Terrain Aided AUV Navigation – A Comparison of the Point Mass Filter and Terrain Contour Matching Algorithms

Kjetil Bergh Ånonsen, Oddvar Hallingstad
University Graduate Center (UniK), Kjeller, Norway
email: anonsen@unik.no

Ove-Kent Hagen, Magne Mandt
Norwegian Defence Research Establishment (FFI), Kjeller, Norway

Abstract

The navigation of most autonomous underwater vehicles (AUVs) and submarines is based on inertial navigation systems. Even with Doppler log aiding, these systems will drift away from the true position, and position fixes are needed. This paper will focus on obtaining submerged position fixes derived from bathymetric measurements, using a prior bathymetric map. We focus on two different terrain aided navigation methods, namely the Point Mass Filter (PMF), a recursive Bayesian method, and the Terrain Contour Matching (TERCOM) algorithm. The paper will focus on AUV navigation, though the methods can also be used in submarines. Terrain aided navigation has been used for decades in aircraft and cruise missiles. For underwater applications, however, documentation of practical results is limited. The paper will report terrain navigation results from real data collected with a HUGIN class AUV, equipped with a Multi Beam Echo sounder (MBE). Characteristics such as accuracy, convergence time and terrain dependency for the two methods are compared. In suitable terrain both methods yield an accuracy of 10-15 m, using a map resolution of 10 m. The overall performance of the PMF is slightly better than that of TERCOM.

1 Introduction

The navigation of most autonomous underwater vehicles (AUVs) and submarines is based on inertial navigation systems (INS). To limit the drift in the INS, aiding sensors such as Doppler Velocity Logs (DVL) or electromagnetic logs are used. The former is common in AUVs, whereas the latter often is used in submarines, due to restrictions on the use of active acoustics. Regardless of the aiding sensors used, the accuracy in these systems will drift off, and position fixes are needed. As GPS signals are blocked by water, GPS fixes are only possible when an antenna is above the water surface. To allow the vehicle to stay submerged for a long period of time, an alternative method for position fixing is needed. For AUVs, one possibility is to use aiding from a mother ship, but this is not possible when the AUV is run in fully autonomous mode. Other possibilities include acoustic ranging to underwater transponders, often referred to as Underwater Transponder Positioning (UTP). In this paper we focus on obtaining fixes from terrain referenced navigation. We focus on AUVs, but the methods discussed may also be used by submarines.

Terrain navigation has been used for decades in air and land navigation systems. However, few results from underwater applications have been published. Some of the issues of underwater terrain navigation are addressed in [10], [12], and [6]. The paper [9] gives an overview of different algorithms, focusing on the use of Multibeam Echo sounder (MBE). Here we focus on two different algorithms for terrain navigation; the Point Mass Filter (PMF) and the Terrain Contour Matching algorithm (TERCOM).

First we introduce the ideas behind the algorithms, before presenting results obtained using real MBE data from a HUGIN class AUV. The HUGIN AUVs have been developed by Kongsberg Maritime and the Norwegian Defense Research Establishment (FFI). A description of the HUGIN integrated navigation system is given in [8]. Figure 1 shows a picture of the Hugin 1000 military AUV, which is used by the Royal Norwegian Navy.

Figure 1: The Hugin 1000 military AUV.

2 The Algorithms

The PMF, a recursive Bayesian algorithm, was first presented for aircraft terrain navigation purposes in [3]. The TERCOM algorithm, [5], was originally patented in 1958, and used in cruise missiles from the 1970s. All terrain navigation algorithms require a certain amount of terrain variation to perform well, and in flat areas, the algorithms are only able to tell that the vehicle is somewhere above the flat area.

2.1 Recursive Bayesian Estimation

2.1.1 System Model (Truth Model)

We consider the following discrete truth model for the motion of our AUV,

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \mathbf{u}_k + \mathbf{v}'_k + \mathbf{v}_k, & (1) \\ \mathbf{v}'_{k+1} &= \mathbf{g}(\mathbf{v}'_k) + \mathbf{v}''_k, & (2) \end{aligned}$$

where $\mathbf{x}_k = (x_N, x_E)^T$ is the AUV position vector, $\mathbf{u}_k$ is the position change calculated from the inertial navigation system, and $\mathbf{v}_k$ and $\mathbf{v}''_k$ are independent white noise sequences. Equation (2) models the strongly correlated error propagation of the inertial navigation system, which we do not specify further here. The system measurement equation is given by

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{w}_k, \tag{3}$$

where the bottom depth measurement $\mathbf{z}_k$ is either a vector or a scalar, depending on the type of sensor considered. For MBE measurements $\mathbf{z}_k$ is vector-valued. The function $\mathbf{h}(\mathbf{x}_k)$ denotes the true sea depth at position $\mathbf{x}_k$, and has to be estimated by a digital terrain map. The sensor measurement noise $\mathbf{w}_k$ is assumed to be white. Due to the strong nonlinearity of the function $\mathbf{h}(\mathbf{x}_k)$ in some terrain types, linearization methods, such as the Extended Kalman Filter, are often not well suited for this problem, [3].

