Obstacle Avoidance Using Echo Sounder Sonar

P. Calado, R. Gomes, M.B. Nogueira, J. Cardoso, P. Teixeira, P.B. Sujit and J.B. Sousa

Abstract— **Developing obstacle avoidance algorithms for low cost Autonomous Underwater Vehicles using single beam returned echo sounder sonar is a difficult task. In this paper, we propose an intelligent obstacle avoidance algorithm that maps the obstacle, avoids it with a guarantee that it will not get stuck and efficiently traverses a path towards the destination using navigation functions. We present a complete obstacle avoidance system with the help of hybrid automata, probabilistic mapping and navigation functions. Simulation results are presented showing the validity of our approach.**

I. Introduction

Low cost autonomous underwater vehicles (AUVs) are becoming popular among the ocean science community to conduct science missions in a cost effective manner. But low cost comes at a price: limited capabilities. Obstacle avoidance is typically quite limited, if not non-existing, because of this. This capability is specially important because in most science missions AUVs have to navigate in unstructured or partially unknown environments.

Currently, there are two types of sensors that can be used for obstacle avoidance: (i) front mounted imaging sonar [1] and (ii) echo sounder sonar [2]. Imaging sonars are too expensive to be mounted on a low cost AUV. An echo sounder sonar is a comparatively low cost sensor (an order of magnitude cheaper than an imaging sonar). The problem is that the echo sounder returns a single point measurement. Often, this sonar is mounted looking forward to trigger simple obstacle avoidance strategies in the XZ plane. When an obstacle is detected the AUV is commanded to reduce the forward speed and depth until the obstacle is no longer present. When the obstacle is no longer present the AUV is commanded to resume tracking the assigned path. This type of maneuver is simple and effective, but it can only handle avoidance motions in the vertical plane.

Several obstacle avoidance techniques for AUVs have been developed over the years. Petillot et al. [3] use sequential quadratic programming for generating a path that avoids the obstacle which is represented in a map of the world from the segmentation of the sonar image. Khorrami and Krishnamurthy [4], developed a hierarchical path planning with obstacle to track and avoid different types of obstacles using a simulation framework. Eichhorn et al. [5] developed a obstacle avoidance module that switches between reactive maneuvers and path re-planning depending on whether the AUV is able to generate a new path or not. Reactive maneuvers will help the AUV to avoid obstacles while trying to maintain the forward motion towards the goal. Horner et al. [6] developed a vertical avoidance maneuver using a Gaussian-based additive function in conjunction with a sliding mode controller to reduce the cross track error. Several other obstacle avoidance techniques are developed using line-of-sight [7], MDP [8], etc. However, most of the approaches use image based sonar techniques.

In this paper we present a novel obstacle avoidance technique that has features similar to [3], [4], [5] while using a single point echo sounder. This entails an intelligent navigation algorithm that generates a behavior to collect additional information about the obstacle to produce horizontal obstacle avoidance with the guarantee that the AUV will not get stuck in traps while traveling in unstructured environments. The problem of getting stuck in traps arises because of the minimum turning radius of the AUV.

II. Problem formulation

Given a non-holonomic AUV, equipped with a single point beam sonar, traveling between two way points, we want to find a control strategy that guarantees that the obstacles will be avoided until the goal is reached. We assume that the AUV is traveling on a horizontal plane and that the goal is reachable. This entails the existence of feasible paths connecting the initial position to the goal.

We consider a Dubin's model [9] in our derivations:

$$\begin{aligned} \dot{x} &= u\cos\psi \\ \dot{y} &= u\sin\psi \\ \dot{\psi} &= k(\psi_d - \psi) \end{aligned} \tag{1}$$

where u is the speed of the AUV which is constant, ψ is the heading angle of the AUV and ψ_d is the desired heading of the AUV, and $k > 0$ is the gain. The rate of change of heading angle ω is constrained by $-\omega_{\max}$ and $\omega_{\max}$.

In our derivations we use technical data from [2]: the range of the echo sounder is $50m$ with 10 deg field of view. With this sensor range, the AUV must avoid obstacles without getting trapped due to minimum turning radius [9]. Note that, one can think of reducing the velocity of the vehicle to avoid the obstacle. This is not done in our approach because our vehicles are slightly buoyant and we keep the velocity constant in most operations.

