimal path will never cause the boat to head into the forbidden regions, which we defined on the banks of the river. Now we have removed our discontinuities and should be able to evolve our level sets like normal with just a modification to our external velocity field. Sure enough, the paths produced with this modification never head into the forbidden regions at the banks and the partial differential equation solver produces more stable results.

Figure 6 shows one example of a level set evolution and the path it yields using this method. The boat starts at the top left waypoint and heads to the bottom right waypoint. Notice that the level sets can evolve out past the banks of the river (-15 and 15), but the backward propagation along the normals does not guide the vehicle into the banks. The small blue lines along the path indicate the heading of the boat, while the path itself indicates the resultant motion of the vehicle (in this case the current is helping the boat in the vertical axis so the boat is dedicating more of its thrust to move along the horizontal axis).

Figures 7 and 8 show the different time-optimal paths generated when one varies the average current velocity and goal states respectively, while hold all other parameters constant. The former figure is very interesting, because there is a large common component that many of the paths share (driving upwards along the right bank). If an expression could be developed for the length of this component in terms of the aver-

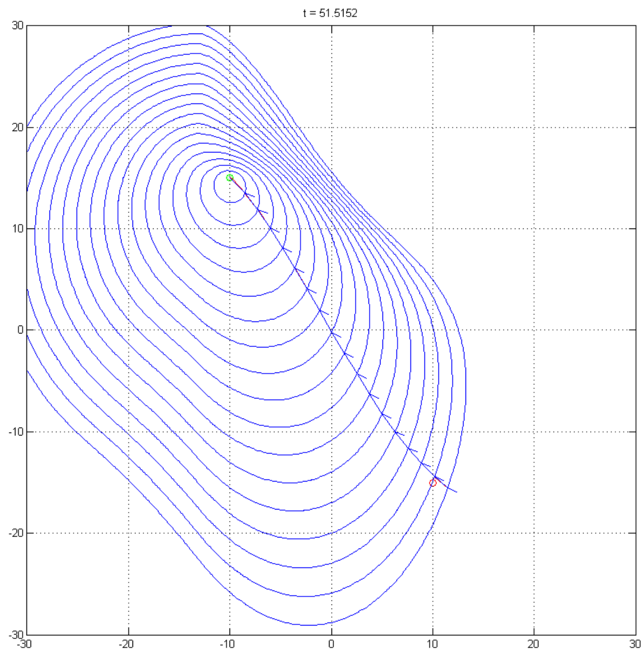

Fig. 6. Example Output from Level Set Evolution Planner

age current velocity, this could be the basis for a fast and robust algorithm that may approximate the optimal solution but can easily be adjusted for changes in currents or faultly environmental models. In the latter figure, there is an obvious phase transition where the optimal path switches which bank it follows. From simulation of a variety of combinations of these scenarios, it seems that this transition occurs when the goal state is further from the center of the river than the starting state. Although this is likely tied to the current distribution function used, in our application this can be a useful heuristic for guiding exploration or quickly determining which direction to head towards.

2.3. Forward Expansion

The idea behind this component of the project was to investigate methods for efficiently simulating the dynamics of a vehicle and generating an interface to track within the level set evolution. Unfortunately, because the control signals are continuous, there is an extremely high branching factor when considering all possible control signals. Without any modifications, it was computationally intractable to compute the dense branches of the search tree more than a few levels deep. Therefore, the search tree must be pruned to allow for deeper expansion. Figures 9 and 10 show two pruned search trees 3 levels deep. In the former case only every 3rd control angle is considered and then only every 3rd state is expanded further. In the latter, only every 5th angle is considered and every 5th resulting state is expanded. We explored some alternative pruning techniques such as penalizing large variations in propulsion assembly angle, and these seem to further decrease the size of the search space.