To simplify the mathematical description of our algorithms, we will express our true position as an offset, $\delta\mathbf{x}_k = \mathbf{x}_k - \tilde{\mathbf{x}}_k$, from the INS position estimate $\tilde{\mathbf{x}}_k$. Our process then becomes

$$\begin{aligned} \delta\mathbf{x}_{k+1} &= \delta\mathbf{x}_k + \mathbf{v}'_k + \mathbf{v}_k, & (4) \\ \mathbf{v}'_{k+1} &= \mathbf{g}(\mathbf{v}'_k) + \mathbf{v}''_k. & (5) \end{aligned}$$

2.1.2 Filter Model

Due to the computational requirements of the PMF, we here restrict ourselves to a two-dimensional state vector, representing the horizontal position of the vehicle. As there is no room for additional error states in our state vector, all noises have to be modeled as white, and we therefore introduce a simpler system than that in (3)-(5). To distinguish between our system (truth) and filter models, we attach asterisks to variables in our filter model. Our filter model reads, using the same delta notation as above,

$$\begin{aligned} \delta\mathbf{x}^*_{k+1} &= \delta\mathbf{x}^*_k + \mathbf{v}^*_k, & (6)\\ \mathbf{z}_k &= \mathbf{h}^*(\mathbf{x}^*_k) + \mathbf{w}^*_k = \mathbf{h}^*(\tilde{\mathbf{x}}_k + \delta\mathbf{x}^*_k) + \mathbf{w}^*_k, & (7) \end{aligned}$$

where $\mathbf{v}^*_k$ and $\mathbf{w}^*_k$ are statistically independent white noise sequences. For the noise sequences and the initial position offset, $\delta\mathbf{x}^*_0$, we assume Gaussian distributions, i.e. $p_{\mathbf{w}^*_k}(\mathbf{w}^*_k)$, $p_{\mathbf{v}^*_k}(\mathbf{v}^*_k)$ and $p_{\mathbf{x}^*_0}(\delta\mathbf{x}^*_0)$ are all 2-dimensional Gaussian distributions, though this is not necessary for the PMF to work. We further assume that the noises and initial position are uncorrelated. The function $\mathbf{h}^*(\mathbf{x}^*_k)$ indicates the depth at position $\mathbf{x}^*_k$, given by the digital terrain map. Our terrain map consist of gridded nodes, and the depth values $\mathbf{h}^*(\mathbf{x}^*_k)$ are found by bilinear interpolation. The noise sequence $\mathbf{w}^*_k$ in (7) therefore models both map noise, including interpolation errors and sensor noise. The measurement $\mathbf{z}_k$ is the total sea depth at the footprint of the bathymetric sensor on the sea floor, and it is computed as the sum of the AUV depth, given by a pressure sensor, and the AUV altitude above the sea floor, given by the bathymetric sensor. The noise $\mathbf{w}^*_k$ therefore contains contributions from map errors, pressure sensor noise and bathymetric sensor noise. The main contribution to the pressure sensor noise stems from the pressure-to-depth conversion. Our measured depth must use the same reference as the map database, usually Mean Sea Level, and errors in our tidal estimates can therefore lead to large depth errors. For a detailed analysis of the depth accuracy for the HUGIN class AUVs, we refer to [7]. Other effects that have proven to be important for the accuracy of terrain navigation algorithms, include sensor misalignment errors, sensor lever arm uncertainty and errors related to estimating the sound velocity around the sensor and the sound velocity profile between the vehicle and the sea floor.

2.1.3 The Recursive Bayesian Filter Equations

Let $\mathbb{Z}_k$ be the augmented measurement vector consisting of all the measurements up to time step k. From Bayes formula (see e.g. [1]) and our filter model, (6)-(7), we have

$$p(\delta\mathbf{x}^*_k|\mathbb{Z}_k) = \frac{p_{\mathbf{w}^*_k}(\mathbf{z}_k - \mathbf{h}^*(\tilde{\mathbf{x}}_k + \delta\mathbf{x}^*_k))p(\delta\mathbf{x}^*_k|\mathbb{Z}_{k-1})}{\int\limits_{\mathbb{R}^2} p_{\mathbf{w}^*_k}(\mathbf{z}_k - \mathbf{h}^*(\tilde{\mathbf{x}}_k + \delta\mathbf{x}^*_k))p(\delta\mathbf{x}^*_k|\mathbb{Z}_{k-1})\mathrm{d}\delta\mathbf{x}^*_k}. \tag{8}$$

Our measurement update can then be written as

$$p(\delta\mathbf{x}^*_k|\mathbb{Z}_k) = \alpha^{-1} p_{\mathbf{w}^*_k}(\mathbf{z}_k - \mathbf{h}^*(\tilde{\mathbf{x}}_k + \delta\mathbf{x}^*_k))p(\delta\mathbf{x}^*_k|\mathbb{Z}_{k-1}), \tag{9}$$

where

$$\alpha = \int\limits_{\mathbb{R}^2} p_{\mathbf{w}^*_k}(\mathbf{z}_k - \mathbf{h}^*(\tilde{\mathbf{x}}_k + \delta\mathbf{x}^*_k))p(\delta\mathbf{x}^*_k|\mathbb{Z}_{k-1})\mathrm{d}\delta\mathbf{x}^*_k. \tag{10}$$