The authors are with the Department of Electrical and Computer Engineering, University of Porto, Portugal - 4200-465. pdcalado, ee04171, marcelo.nogueira, jfc, ee04165, sujit, jtasso@fe.up.pt

978-1-61284-4577-0088-0/11/$26.00 ©2011 IEEE

III. Solution approach

The AUV uses an echo sounder sonar to detect obstacles in the direction of its heading. The problem is that the typical sensor measurements are noisy. We use a mapping technique to filter out noise and to map the obstacle. Using this map, a supervisor triggers a safe controller to avoid obstacles. This controller does not ensure that the vehicle will reach the destination, but it ensures obstacle avoidance and guarantees that the AUV is not stuck in traps. A navigation function based controller drives the AUVs towards the destination. The safe controller and the navigation function controller work in tandem to avoid the obstacle and to reach the destination. Figure 1 describes our approach.

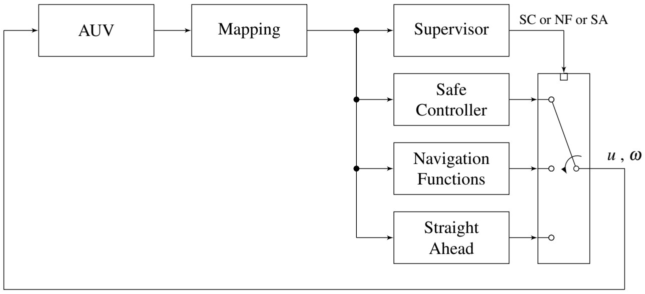

Fig. 1. Block diagram of the complete control scheme.

A. Mapping

In underwater environment, AUVs cannot usually rely on optical instruments (cameras and laser range finders) as water severely attenuates and distorts this kind of signals. So acoustic sensor, such as sonars, are more adequate for underwater environments. Since water does not attenuate sound energy nearly as much as light energy, sonar has a greater range and is also less susceptible to refraction. However, these benefits come at the price of lower resolution and increased noisy sonar data [10].

The difficulty of using an echo sounder sonar to build map is due to the poor directionality of the echo sounder and the error in range measurements. The range measurement errors can typically be modeled by a zero mean Gaussian noise, but there may be outliers due to time-varying multipath and high levels of non-Gaussian ambient noise, such as reflection on ocean floor or surface, regions with different water salinity or temperature [11]. The poor directionality means that the reflection of the sonar ray will not always come from the point of the object intersected by the acoustic axis, but can come from any point of the object inside the cone region depicted in Figure 2. In order to remove these two effects, we must perform some processing of the raw sonar data to extract the obstacles and construct the map in a robust way.

In order to map the sensed environment by the AUV using a point beam, we use concepts from Histogramic In-Motion Mapping(HIMM) [12]. HIMM approach builds a grid map using noisy sensor readings and incrementally updates the map as new sensor measurements are acquired. After building the grid map, we transform it to a feature based representation in order to provide the world as an input to the navigation functions.

The map will be represented by a set $\mathbf{M} = \{o_1,\ o_2,\ \ldots\}$, where $o_i \in \mathbb{R}^{a_i \times 2}$ represents an obstacle in the form of a polygon with a_i vertices (set of pair of points describing the vertices of the polygon). At the beginning of a mission this set is empty (or could contain some previous known obstacles in the working environment), and as the vehicle moves around collecting data the mapping function will populate this set with the obstacles found.

