Path Planning for Bathymetry-aided Underwater Navigation

Gao Rui
ARL, Tropical Marine Science Institute
National University of Singapore
gaorui@arl.nus.edu.sg

Mandar Chitre
ARL, Tropical Marine Science Institute and
Department of Electrical & Computer Engineering
National University of Singapore
mandar@arl.nus.edu.sg

***Abstract*—It has been shown that significant variation in the bottom topography of underwater terrain could be used to dramatically improve underwater localization accuracy. However, path planning with bathymetric aids has not been extensively explored in literature. Localization accuracy strongly depends on the path taken by an underwater vehicle. Given a starting point and a destination, we develop an algorithm to plan a path that will ensure good localization. We adopt an information entropy measure to assess the localization uncertainty of a particle filter, such that a generic posterior can be described. We use this to drive a path planning algorithm that minimizes uncertainty using reinforcement learning and Gaussian process regression. We test our algorithm using bathymetric data and show that it generates near-optimal paths with good localization accuracy at the destination.**

***Index Terms*—underwater navigation, bathymetry, terrain-aided navigation, underwater vehicles, entropy, Gaussian process regression, particle filter.**

I. Introduction

The idea of using bathymetry for underwater navigation has been explored before [1]–[3]. The key idea is to match the local bathymetry as seen by an underwater vehicle against a reference map, and to estimate the location of the vehicle on that map. High-end autonomous underwater vehicles (AUVs) may use a multibeam sonar to sense the local bathymetry, while low-cost AUVs may make do with a single echo-sounder or altimeter. In [1], the authors showed a strong correlation between localization accuracy and variation of bottom topography. In this paper, we take this idea a step further and ask what path should an AUV take from a given starting point to a destination to ensure good localization?

As far as we are aware, this question has not been systematically answered in literature. In [4], heuristics were used to visit salient points (locations with more bathymetric variation) for better localization. In [5], terrain dispersion, roughness and terrain entropy were evaluated. However, the waypoints along the path were selected manually. Rather than appeal to heuristics, we pose path planning as an optimization problem and solve it using the framework of reinforcement learning and Gaussian process regression (GPR).

Given a starting point and a destination, our goal is to plan a path such that the positioning uncertainty is minimized when the vehicle reaches the destination. With bathymetric measurements incorporated into localization, the *a posteriori* description of the location uncertainty is often poorly modeled by a Gaussian distribution. Multi-modal distributions may arise when the location uncertainty is bifurcated at bathymetric ridges. Conventional characterization such as mean and covariance does not capture such distributions well. To tackle this problem, particle filters (PF) are often used. To describe and compare the uncertainty in PF, we adopt an information entropy measure [6] and Bhattacharyya Coefficient [7].

In what follows, we first formulate the particle filtering localization and information theoretic measure for localization uncertainty in Section II. We then demonstrate the information entropy measures for localization along two different paths in Section III. In Section IV, we state the path planning problem and formulate the criterion for an optimal path. We propose a path planning algorithm using reinforcement learning and GPR. The simulation and results based on bathymetric data are presented in Section V.

II. Particle Filter Based Localization and Entropy Measure

Let $\mathbf{x}_k$ ($\mathbf{x}_k \in \mathbb{R}^2$) be the position to be estimated at time step k. We are only interested in the easting and northing as the depth can be measured by a depth sensor. We model the underwater vehicle to move at constant speed in the heading direction. Given an action a_k that directs the vehicle's heading, the general system evolution model is:

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, a_k, \omega_k), \tag{1}$$

where ω_k is the process noise. The single-point bathymetry measurement model is:

$$z_k = h(\mathbf{x}_x, \nu_k), \tag{2}$$

where ν_k is the measurement noise. The measured bathymetry is obtained as the sum of measurements from the depth sensor and altimeter, and is assumed to be corrupted with additive noise. The process noise ω_k and measurement noise ν_k are modeled as mutually independent Gaussian white-noise sequences.

