This paper was submitted as a final project report for CS6424/ECE6424 *Probabilistic Graphical Models and Structured Prediction* in the spring semester of 2016.

The work presented here is done by students on short time constraints, so it may be preliminary or have inconclusive results. I encouraged the students to choose projects that complemented or integrated with their own research, so it is possible the project has continued since this report was submitted. If you are interested in the topic, I encourage you to contact the authors to see the latest developments on these ideas.

Bert Huang
Department of Computer Science
Virginia Tech

Autonomous Exploration Of Benthic Environment Using Gaussian Process Filter

Abhilash Chowdhary
Bradley Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24061
abhilash@vt.edu

Abstract

Autonomous underwater vehicle (AUV) navigation and localization in underwater environments is particularly challenging due to the rapid attenuation of Global Positioning System (GPS) and radio-frequency signals. Underwater communications are low bandwidth and unreliable, and there is no access to a global positioning system. In this project, we propose a Gaussian Process(GP) based predictive and filtering model for localizing and mapping the sea bed with AUV. We show how bayesian filtering can be hierarchically modeled with Gaussian Processes in a nonlinear stochastic dynamic system such as AUV to infer about its location given the depth (observation) at that point. The accuracy of this model increases with availability of more ground truth regarding depth of the sea bed.

1 Introduction

1.1 Motivation

Autonomous underwater vehicles (AUVs) are becoming very useful tools in carrying out ecological surveys in large water bodies. Recently, they have also been used for various search and rescue missions owing to natural and man-made disasters in large water bodies. AUVs offer many advantages over other traditional techniques like towed video surveys and diver transects, one of them being their ability to operate at depths of thousands of meters from the water surface. However, as any robot, an AUV also has its limitations. The extent of its mission are limited by the battery power it carries and there is usually also a time constraint, as in most cases the AUV requires a support vessel with finite resources. All these constraint limit the spatial extent and the resolution of practical survey. Therefore there is a need to optimise missions to provide the maximum benefit.

Majority of AUVs do not have full autonomy in carrying out there missions. Usually, a preprogrammed trajectory is fed into the AUVs for carrying out its mission. Mission cannot be altered manually once an AUV is under water due to limited bandwidth of communication. Sensing too is usually passive, in that the perceived sensor data does not influence the actions taken by the vehicle and many sensor like GPS do not work under couple of feet of water surface. Hence, there is a need for optimal online mission planning strategies so that AUV can by itself learn the benthic environment and steer under the water without running into obstacles.

1.2 Methodology

To let the AUV steer under water autonomously, it needs to to be aware of its location ($\mathbf{x}$) in the water body and should be able to predict the depth ($\mathbf{z}$) at next timestamp given current control input ($\mathbf{u}$).

Probabilistic Graphical Models and Structured Prediction Final Project Report. Spring 2016, Virginia Tech.

Using Bayesian filtering, substantial progress has been made in the past decade to localize a mobile robot. Briefly, this involves maintaining a probability distribution, $p(\mathbf{x_k})$, over the robot's possible poses at time $\mathtt{k}$. This distribution is updated at each time step, given an action $\mathbf{u_{k-1}}$ and an observation $\mathbf{z_k}$, using Bayes rule:

$$p(\mathbf{x_k}|\mathbf{x_{k-1}}, \mathbf{u_{k-1}}, \mathbf{z_k}) \propto p(\mathbf{z_k}|\mathbf{x_k}) \int p(\mathbf{x_k}|\mathbf{x_{k-1}}, \mathbf{u_{k-1}}) p(\mathbf{x_{k-1}}|\mathbf{z_{1:k}}) \mathbf{dx_{k-1}} \tag{1}$$

$p(\mathbf{x_k}|\mathbf{x_{k-1}}, \mathbf{u_{k-1}}, \mathbf{z_k})$ represents the estimate for present location of AUV at time $\mathtt{k}$.$p(\mathbf{z_k}|\mathbf{x_k})$ represents the likelihood of depth measurement $\mathbf{z_k}$ at time $\mathtt{k}$ and location $\mathbf{x_k}$. So the actual problem can be defined in two parts. Firstly, modeling the likelihood function in (1) based on system dynamics and sensor readings is important for having good prediction and filtering of latent variable $\mathbf{x}$. Secondly, we are interested in inferring about the current location $\mathbf{x_k}$.