The mapping function receives a set of n valid range measurements (by valid range measurements, we mean range measurements that are below the maximum range returned by the sonar when there is no obstacle ahead) $\mathbf{D} = \{d(k_1),\ d(k_2),\ \ldots,\ d(k_n)\}$ taken at time step k_i, $i = 1, \ldots, n$, and the set of vehicle's pose $\mathbf{x}(k_i) = [x(k_i),\ y(k_i),\ \psi(k_i)]$ at the same time instants that the range measurements were taken $\mathbf{X} = \{\mathbf{x}(k_1),\ \mathbf{x}(k_2),\ \ldots,\ \mathbf{x}(k_n)\}$. From these two sets we construct a grid map using concepts of HIMM. The map G is a grid filled with rectangular cells of length g_h and width g_w. Assume that there are G_h cells along x direction and G_w cells in the y direction. Each cell $G(l, c)$ has an initial value of 0, $\forall l = 1, \ldots, G_h, \forall w = 1, \ldots, G_w$. As the vehicle moves around the environment and collect range measurements, we increase the corresponding values of the grid cells $G(l, c)$. Similar to [12], we update the value of the cell $G(l, c)$ which is in the acoustic axis (which is typical the same as the vehicle heading $\psi(k_i)$) and corresponds to the measured distance $d(k_i)$ from the vehicle pose $\mathbf{x}(k_i)$, as shown in Figure 2. Unlike proposed in [12], we do not decrement the value of any cells during the grid construction.

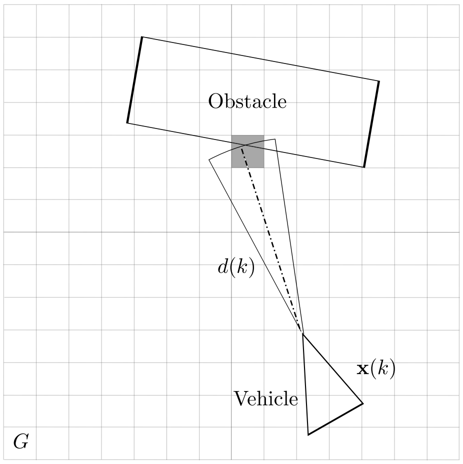

Fig. 2. Construction of the grid G. One cell (shaded) is incremented given a pair of range $d(k)$ and vehicle pose $\mathbf{x}(k)$ at time k.

After processing the sets $\mathbf{D}$ and $\mathbf{X}$ and computing the corresponding grid G, we select from the grid the subset $\mathbf{C} \subseteq G$ of cells such that $\mathbf{C} = \{G(l, c) :\ G(l, c) > v, \forall l, \forall c \in G\}$

for some threshold value v. Since the navigation functions and safe controller requires feature based obstacles, we needed to convert the set $\mathbf{C}$ of point obstacles to a geometric representation. This is done by first using the classical Hough Transform [13] to extract a set $\mathbf{Z} = \{z_1, \ldots, z_m\}$ of m line segments from $\mathbf{C}$, and then each line segment z_i is converted to a convex polygon $o_i \in \mathbb{R}^{4\times 2}$ (convex obstacle representation required by the safe controller), represented by a set of four points describing a rectangle. This process is shown in Figure 3, for a set of points generating a single line. As the Hough transform can act only on a relatively large set of points, and not on individual or small set of points, the elements of the set $\mathbf{S} = \{\mathbf{C}(l,c) : C(l,c) > u, \forall l, \forall c \in \mathbf{C}\}$, for some $u > v$ are converted into small squared obstacles o_p (the size of the square side is equal to the grid spacing). This is important as the vehicle, when traveling straight forward, will obtain a set of range measurements that will increment the value of a small set of cells or even of a single grid cell. The output of the mapping function is a set of obstacles $\mathbf{O} = \{o_1, \ldots, o_{m+s}\}$ consisting of the m rectangular and the s squared obstacles found.

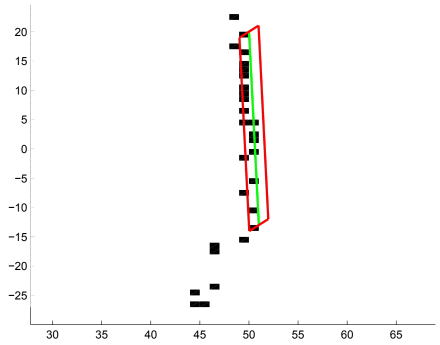

Fig. 3. Hough transform generating line segment (green line) and line segments being converted into rectangular polygon (red).

Once the mapping function returns the set $\mathbf{O}$, we can update the map $\mathbf{M}$ by making $\mathbf{M} = \mathbf{M} \cup \mathbf{O}$. After this, the algorithm resets the sets $\mathbf{D}$, $\mathbf{X}$ and the grid map G. This way we can save memory by keeping only the set $\mathbf{M}$ of obstacles found by the mapping function, and not the entire grid G. It is also possible not to fully reset the sets $\mathbf{D}$ and $\mathbf{X}$, but erase only the oldest values, keeping the $q < n$ more recent data. After resetting the mapping variables, the algorithm starts accumulating new range measurements and vehicle poses on the sets $\mathbf{D}$ and $\mathbf{X}$ respectively, and once the number of valid range measurements reaches n, the mapping function is called again.