The particle filter localization follows a standard Particle filtering (PF) framework [8]. We have the particle set $\{\mathbf{x}_k^i, q_k^i\}$ where q_k^i is the weight for ith particle positioned at $\mathbf{x}_k^i$ and $\sum_{i=1}^N q_k^i = 1$. The position estimate $\hat{\mathbf{x}}_k$ is chosen as the mode of the distribution described by the particle set.

The PF-based entropy is derived from differential entropy of the posterior approximated by the particles [6]:

$$H\left(p(\mathbf{x}_k|Z_k)\right)) \approx \log\left(\sum_{i=1}^{N} p(z_k|\mathbf{x}_k^i) q_{k-1}^i\right) - \sum_{i=1}^{N} \log\left(p(z_k|\mathbf{x}_k^i)(\sum_{j=1}^{N} p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^j) q_{k-1}^j)\right) q_k^i, \quad (3)$$

where $Z_k = \{z_1, z_2, \ldots, z_k\}$ includes the bathymetry measurements in history up to the current time step k. $p(\mathbf{x}_k|Z_k)$ is the posterior distribution after the series of bathymetry measurements. As measurements may not be available at every step, we derive the PF-based entropy for the prior. Given the probability distribution represented by PF:

$$p(\mathbf{x}_k|Z_{1:k-1}) \approx \sum_{i=1}^{N} q_{k|k-1}^i \delta(\mathbf{x} - \mathbf{x}_{k|k-1}^i). \quad (4)$$

The weak convergence law for PF states [9]:

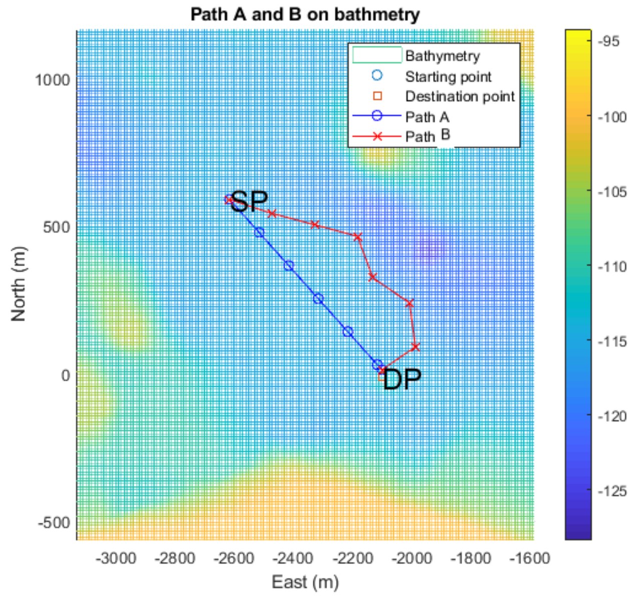
$$\lim_{N\to\infty} \sum_{i=1}^{N} g(\mathbf{x}_{k|k-1}^i) q_{k|k-1}^i = \int_{\mathcal{X}} g(\mathbf{x}_k) p(\mathbf{x}_k|Z_{1:k-1}) d\mathbf{x}_k, \quad (5)$$

where $g(\cdot)$ is a continuous and bounded function, the information entropy for $p(x_k|Z_{k-1})$ is:

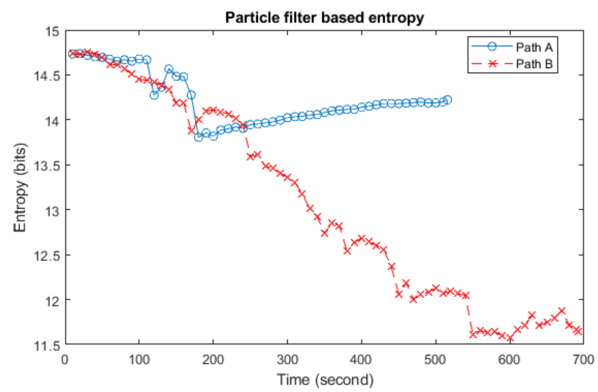
$$\begin{aligned} H\left(p(\mathbf{x}_k|Z_{1:k-1})\right) &= -\int_{\mathcal{X}} \log p(\mathbf{x}_k|Z_{1:k-1}) p(\mathbf{x}_k|Z_{1:k-1}) d\mathbf{x}_k \\ &= -\lim_{N\to\infty} \sum_{i=1}^{N} \log p(\mathbf{x}_{k|k-1}^i|Z_{1:k-1}) q_{k|k-1}^i \\ &= -\lim_{N\to\infty} \sum_{i=1}^{N} \log\left(\lim_{N\to\infty} \sum_{j=1}^{N} p(\mathbf{x}_{k|k-1}^i|\mathbf{x}_{k-1}^j) q_{k-1}^j\right) q_{k|k-1}^i \\ &\approx -\sum_{i=1}^{N} \log\left(\sum_{j=1}^{N} p(\mathbf{x}_{k|k-1}^i|\mathbf{x}_{k-1}^j) q_{k-1}^j\right) q_{k|k-1}^i. \end{aligned} \quad (6)$$