The first part of the problem can be stated more precisely as follows: let $D \in \{(\mathbf{x_0}, \mathbf{z_0}), (\mathbf{x_1}, \mathbf{z_1}), ...(\mathbf{x_m}, \mathbf{z_m})\}$ denote a set of geo-referenced depth sensor readings, where $\mathbf{x_i}$ represents location at which noisy sensor readings $\mathbf{z_i}$ was taken.We need to estimate the density $p(\mathbf{z}|\mathbf{x})$ with the prior non-uniform distribution over $\mathbf{x}$. A straightforward approach to this is to treat it as a regression problem. By fitting a Gaussian noise on both GPS readings $\mathbf{x_i}$ and depth sensor readings $\mathbf{z_i}$ we get point estimate on the weight $\mathbf{w_i}$ parameters with maximum a posteriori (MAP) estimate. The problem with this approach is it puts all the weight on the MAP estimate of weights of the training data and it does not put a uncertainty over them.

$$\mathbf{z_i} = \mathbf{\Sigma_j w_j x_{ij}} + \epsilon \tag{2}$$

In this project, we're interested in a Bayesian approach to the problem. The introduction of training data D induces a posterior over weights $\mathbf{w_j}$. Where the straightforward approach simply keeps the MAP estimate of weights and adds a noise, the Bayesian approach keeps the entire distribution over possible weights and hence maintains a measure of uncertainty in the weights at any point in the map. Here we use a Bayesian approach to regression, Gaussian Process(GP) model, to estimate the distribution of likelihood $p(\mathbf{z_k}|\mathbf{x_k})$ in equation (1).

The second part is the problem of filtering and smoothing. Here we use particle filtering technique to infer about the the current location $\mathbf{x_k}$ given the likelihood and assuming a Gaussian noise on prior estimate of location from control input $\mathbf{u_{k-1}}$.

$$\begin{aligned} \mathbf{x_k} &= f(\mathbf{u_{k-1}}) + \epsilon \\ \epsilon &\sim \mathcal{N}(0, \sigma^2) \end{aligned} \tag{3}$$

To summarize, we use the following approach for localization:

1. Obtain a training set D from the environment
2. Using Gaussian Process regression, fit a model(including uncertainty) to the training set.
3. Use the model to produce a likelihood function for localization.
4. Use a Bayesian filtering mechanism (in this case particle filtering) to obtain localization of the AUV

Section 3 introduces Gaussian Process models. Section 2 discusses about the related work in using Gaussian Process models for robot localization and different localization models being used for AUVs. Section 4 applies the Gaussian Process models in current scenario.

2 Related Work

The problem which this project aims to solve is referred to as "Terrain Aided Navigation"(TAN) or "Terrain-Relative Navigation" (TRN)in marine robotics literature. Many different approaches have been made in this area to to let the AUV navigate autonomously. This section gives various approaches which have been used in the past decade. Nygren, et al [1] show a method to accurately determine the AUV's position in an effective way by using a multibeam sonar for three-dimensional sampling of the bottom topography and a linear Kalman filter in a nearly optimal way.

Dektor, et al [2] demonstrated an improvement in TRN technique for localizing a vehicle in underwater environments. Standard TRN algorithms use particle filters for position localization and use the correlation between a priori map noise and vehicle sensor noise when performing measurement updates in the filter. This leads to underestimation of position uncertainty in the regions where terrain information is low and map error is high. They tackled this problem by adjusting the filter variance to depend on the estimated terrain information in addition to map error and sensor error. Claus, 2015 [3] solved the localizing problem using a jittered bootstrap based particle filtering algorithm and it makes use of the vehicle's dead-reckoned navigation solution, onboard altimeter, and a local digital elevation model (DEM). The proposed algorithm estimates the location of vehicle by comparing the water depth estimate measured by the vehicle with a digital elevation model (DEM) of the sea bed.