After executing the mapping function a few times, the set of obstacles $\mathbf{M}$ might be populated with redundant obstacles, in the sense that there may be obstacles o_i and o_j that are overlapping, representing the same obstacle or the same region of an obstacle in the real world. It is also possible that o_i and o_j do not overlap, but represent different regions of the same obstacle, or two different obstacles on the real world, that are close to each other (pair of vertices $v_o \in o_i$ and $v_p \in o_j$ such that $\|v_o - v_p\|^2 < d_{min}$, for some distance d_{min} that is a function of the vehicle dynamics). This may create a small obstacle free region where the AUV may get trapped. On this cases we will remove o_i and o_j from $\mathbf{M}$ and add a new fused obstacle $o_h = o_i \cup o_j$. The fused obstacle o_h might not be convex, so it necessary to convert it to a convex obstacle before adding it to $\mathbf{M}$. This is done by extracting the convex hull of o_h, that is, the set of vertices of o_h that represents the smallest convex polygon containing all vertices of o_h. It is valid to point out that this conversion might overestimate obstacles, specially the ones that are concave. This over estimation is reasonable as it prevents the vehicle from getting close to obstacles, however it may result in blocking an existing path to the goal, or even rendering no feasible paths connecting the vehicle's current position to the goal when actually there is one.

B. Path planning using navigation functions

We use navigation functions to generate a path that avoids the obstacles detected by the pencil beam sonar. This method was first introduced by Rimon and Koditschek in [14]. It is an improved potential field method with no local minima, that guarantees path convergence to the destination in a world with stationary obstacles. Specifically, we use Dipolar Navigation functions [15] which allow the use of a desired orientation at the goal point. Moreover, this method is adapted to work in an environment with partially known objects.

The vehicle's state is:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} = \begin{bmatrix} q \\ \psi \end{bmatrix}$$

where $q = [x,\ y]^T \in \mathbb{R}^2$ is the vehicle's position in the horizontal plane. Denoting the target destination by $q_d = [x_d,\ y_d]^T \in \mathbb{R}^2$ we can write:

$$\gamma(q) = \|q - q_d\|^2$$

as a distance index to that destination. Similar to [14] consider the vehicle's workspace (area for which the navigation function will be computed) to be a circumference with radius r_{world}. The bounds on that workspace can be represented as $\beta_0 = r_{world}^2 - \|q\|^2$. To align the end point of the resulting trajectories in a certain direction the following term is used:

$$\begin{aligned} H &= \epsilon + n_h \\ &= \epsilon + ([\cos\psi \ \ \sin\psi] \cdot (q - q_d))^2 \end{aligned}$$

where ϵ is a small positive constant. This term also renders the navigation function as dipolar [15].

Yet contrarily to [15] and [14] we represent the obstacles as sets of line segments. The distance index to a line segment is defined as:

$$\beta_i = \min\left\{\|q - l_1\|,\ \frac{|det(l_2 - l_1\ l_1 - q)|}{|l_2 - l_1|},\ \|q - l_2\|\right\}^2$$

where l_1 and l_2 are the points that once connected will define a line segment and i is the index of the line segment constituting the obstacles. The first and third terms in the $\min$ function are the vehicle's distance to each of those points and the second term is the distance to the infinite line defined by l_1 and l_2. Naturally, this implies that they are noncoincident.

We are now ready to write the navigation function we used to compute the paths:

$$\phi = \frac{\gamma}{\left(\gamma^k + H \cdot \prod_{i=1}^{p} \beta_i \cdot \beta_0\right)^{\frac{1}{k}}}$$

where $k > 0$ is a gain parameter and p is the total number of line segments. k should be chosen according to the number of obstacles in the considered scope.