III. Illustrative Examples

We examine the PF-based entropy along two paths shown in Fig. 1(a). Both paths have the same large initial uncertainty. Bathymetric measurements made at every 10 seconds help reduce the localization uncertainty initially for both paths. Straight-line path A goes through a flat area with little bathymetric variation. The entropy of localization uncertainty increases from roughly the 200$^{\text{th}}$ second. Path B takes a detour and therefore longer time to reach the destination. Without bathymetric information (pure dead reckoning), the vehicle would incur a larger uncertainty for longer missions. But the PF-based entropy decreases rapidly as vehicle moves along the area with significant bathymetric variability. The significant variation along path B makes the measured bathymetry unique and therefore improves localization accuracy. At the destination, Path B has a smaller entropy compared with Path A.



(a) Two different paths (A and B) from the same starting point (SP) to the destination point (DP).



(b) Entropy of the particle filter for paths A and B

Fig. 1. The entropy of the particle filter in a bathymetric navigation depends on the path taken from starting point to destination.

With this example, we have shown that the localization accuracy strongly depends on the path that an AUV takes, if the AUV uses bathymetric aids for navigation. So how does one select a path that yields good localization?

IV. Path Planning Using Reinforcement Learning

A. Problem Statement

We define the state as $S_k = \{\mathbf{x}_k^i, q_k^i\}$. For simplicity, we resample the particle set for the state, such that particles have the same weight. The policy contains a series of actions $\pi(S_k) = \{a_k, a_{k+1}, a_{k+2}, \ldots\}$ and is chosen such that it leads to a minimum entropy when vehicle reaches the destination. Each action is applied for τ time steps with constant heading speed. We have:

$$\begin{aligned}\pi(S_k) &\leftarrow \arg \min_{a \in \mathcal{A}(S_k)} Q(S_k, a_k), \\ Q(S_k, a_k) &= \sum_{S_{k+1}} p_{\text{tr}}(S_k \xrightarrow{a_k} S_{k+1}) V(S_{k+1}), \\ V(S_k) &= \min_{a_k \in \mathcal{A}(S_k)} Q(S_k, a_k),\end{aligned} \tag{7}$$

where $V(\cdot)$ is the value function of a state, that minimizes the reinforcement learning Q-function $Q(\cdot)$ across all possible actions [10]. $p_{\text{tr}}(S_k \xrightarrow{a_k} S_{k+1})$ is the transition probability from state S_k to state S_{k+1}, due to the non-deterministic evolution of the position by taking action a_k. Compared to standard Bellman's equation [11], the transition reward is zero and the discount factor is 1. Therefore, the value of any given state S is the entropy value of state at the destination. All states along the same path have the same entropy value.

In subsequent sections, we set $p_{\text{tr}}(S_k \xrightarrow{a_k} S_{k+1}) = 1$ during path planning, but evaluate the performance of the algorithm over Monte Carlo simulations to include the non-deterministic evolution of vehicle position.

B. Gaussian Process Regression (GPR) and Algorithm

Because of a very large state and action space, it is not possible to use dynamic programming to solve the sequential decision process problem posed in (7). We instead resort to using approximate techniques motivated by ideas in [11].