Barkby, et al [5] addressed the problem of simultaneously localizing and mapping via AUV for a featureless seafloor. The SLAM technique used in this approach follows from the distributed particle mapping(DPM). For weighting the particles for re-sampling, it's weight was based on how well the observed swath matched the previously stored estimate of seabed depth.

All these approaches deliver acceptable results in the environment where they were tested. However, all of them lack the aspect of learning and inferring about the inherent structure of sea-bed. Hence these approaches cannot be used in an environment where inherent structure is not known. This project aims to bridge this gap by learning about the terrain of sea-bed and how the depth at certain locations are correlated. Here, the Gaussian Process regression over the training set of data models the terrain and builds a covariance matrix which tells the inherent correlation at two different locations. The hyper-parameters of this covariance matrix are optimized with MLE estimate of the log-likelihood of s Gaussian Process model. This follows from the work of Jonathan Ko et al [6].

3 Gaussian Process Regression

In the previous section we saw a linear model of regression in equation 2. A simple way to overcome this problem is to project the inputs to high dimensional space using a set of basis functions and then apply the linear model in this high dimensional space instead of directly on the inputs themselves. In this case, the input $\mathbf{x_i}$ could be projected into the space of powers of $\mathbf{x_i}$: $\phi(\mathbf{x_i})$ = $(1, \mathbf{x_i}, \mathbf{x_i^2}, \mathbf{x_i^3}, ..., \mathbf{x_i^n})^T$. Since the projections are fixed functions and independent of the parameters $\mathbf{w}$, the model is still linear and analytically tractable. The equation 2 be rewritten as

$$\mathbf{z_i} = g(\mathbf{x_i}) + \epsilon \qquad g(\mathbf{x_i}) = \mathbf{\Sigma_{j=0}^{n}} \mathbf{w_j} \phi(\mathbf{x_i}) \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_\mathbf{i}^\mathbf{2}) \tag{4}$$

In the MLE/MAP approach of regression as discussed in previous section, this model is parametric in terms of the weights $\mathbf{w_j}$ used to model the problem. Problem with finite parameterization is that either it gives us a simple fit for the training data or a very complex fit if the basis function is projected into very high dimension. Hence there is a need for a non-parametric model which models the depth of sea-bed. One such non-parametric model is a Gaussian Process.

A Gaussian Process can be defined as a collection of random variables, any finite number of which have a joint Gaussian distribution[]. We can interpret them as a simple and general class of models of functions. In a Gaussian Process model the function values $g(\mathbf{x_1})$,$g(\mathbf{x_2})$, $\ldots$ $\mathbf{g}(\mathbf{x_n})$ in equation 4 have a joint Gaussian distribution. A Gaussian Process model, before conditioning on data, is completely specified by its mean function and its covariance function. Mean function $\mu(\mathbf{x})$ and the covariance function $k(\mathbf{x_i}, \mathbf{x_j})$ of a real process $g(\mathbf{x})$ are defined as

$$\mathbb{E}\left[g(\mathbf{x})\right] = \mu(\mathbf{x}) \tag{5}$$

$$\text{Cov}\left[g(\mathbf{x}), g(\mathbf{x}')\right] = k(\mathbf{x}, \mathbf{x}') \tag{6}$$