For the sake of simplicity, the longitudinal speed of the vehicle u is assumed to be constant. The control law for the yaw rate takes the following form:

$$\omega = -k_\psi(\psi - \psi_h) + \dot{\psi}_h$$

where $k_\psi > 0$ is a constant gain and ψ_h is the angle of the gradient $\nabla\phi$, and is given by:

$$\psi_h \triangleq \begin{cases} \tan^{-1}(\text{sign}(s) \cdot \phi_y, \text{sign}(s) \cdot \phi_x), & q \neq q_d \\ \psi_d, & q = q_d \end{cases} \qquad (2)$$

where $\phi_x = \frac{\partial \phi}{\partial x}$, $\phi_y = \frac{\partial \phi}{\partial y}$ and $s = J_d^T(q - q_d), J_d(\psi_d) = [\cos\psi_d \;\; \sin\psi_d]^T$ is the current position vector with respect to the destination, projected on the longitudinal axis of the desired orientation [15]. We define $\text{sgn}(x)$ by:

$$\text{sgn}(x) \triangleq \left\{ \begin{array}{ll} 1, & x \geq 0 \\ -1, & x < 0 \end{array} \right\}$$

The stability of the control law (2) was studied and proved in [16]. In figure 4, the isoclines of a navigation function are plotted using polygons defined by line segments. The computed path is depicted in a dotted line. The scenario is the same as in figure 7.

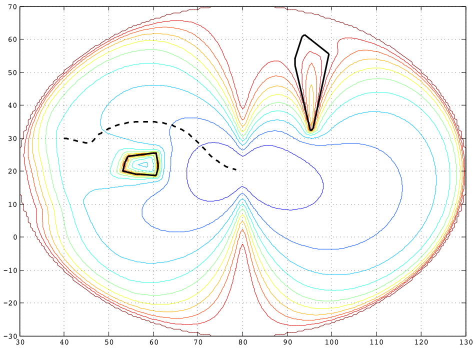

Fig. 4. Isoclines of a navigation function and the path computed using it.

C. Safe controller

We are interested in finding a controller that keeps the vehicle inside a safe set, that is, a set for which we can guarantee that the vehicle does not enter any obstacle, as long as its trajectory remains in that set. In order to avoid collisions we introduce the two avoidance maneuvers that drive the vehicle away from the obstacle, turning it in the desired direction. These maneuvers are symmetric semi-circular trajectories, departing from the vehicle's position, with radius equal to the minimum turning radius of the vehicle. Thus, when the vehicle is moving, we test its state continuously, analyzing the viability of the avoidance maneuvers starting from the state at the next time step (a trajectory is viable if and only if it does not collide with the obstacle). An avoidance maneuver should be engaged every time that, by not doing so, the vehicle will enter a collision course with the obstacle. Our control strategy is implemented through a hybrid automaton (see figure 5), covered in greater detail in [17].

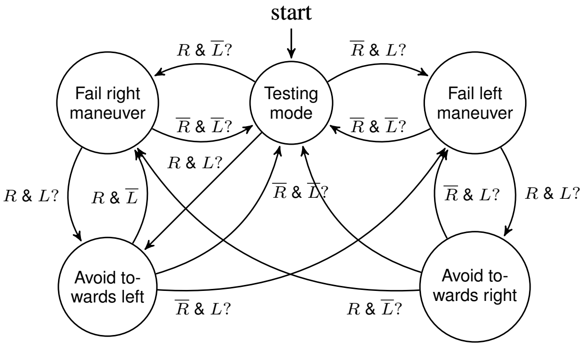

Fig. 5. Hybrid automaton for the safe controller

The controller starts in a test mode, testing the viability of the left and right avoidance maneuvers separately. Should one of the maneuvers fail (condition R or L), there is still a way to avoid the obstacle, and so the controller transits to the correspondent fail mode. If both maneuvers become inviable simultaneously or, if in fail mode, the remaining maneuver becomes inviable, the controller starts the correspondent avoidance maneuver.

D. Supervisor

The Supervisor is responsible for switching between the Safe Controller (SC) and the Navigation Function Controller (NFC). Whenever the SC declares that the present state is safe, then the NFC is used to compute a new path to the goal and the vehicle is driven towards that path. Otherwise, the SC is kept enabled until a new safe set is found.