We model the value function as a Gaussian process. The continuous value in the state space can be inferred with a Gaussian process prior. We start with randomly generated paths and simulate the localization to get the estimated values $V^*(S)$. A state-value table is constructed with each entry recording a state S and corresponding value $V^*(S)$. In each iteration we generate the policy according to the estimated value from the table and evaluate the policy from simulation. Then we update the state-value table with the new values. We refine the state-value table over iterations.

1) Policy Generation: Given any state S_k, we form an action space $\mathcal{A}(a_k)$ containing all possible actions. The possible actions are the headings linearly spaced within $(-\frac{\pi}{2}, \frac{\pi}{2})$ when vehicle heads towards the destination. To cover all the bathymetry grids, the resulting positions after $\tau = 100$ time steps need to be roughly 10 meters apart to each other. Therefore, there are 29 actions in the action space. If the current position is within τ time steps movement to the destination, we navigate the vehicle to the destination and the path planning is completed.

A resulting state $(S_k, a_k) \rightarrow S_{k+1}$ is generated for each action. We estimate the state value $V(S_{k+1})$ using GPR [12], based on the state-value table. The policy is updated with the action that leads to the next state with minimum value.

To estimate $V(S)$ (we drop the subscript for simplicity of notation), we choose nearby states with smaller values. For example, we only use the nearby states whose values are below the 75^{th} percentile. We define $\mathcal{B}(S,T)$ - the distance between states S and T using the Bhattacharyya Coefficient $\rho(S,T)$ [7]:

$$\mathcal{B}(S,T) = \sqrt{1 - \rho(S,T)}. \tag{8}$$

Coefficient $\rho(S,T)$ measures the overlap between two distributions. Let the discrete densities of particle sets S and T be $\{\hat{s}_u\}_{u=1,\dots m}$ and $\{\hat{t}_u\}_{u=1,\dots m}$, where m is the number of bins and $\sum_{u=1}^{m} \hat{s}_u = 1, \sum_{t=1}^{m} \hat{s}_u = 1$. We have $\rho(S,T) = \sum_{u=1}^{m} \sqrt{\hat{s}_u \hat{t}_u}$.

2) Policy Evaluation: After the path is generated, we evaluate the path by re-running along the planned waypoints. The waypoints are generated τ time steps apart along the planned path. When executing the waypoints, vehicle compares its estimated position with the targeted waypoint, and generates control commands accordingly. The details of path execution are presented in Section V. The path is evaluated at the median value over Monte Carlo simulations. This minimizes the discretization error from limited number of the particles. The new state is added to state-value table if its value is smaller than the nearby ones. We also remove the nearby states with large values.

3) The Policy Iteration Algorithm: We iterate the policy generation and evaluation, and summarize the algorithm as follows:

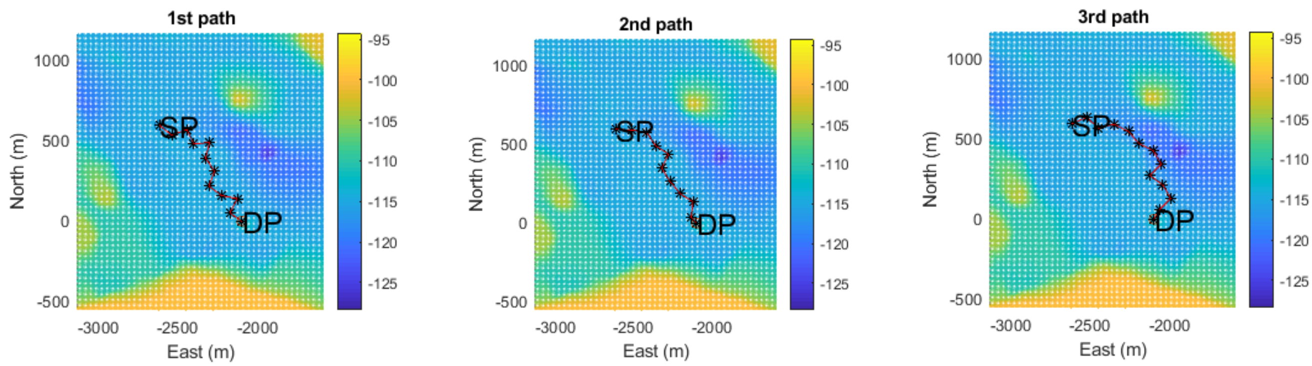
Algorithm 1 Policy iteration

```
Randomly generate a number of paths (e.g. 500) and construct
the state-value table
repeat
    Start from starting point
    while The destination is not reached do
        Generate action space
        for each action in the action space do
            Estimate the value based on state-value table
        end
        Choose the action with the minimum value and move
    end
    Re-evaluate the value of the generated path
    Update the state-value table
until Stabilized;
```