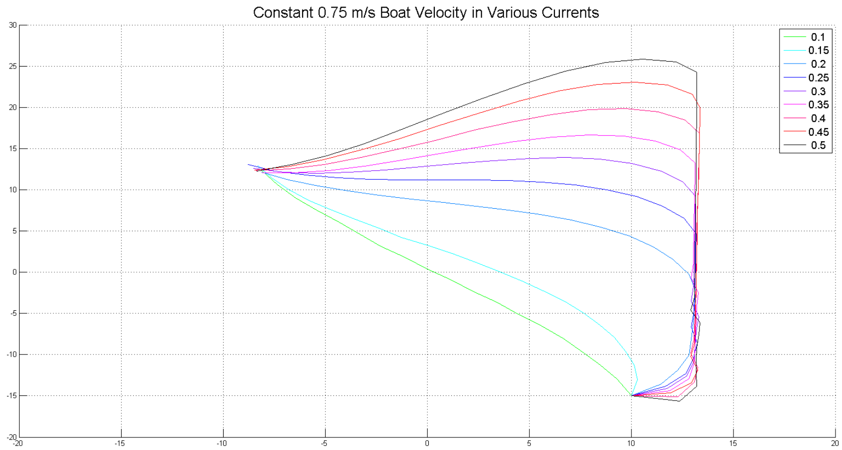

Fig. 7. Optimal Paths as Currents Vary

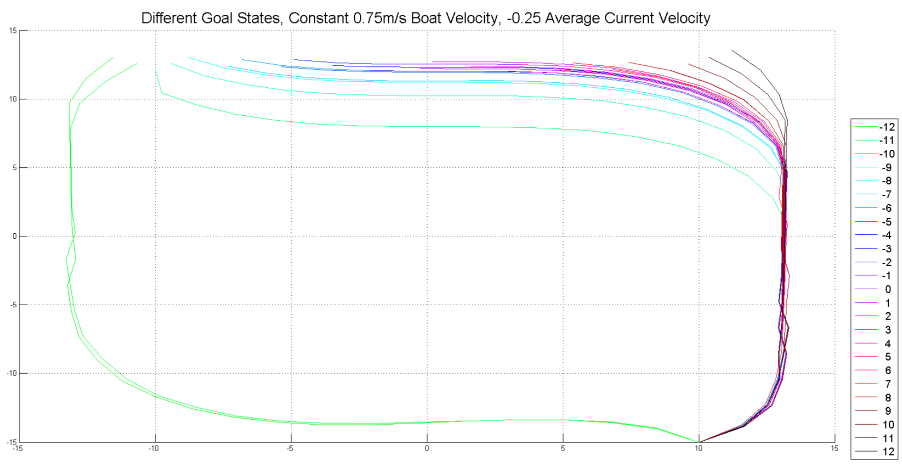

Fig. 8. Optimal Paths as Goal State Varies

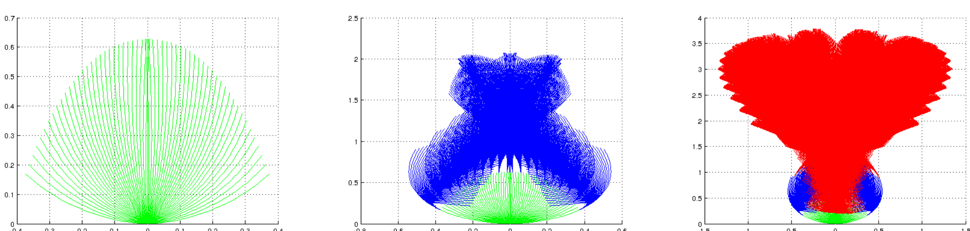

Fig. 9. Three 1 Second Iterations, 3 Degree Increments, Expand Every 3rd Branch

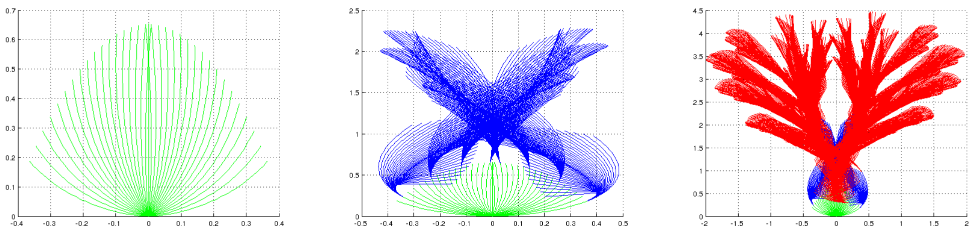

Fig. 10. Three 1 Second Iterations, 5 Degree Increments, Expand Every 5th Branch

3. CONCLUSIONS

Although we have yet to integrate the forward expansion techniques with the level set evolution algorithm, we have thusfar developed a dynamic model for our platform, identified methods to utilize the lower-dimensionality of our current distribution in order to produce more numerically stable results when solving the Hamilton Jacobi partial differential equations, and identified several techniques to prune our search tree when sampling random control signals. Future work includes the development of further search tree pruning techniques, contour extraction from the forward simulation, and appropriate level set propagation while preseving the dynamics of the system. We would also like to develop some more realistic models for current flow in a river using data gathered from the Nortek AD2CP current profiler that was awarded to use as part of a grant in March.

REFERENCES

Russ E. Davis, Naomi E. Leonard, and David M. Fratantoni. 2009. Routing strategies for underwater gliders. *Deep Sea Research Part II: Topical Studies in Oceanography* 56, 35 (2009), 173 – 187. DOI:http://dx.doi.org/10.1016/j.dsr2.2008.08.005 ¡ce:title¿AOSN II: The Science and Technology of an Autonomous Ocean Sampling Network¡/ce:title¿.

B. Garau, A. Alvarez, and G. Oliver. 2005. Path Planning of Autonomous Underwater Vehicles in Current Fields with Complex Spatial Variability: an A* Approach. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. 194–198. DOI:http://dx.doi.org/10.1109/ROBOT.2005.1570118