and a Gaussian Process is defined as

$$g(\mathbf{x}) \ \sim \ \mathcal{GP}(\mu_a, k_a) \tag{7}$$

Since any finite number of random variables have a joint Gaussian distribution e.g. $(g(\mathbf{x}_i), g(\mathbf{x}_j)) \ \sim \mathcal{N}(\mu, \Sigma)$, the the marginal distribution $g(\mathbf{x}_i)$ too is a Gaussian distribution $\mathcal{N}(\mu_i, \Sigma_i i)$. The marginal likelihood under a GP prior of a set of function values $[g(\mathbf{x}_1), g(\mathbf{x}_2), \dots g(\mathbf{x}_N)] := \mathbf{g}(\mathbf{X})$ at locations $\mathbf{X}$ is given by:

$$
\begin{aligned}
p(\mathbf{g}(\mathbf{X})|\mathbf{X}, \mu(\cdot), k(\cdot,\cdot)) &= \mathcal{N}(\mathbf{g}(\mathbf{X})|\mu(\mathbf{X}), k(\mathbf{X},\mathbf{X})) \\
&= (2\pi)^{-\frac{N}{2}} \times \underbrace{|k(\mathbf{X},\mathbf{X})|^{-\frac{1}{2}}}_{\text{controls model capacity}} \\
&\quad \times \underbrace{\exp\left\{-\frac{1}{2}\left(\mathbf{g}(\mathbf{X}) - \boldsymbol{\mu}(\mathbf{X})\right)^{\mathsf{T}} k(\mathbf{X},\mathbf{X})^{-1}\left(\mathbf{g}(\mathbf{X}) - \boldsymbol{\mu}(\mathbf{X})\right)\right\}}_{\text{encourages fit with data}}
\end{aligned} \tag{8}
$$

Using the conditional property of a Gaussian distribution, we could predict the function value $\mathbf{g}(\mathbf{x}_\star)$ at the test location $\mathbf{x}^\star$. The predictive distribution has the form

$$
\begin{aligned}
p(g(\mathbf{x}^\star)|\mathbf{g}(\mathbf{X}), \mathbf{X}, \mu(\cdot), k(\cdot,\cdot)) = \mathcal{N}\big(g(\mathbf{x}^\star) \,|\, &\underbrace{\mu(\mathbf{x}^\star) + k(\mathbf{x}^\star,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}\left(\mathbf{g}(\mathbf{X}) - \mu(\mathbf{X})\right)}_{\text{predictive mean follows observations}}, \\
&\underbrace{k(\mathbf{x}^\star,\mathbf{x}^\star) - k(\mathbf{x}^\star,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}k(\mathbf{X},\mathbf{x}^\star)}_{\text{predictive variance shrinks given more data}}\big)
\end{aligned} \tag{9}
$$

Let's define the mean and covariance of the predictive distribution as follows:

$$
\mu_\star(\cdot) = \mu(\mathbf{x}^\star) + k(\mathbf{x}^\star,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}\left(\mathbf{g}(\mathbf{X}) - \mu(\mathbf{X})\right) \tag{10}
$$

$$
\Sigma_\star = k(\mathbf{x}^\star,\mathbf{x}^\star) - k(\mathbf{x}^\star,\mathbf{X})k(\mathbf{X},\mathbf{X})^{-1}k(\mathbf{X},\mathbf{x}^\star) \tag{11}
$$

The appropriateness of a Gaussian Process model is entirely dependent on the choice of covariance function $k(\mathbf{x_i}, \mathbf{x_j})$, which is user dependent. We must choose a covariance function that relates the points in $\mathbf{X}$. A popular covariance function to model geo-spatial data is radial basis function(RBF) kernel

$$
k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}\frac{(\mathbf{x} - \mathbf{x}')^2}{2\ell^2}\right) \tag{12}
$$

where ℓ is a length scale that determines the strength of correlation between points. As seen from the above equation, when two points are separated by large distance, the covariance will tend to zero and Gaussian Process variance at test points far from measurements will tend to the underlying variance of the function σ_f^2.

The parameters of a Gaussian Process, with RBF as the kernel, $\boldsymbol{\Theta}$ = [σ_f^2, ℓ] are called hyperparameters. These hyperparameters can be learned by MLE of log-likelihood described in equation 8.