V. Simulation and Performance Evaluation

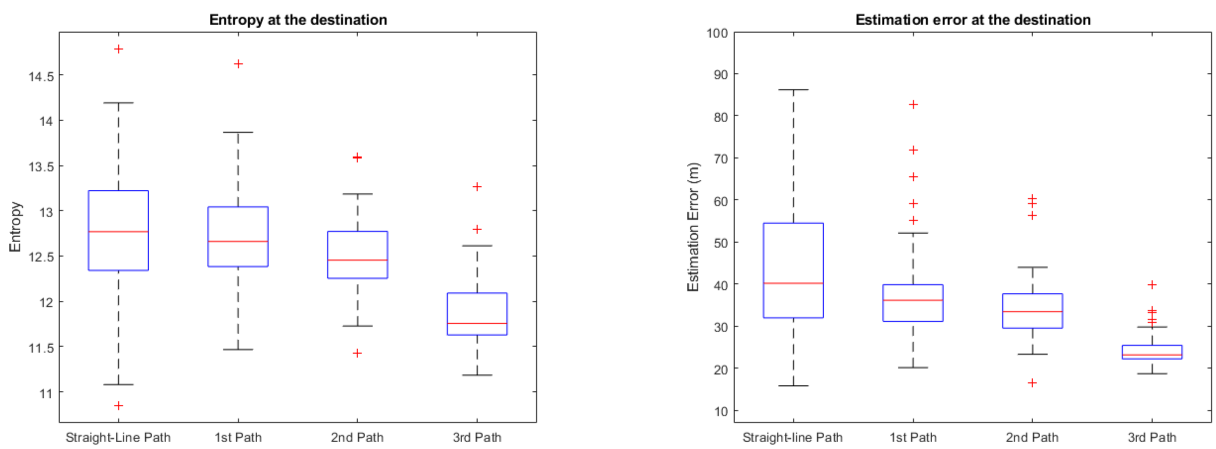
A. Underwater Vehicle Navigation

Navigation is the activity of ascertaining one's position, planning and following a route. To evaluate how the path planning benefits localization, we simulate the route-following using waypoints. A series of waypoints are sampled from the planned path, including the destination. The vehicle compares its estimated position $\hat{\mathbf{x}}_\mathbf{k}$ with the targeted waypoint, and gives an action a_k that directs the vehicle heading such that the vehicle heads towards the targeted waypoint. We set a 10-meter range to determine whether vehicle has reached the waypoint. Once $\hat{\mathbf{x}}_\mathbf{k}$ is within 10 meters to the waypoint, the vehicle changes to target the subsequent waypoint. If the vehicle reaches a later waypoint before the current one, it continues route-following using the waypoint after it. The mission ends



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Fig. 2. Mission 1: Planned path generated over iterations.



(a) Estimation entropy. (b) Estimation error.

Fig. 3. Mission 1: Performance at the destination.

when vehicle reaches the destination (within a 10-meter range) or the mission exceeds the maximum allowable duration.

Our bathymetry map has a resolution of 10 meters. The vehicle makes a measurement at every 10 seconds when moving at 1 meter per second.

B. Mission 1

We tested our algorithm on bathymetry data collected at a test location in Singapore waters. With the starting point (SP) and destination point (DP), Fig. 2 shows that as the algorithm iterates, the planned path evolves to the path through an area with more bathymetric variation.