T. Lolla, M. P. Ueckermann, K. Yigit, Jr. Haley, P.J., and P. F J Lermusiaux. 2012. Path planning in time dependent flow fields using level set methods. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. 166–173. DOI:http://dx.doi.org/10.1109/ICRA.2012.6225364

IanM. Mitchell. 2008. The Flexible, Extensible and Efficient Toolbox of Level Set Methods. *Journal of Scientific Computing* 35, 2-3 (2008), 300–329. DOI:http://dx.doi.org/10.1007/s10915-007-9174-4

Paul Scerri, Prasanna Velagapudi, Balajee Kannan, Abhinav Valada, Christopher Tomaszewski, John M Dolan, Adrian Scerri, Kumar Shaurya Shankar, Luis Lorenzo Bill-Clark, and George A Kantor. 2012. Real-World Testing of a Multi-Robot Team. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Padgham Conitzer, Winikoff and van der Hoek (Eds.). International Foundation for Autonomous Agents and Multiagent Systems, www.ifaamas.org.

Abhinav Valada, Christopher Tomaszewski, Balajee Kannan, Prasanna Velagapudi, George A Kantor, and Paul Scerri. 2012a. An Intelligent Approach to Hysteresis Compensation while Sampling using a Fleet of Autonomous Watercraft. In *The 2012 International Conference on Intelligent Robotics and Applications (ICIRA 2012)*.

Abhinav Valada, Prasanna Velagapudi, Balajee Kannan, Christopher Tomaszewski, George A Kantor, and Paul Scerri. 2012b. Development of a Low Cost Multi-Robot Autonomous Marine Surface Platform. In *The 8th International Conference on Field and Service Robotics (FSR 2012)*.

Paul Vernaza and Daniel D Lee. 2012. Learning and exploiting low-dimensional structure for efficient holonomic motion planning in high-dimensional spaces. *The International Journal of Robotics Research* 31, 14 (2012), 1739–1760. DOI:http://dx.doi.org/10.1177/0278364912457436

C.W. Warren. 1990. A technique for autonomous underwater vehicle route planning. In *Autonomous Underwater Vehicle Technology, 1990. AUV '90., Proceedings of the (1990) Symposium on*. 201–205. DOI:http://dx.doi.org/10.1109/AUV.1990.110457