4 A GP based Filtering Model for AUV Localization and Mapping

In the equation 1 describing a Bayes filter, the term $p(\mathbf{x_k}|\mathbf{x_{k-1}}, \mathbf{u_{k-1}})$ is the *prediction model*, a probabilistic model of system dynamics. For our AUV and this project, we consider a kinematic model with an additive Gaussian noise for *prediction model*. It has the following form

$$
p(\mathbf{x_k}|\mathbf{x_{k-1}}, \mathbf{u_{k-1}}) \sim \mathcal{N}(u\boldsymbol{\Delta}\mathbf{t}, \sigma_{\mathbf{x}}^{\mathbf{2}}\mathbf{I}) \tag{13}
$$

where $\boldsymbol{\Delta}\mathbf{t}$ is the time difference between two sensor readings at $\mathbf{x_k}$ and $\mathbf{x_{k-1}}$. And *u* is the velocity vector of AUV along its heading, as mentioned in Figure 1. *u* is dependent on the control input received by the AUV at $\mathbf{x_{k-1}}$. $\sigma_{\mathbf{x}}^{\mathbf{2}}$ is the variance of Gaussian noise associated with the estimation of $\mathbf{x_k}$ using kinematic model based on $\mathbf{u_{k-1}}$ and *u*.

The likelihood term $p(\mathbf{z_k}|\mathbf{x_k})$, is also described as the *observation model*. For *observation model* we use the results from section 3. Given the training set $D \in \{(\mathbf{x_0}, \mathbf{z_0}), (\mathbf{x_1}, \mathbf{z_1}), ...(\mathbf{x_m}, \mathbf{z_m})\}$ we fit a

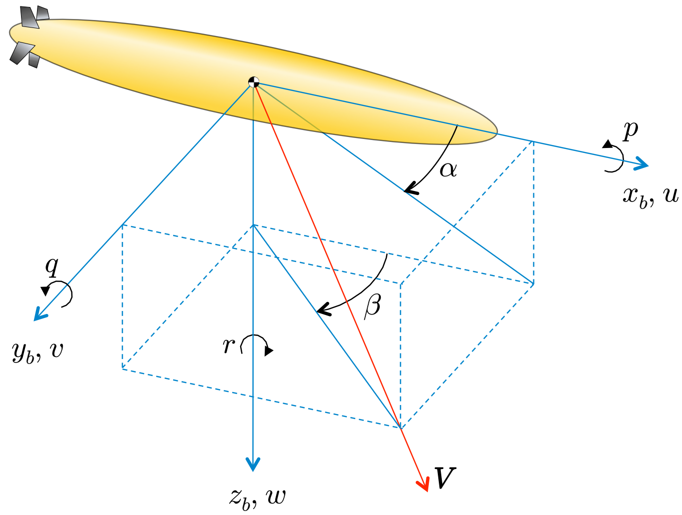


Figure 1: The AUV, or body, frame is illustrated above with body axes xb, yb, and zb.

Gaussian Process regression over it using (4) and (7) with a RBF kernel as defined in (12). Then the predictive distribution given by (9),(10) and (11) would serve as the likelihood or the *observation model* for our bayesian filter. Here relationship between the functional realization $g(\cdot)$ of the Gaussian Process model and observation $\mathbf{z}$ in AUV is given by (4). This gives *observation model* the form

$$p(\mathbf{z_k}|\mathbf{x_k}) \sim \mathcal{N}(\mu_\mathbf{k}, \mathbf{\Sigma_k}) \tag{14}$$

where

$$\mu_\mathbf{k}(\cdot) = \mu(\mathbf{x_k}) + k(\mathbf{x_k}, \mathbf{X})k(\mathbf{X}, \mathbf{X})^{-1}\left(\mathbf{g}(\mathbf{X}) - \mu(\mathbf{X})\right) \tag{15}$$

$$\mathbf{\Sigma_k} = k(\mathbf{x_k}, \mathbf{x_k}) - k(\mathbf{x_k}, \mathbf{X})k(\mathbf{X}, \mathbf{X})^{-1}k(\mathbf{X}, \mathbf{x_k}) \tag{16}$$

and $\mathbf{X} \in (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots \mathbf{x}_{k-1})$.