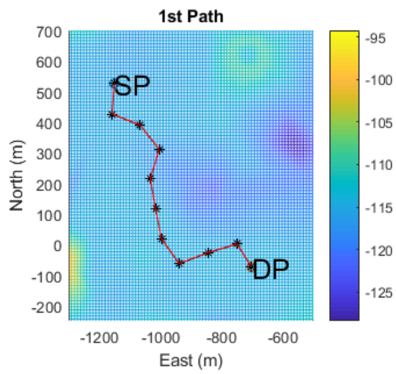
The navigation accuracy is estimated with 50 simulated runs. The entropy at the destination (Fig. 3(a)) drops with iterations, and is smaller compared with the entropy at the end of a straight-line path. The localization errors at the destination are shown in Fig. 3(b) for straight-line path and generated paths over iterations 1 to 3. With the same propagation and observation capability, routes through more bathymetric variation have better localization accuracy. A good path is generated within a few iterations.

C. Mission 2

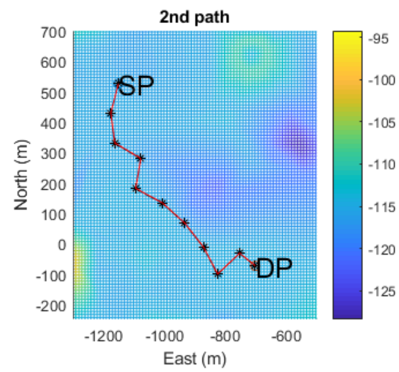
We test another pair of starting and destination points. In contrast with the pair in Mission 1, this pair has a small basin between them. In the first three iterations in Fig. 4, paths are generated along one side of the basin. From the fourth iteration, the generated path starts to move to the other side of the basin. It is important to highlight that paths along maximum bathymetric variation may not always lead to the smallest positioning error. The bathymetry matching performance depends on the prior and the exact bathymetry, specifically, how unique the measured bathymetry is compared with the others in the prior.

VI. Conclusions

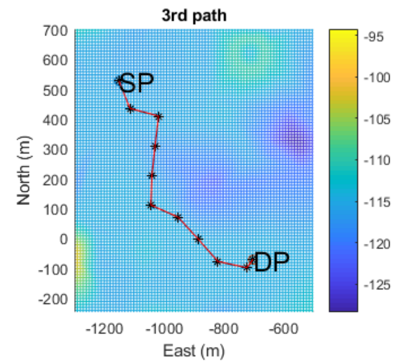
We proposed a path planning algorithm to improve underwater localization with the aid of bathymetric measurements. The algorithm plans the path such that the localization uncertainty at the destination is minimized. Simulation studies, with