This scheme comprises three distinct states (see figure 6). It starts in a control mode that drives the vehicle straight towards the target destination q_d. Condition U becomes true when the mapping process builds its first object and the vehicle's state is considered unsafe. This transition triggers the SC which steers the vehicle right or left of the detected obstacle to acquire new data. Eventually, the mapping stops detecting obstacles in front of the vehicle and the unsafe condition U becomes false. The NFC comes into play and computes a path around the mapped obstacles to reach a point between the vehicle's current position, q and the target q_d. Once that path has been successfully computed, a simple path-tracking controller drives the AUV towards it. If new obstacles are detected or the distance to the closest detected obstacle becomes too small then U becomes true and the Supervisor switches SC back on. If on the other hand, no new obstacles are detected nor does the vehicle travel close to an obstacle and reaches the end of the computed path, then P becomes false. Consequently, the Supervisor transitions back to the Straight Ahead state.

The controllers used to keep the vehicle heading straight or to track a path given by the navigation function are not discussed here.

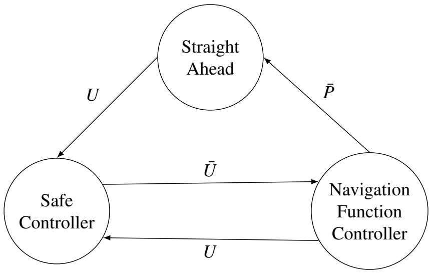

Fig. 6. Block diagram showing how the supervisor controller works.

IV. Simulation Results

We developed a simulation framework in Matlab to test our method. The vehicle maintains a constant forward speed of 1 m/s and has a maximum turn rate ω_{max} of 0.6. We consider two different scenarios. In the first one the AUV must reach the target while avoiding multiple small obstacles (figure 7), while in the second one there is only one large obstacle(figure 8). The vehicle has no prior knowledge of the world, yet the mapping method is tuned differently for each scenario. In the multiple obstacles scenario the parameter d_{min} is adjusted to prevent the creation of large objects from smaller distant objects. The yellow structures represent the obstacles present in the world and the black frames represent the obstacles created in the mapping process. The trajectory described by the AUV is colored differently for each control mode: blue for Straight Ahead, green for NFC and red for SC. The magenta dots represent the measurements taken by the echo sounder sonar. The initial position is $(0, 0)$ and the final position is $(110, -15)$.

As it can be seen from both figures, the path taken by the AUV is not optimal, but it avoids all the obstacles. It is also clear that the SC is enabled for a short amount of time, since its only goal is determining whether the vehicle should turn right or left. Additionally, note that the obstacles generated by the mapping process are an over-approximation of the real ones to prevent the vehicle from getting too close to them.

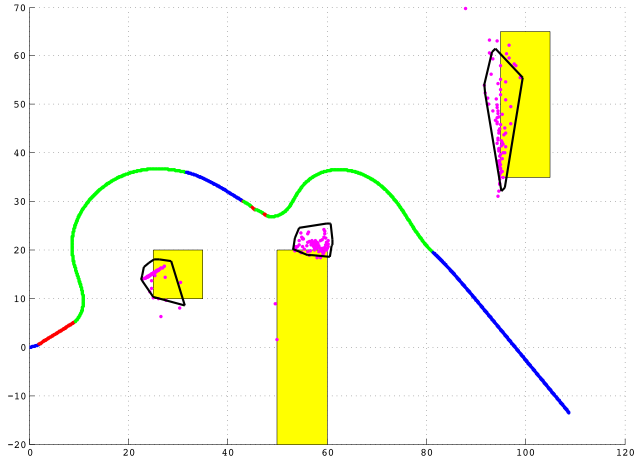

Fig. 7. Vehicle's behavior in a scenario with multiple obstacles.