In the following section we discuss how the above mentioned models can be integrated into particle filtering.

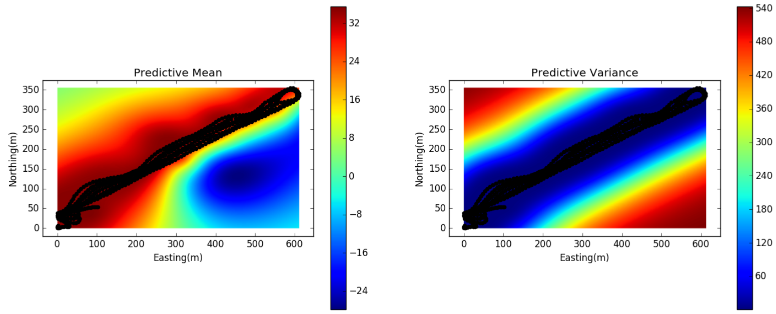
4.1 Gaussian Process Particle Filters

Particle filters are the sample-based implementations of Bayes filters. It is a Bayes filter that works by representing a probability distribution p(x) as a set of samples. The key idea of particle filters is to represent the posteriors over the state $\mathbf{x_k}$ by the sets $\mathcal{X}_{\|}$ of the weighted samples:

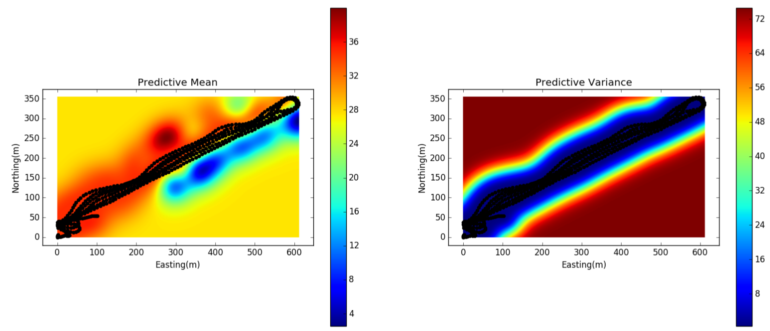
$$\mathcal{X}_k = \{< \mathbf{x}_k^m, w_k^{(m)} > | m = 1,\ldots,\mathrm{M}\} \tag{17}$$

Here each $\mathbf{x}_k^m$ is the m^{th} particle for the k^{th} state. And each $w_k^{(m)}$ is the non-negative numerical factor called *importance weight*. The posteriors of particle filters are updated according to sampling procedure. In this case the particle filtering can be summarized as below:

1. *Sampling*: Particles are sampled for the state, GPS coordinate of AUV, at time k based on the previous state $\mathbf{x}_k^m$ and control $\mathbf{u_{k-1}}$ using the prediction model defined in (13)
2. *Importance Weighting*: After sampling M particles, importance weighting of these particles is carried out where each particle is weighted by the likelihood of the most recent measurement $\mathbf{z_k}$ given the sampled state $\mathbf{x}_k^m$. This likelihood is GP based observation model as represented in (14),(15) and (16).



(a) Matern + Gaussian white noise kernel



(b) Squared-exponential + Bias kernel

Figure 2: Predictive mean and variances of different kernels used to model terrain using GP. We see that Matern kernel has less variance near training data points. Black dots represents training data.