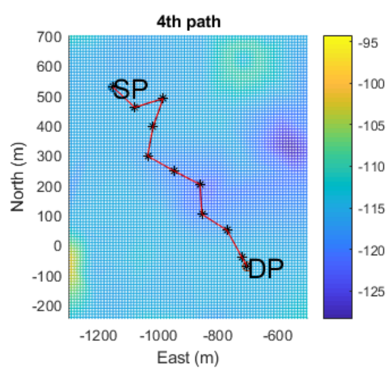
(a) Iteration 1



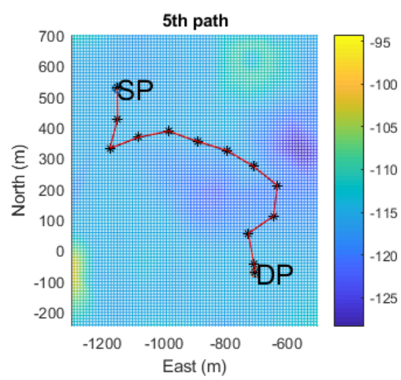
(b) Iteration 2



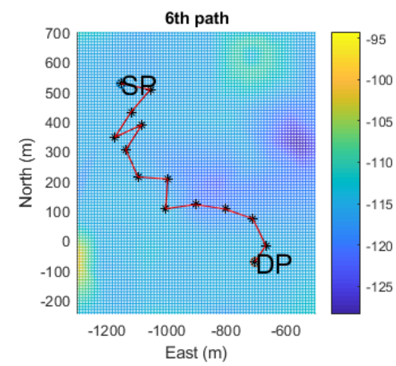
(c) Iteration 3



(d) Iteration 4

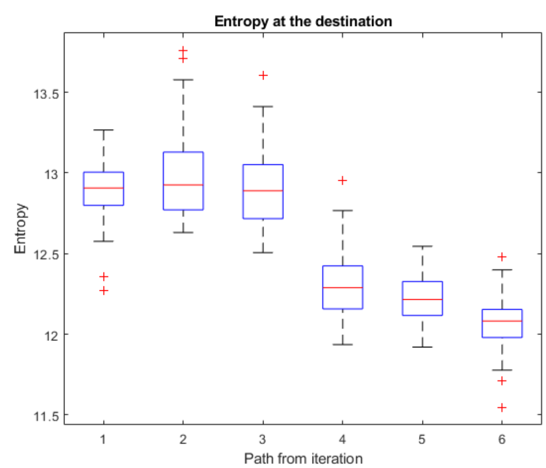


(e) Iteration 5

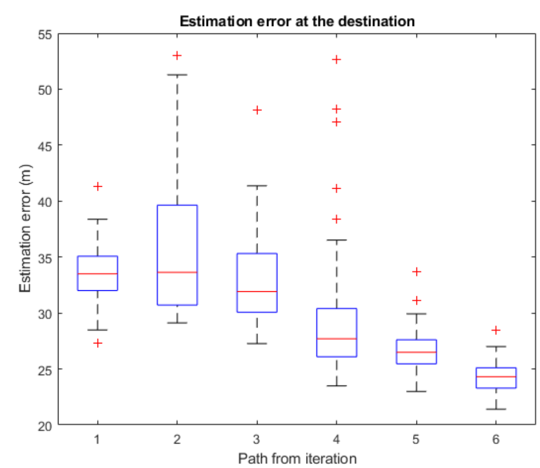


(f) Iteration 6

Fig. 4. Mission 2: Planned path generated over iterations.



(a) Estimation entropy.



(b) Estimation error.

Fig. 5. Mission 2: Performance at the destination.

measured bathymetry from Singapore waters, were conducted to show the generated paths and the localization performance along these paths. We showed that a good path can be generated within only a few iterations of the algorithm.

REFERENCES

[1] B. Kalyan and M. Chitre, "A feasibility analysis on using bathymetry for navigation of autonomous underwater vehicles," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC '13, (New York, NY, USA), pp. 229–231, ACM, 2013.

[2] D. Meduna, S. Rock, and R. McEwen, "Closed-loop terrain relative navigation for auvs with non-inertial grade navigation sensors," in *Autonomous Underwater Vehicles (AUV), 2010 IEEE/OES*, pp. 1–8, Sept 2010.

[3] S. Carreno, P. Wilson, P. Ridao, and Y. Petillot, "A survey on terrain based navigation for auvs," in *OCEANS 2010*, pp. 1–7, Sept 2010.

[4] E. Galceran, S. Nagappa, M. Carreras, P. Ridao, and A. Palomer, "Uncertainty-driven survey path planning for bathymetric mapping," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 6006–6012, Nov 2013.

[5] D. Peng, T. Zhou, H. Li, and W. Zhang, "Terrain aided navigation for underwater vehicles using maximum likelihood method," in *2016 IEEE/OES China Ocean Acoustics (COA)*, pp. 1–6, Jan 2016.

[6] Y. Boers, H. Driessen, A. Bagchi, and P. Mandal, "Particle filter based entropy," in *2010 13th International Conference on Information Fusion*, pp. 1–8, July 2010.

[7] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, vol. 2, pp. 142–149 vol.2, 2000.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Feb 2002.

[9] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods for practitioners," *IEEE Transactions on Signal Processing*, vol. 50, pp. 736–746, Mar 2002.

[10] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[11] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality (Wiley Series in Probability and Statistics)*. New York, NY, USA: Wiley-Interscience, 2007.

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.