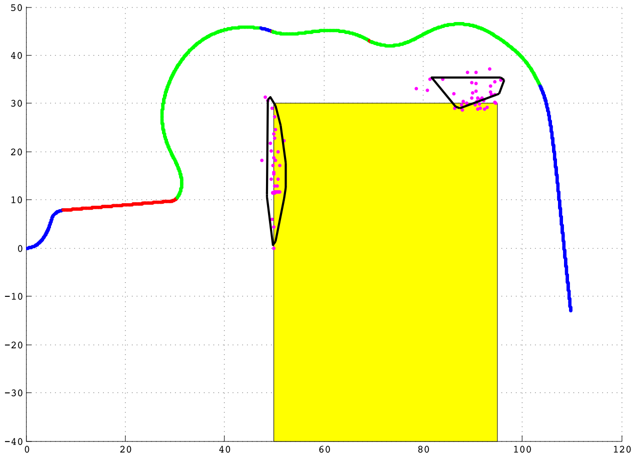

Fig. 8. Vehicle's behavior in a scenario with a single obstacle.

V. Conclusions and Future Work

In this paper, we presented a collision avoidance technique for a AUV navigating in unstructured environments using single beam echo sounder. We build a map of the obstacles by collecting and processing sonar data and then, using the map, we determine safe paths with the help of a hybrid controller and navigation functions. The hybrid controller guarantees that the AUV does not get stuck in traps while the navigation functions generate a smooth trajectory with the limited information available from the map. The simulation results validated our approach.

Future extensions to this work should consider the relaxation of the assumptions made. Moreover, we will derive a formal proof of controller stability and safety and discuss the computational complexity of the methods. In the future, we will field test the approach in one of our AUVs.

ACKNOWLEDGEMENTS

The authors wish to recognize the support of the USTL. This work was partly funded by FEDER-COMPETE and the Portuguese government under project FCOMP-01-0124-FEDER-007221.

REFERENCES

[1] www.blueview.com.

[2] http://www.imagenex.com.

[3] Y. Petillot, I. Tena Ruiz, and D. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 2, pp. 240 –251, apr. 2001.

[4] F. Khorrami and P. Krishnamurthy, "A hierarchical path planning and obstacle avoidance system for an autonomous underwater vehicle," in *Proc. of the American Control Conference*, 2009, pp. 3579–3584.

[5] M. Eichhorn, "A new concept for an obstacle avoidance system for the auv and slocum glider operation under ice," in *Proc. of the IEEE OCEANS*, 2009, pp. 1–8.

[6] D. P. Horner, A. J. Healey, and S. P. Kragelund, "Auv experiments in obstacle avoidance," in *Proc. of the MTS/IEEE OCEANS*, 2005, pp. 1464–1470.

[7] X. Wu, Z. Feng, J. Zhu, and R. Allen, "Line of sight guidance with intelligent obstacle avoidance for autonomous underwater vehicles," in *Proc. of the MST/IEEE OCEANS*, 2006, pp. 1–6.

[8] H. Kawano, "Real-time obstacle avoidance for underactuated autonomous underwater vehicles in unknown vortex sea flow by the mdp approach," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 3024–3031.

[9] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.

[10] A. E. Johnson and M. Hebert, "Seafloor map generation for autonomous underwater vehicle navigation," pp. 3–145, 1996. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.33.4051

[11] M. Chitre, "A high-frequency warm shallow water acoustic communications channel model and measurements," *The Journal of the Acoustical Society of America*, vol. 122, no. 5, pp. 2580–2586, Nov. 2007, PMID: 18189549. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18189549

[12] J. Borenstein, M. Ieee, Y. Koren, and S. M. Ieee, "Histogramic In-Motion mapping for mobile robot obstacle avoidance," *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, vol. 7, pp. 535—539, 1991. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.22.2998

[13] P. V. C. Hough, "Machine analysis of bubble chamber pictures," in *International Conference on High Energy Accelerators and Instrumentation*, CERN, Geneva, Switzerland, 1959.

[14] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, October 1992.

[15] G. Roussos, G. Chaloulos, K. Kyriakopoulos, and J. Lygeros, "Control of multiple non-holonomic air vehicles under wind uncertainty using model predictive control and decentralized navigation functions," in *Decision and Control, 2008. CDC 2008. Proc. of the 47th IEEE Conference on*, dec. 2008, pp. 1225 –1230.

[16] G. Roussos, D. V. Dimarogonas, and K. J. Kyriakopoulos, "3d navigation and collision avoidance for nonholonomic aircraft-like vehicles," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, pp. 900–920, 2010.

[17] J. F. Cardoso, "Invariance and control of autonomous vehicles," Master's thesis, FEUP, 2010.