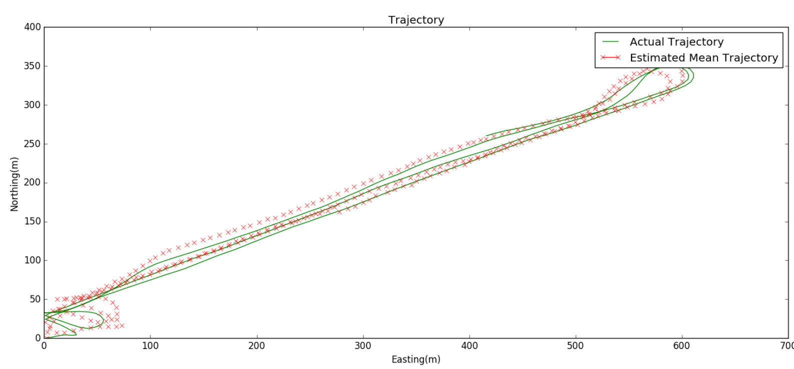


Figure 3: Estimated trajectory using GP-particle filter.

3. *Resampling*: Particles with less weights are thrown away and are replaced by samples with a high weight. This step is necessary since only a finite number of particles are used to approximate a continuous distribution. Furthermore, resampling allows to apply a particle filter in situations in which the true distribution differs from the proposed one.

5 Evaluation

. The evaluation of our method was done offline. Experiments were conducted in Claytor Lake, Pulaski, VA. The training data to model the depth of the lake was collected via May-Craft boat fitted with a depth-sounder for getting geo-referenced depth readings of the lake-bed. Data was not very dense with only 563 data points of geo-referenced depth readings over an area of 241500 m^2. We tried different kernels for fitting and optimizing the Gaussian Process over this 2-d data ((lat,lon) -> (depth)). The Matern kernel with an additional bias kernel gave the minimum variance around the observed data. Figure 2 shows various kernels used for modeling Gaussian Processes with their predictive means and variance.

For testing our localization of the robot, we couldn't actually use AUV since the it has not yet been set up with a Doppler Velocity Log(DVL). Hence, for testing purposes, localization was done on the trajectory of the boat and calculating the velocity vector at each timestamp in direction of motion. Figure 3 shows the estimated trajectory of the boat using this GP-Particle filter based approach. We can see that the estimate is pretty decent.

6 Conclusion

In this project we have shown how a GP based particle filtering technique could be used to localize an AUV underwater as well as map the floor of the water body. The approach of localizing an AUV using GP-based bayes filters provides us with the variance of estimation of location which is modeled according to the terrain of the water body. This may prove beneficial in find a path which lowers the uncertainty of AUV so that it doesn't crash into the shore and remains afloat. This approach can also be beneficial in an informative path planning under uncertainties for various rescue missions on the sea which may be assisted by AUVs.

References

[1] I. Nygren and M. Jansson. Terrain navigation for underwater vehicles using the correlator method. *The IEEE Journal of Oceanic Engineering*, Volume:29 , Issue: 3. (page 2)

[2] S. Dektor and S. Rock. Improving Robustness of Terrain-Relative Navigation for AUVs in Regions with Flat Terrain. *The IEEE Journal of Oceanic Engineering*, Volume:29 , Issue: 3. (page 3)

[3] B. Claus and R. Bachmayer. Terrain-aided Navigation for an Underwater Glider *Journal of Field Robotics*, Volume:32 , Issue: 7. (page 3)

[4] A. Eliazar and R. Parr. DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks *IJCAI'03 Proceedings of the 18th international joint conference on Artificial intelligence*, Pages 1135-1142

[5] S. Barkby, S. Williams, O. Pizarro and M. Jakuba. A featureless approach to efficient bathymetric SLAM using distributed particle mapping *Journal of Field Robotics*, Volume:28 , Issue: 1.

[6] Jonathan Ko and Dieter Fox. GP-BayesFilters: Bayesian Filtering Using Gaussian Process Prediction and Observation Models *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Volume:28 , Issue: 1. (page 3)

(page 3)