The minimum mean square error (MMSE) estimate, [1], and the estimated error covariance matrix is then given by,

$$\begin{aligned} \delta\hat{\mathbf{x}}_k &= E\{\delta\mathbf{x}^*_k|\mathbb{Z}_k\} = \int\limits_{\mathbb{R}^2} \delta\mathbf{x}^*_k p(\delta\mathbf{x}^*_k|\mathbb{Z}_k)\mathrm{d}\delta\mathbf{x}^*_k, & (11)\\ \hat{\mathbf{P}}_k &= \int\limits_{\mathbb{R}^2} (\delta\mathbf{x}^*_k - \delta\hat{\mathbf{x}}^*_k)(\delta\mathbf{x}^*_k - \delta\hat{\mathbf{x}}^*_k)^T p(\delta\mathbf{x}^*_k|\mathbb{Z}_k)\mathrm{d}\delta\mathbf{x}^*_k. & (12) \end{aligned}$$

For the time update of our position distribution we have, from conditioning on the position offset from the previous time step, and using (6),

$$p(\delta\mathbf{x}^*_{k+1}|\mathbb{Z}_k) = \int\limits_{\mathbb{R}^2} p_{\mathbf{v}^*_k}(\delta\mathbf{x}^*_{k+1} - \delta\mathbf{x}^*_k)p(\delta\mathbf{x}^*_k|\mathbb{Z}_k)\mathrm{d}\delta\mathbf{x}^*_k. \tag{13}$$

Given the distribution of the initial position offset, $p(\delta\mathbf{x}^*_0)$, the equations (8) and (13) can now be used recursively to obtain the posterior distribution of the position offsets at each time step. The integrals in the equations are, however, not analytically solvable, and we therefore need to evaluate these integrals numerically.

2.2 The Point Mass Filter

First we choose a search area around our INS position estimate $\tilde{\mathbf{x}}_k$, e.g. ±3σ in each direction. We start the algorithm by discretization of our two-dimensional search area into a grid with M and N grid points in each direction. At each grid point we approximate the density $p(\delta\mathbf{x}^*_k|\mathbb{Z}_k)$ by a probability mass weight $p(\delta\mathbf{x}^*_k(i,j)|\mathbb{Z}_k)_{i=1,\ldots,M,\ j=1,\ldots,N}$. For simplicity we here consider uniform grids only, with grid resolution Δ in both directions. We start by letting

$$p(\delta\mathbf{x}^*_0(i,j)|\mathbb{Z}_0) = p(\delta\mathbf{x}^*_0(i,j)) = \alpha_0^{-1}p(\delta\mathbf{x}^*_0)|_{\delta\mathbf{x}^*_0=\delta\mathbf{x}^*_0(i,j)},$$

where $\alpha_0 = \sum_{m=1}^{M}\sum_{n=1}^{N} p(\delta\mathbf{x}^*_0)|_{\delta\mathbf{x}^*_0=\delta\mathbf{x}^*_0(i,j)}\Delta^2$. Notice the difference between $p(\delta\mathbf{x}^*_0(i,j))$, which is a point-wise probability mass approximation and $p(\delta\mathbf{x}^*_0)$, which is a continuous probability density function. Also notice that the point masses sum to one. For each time step, the measurement update is performed as

$$\begin{aligned} p(\delta\mathbf{x}^*_k(i,j)|\mathbb{Z}_k) &= \alpha_k^{-1}p_{w^*_k}(\mathbf{z}_k - \mathbf{h}^*(\tilde{\mathbf{x}}_k + \delta\mathbf{x}^*_k(i,j)))p(\delta\mathbf{x}^*_k(i,j)|\mathbb{Z}_{k-1}), & (14)\\ \alpha_k &= \sum_{i=1}^{M}\sum_{j=1}^{N} p_{w^*_k}(\mathbf{z}_k - \mathbf{h}^*(\tilde{\mathbf{x}}_k + \delta\mathbf{x}^*_k(i,j)))p(\delta\mathbf{x}^*_k(i,j)|\mathbb{Z}_{k-1})\Delta^2, & (15)\end{aligned}$$

and we are now able to compute our estimated position offset and its covariance

$$\begin{aligned} \delta\hat{\mathbf{x}}_k &= \sum_{i=1}^{M}\sum_{j=1}^{N} \delta\mathbf{x}^*_k(i,j)p(\delta\mathbf{x}^*_k(i,j)|\mathbb{Z}_k)\Delta^2, & (16)\\ \hat{\mathbf{P}}_k &= \sum_{i=1}^{M}\sum_{j=1}^{N} (\delta\mathbf{x}^*_k(i,j) - \delta\hat{\mathbf{x}}_k)(\delta\mathbf{x}^*_k(i,j) - \delta\hat{\mathbf{x}}_k)^T p(\delta\mathbf{x}^*_k(i,j)|\mathbb{Z}_k)\Delta^2. & (17)\end{aligned}$$

We then proceed to the time update, which is computed by

$$p(\delta\mathbf{x}^*_{k+1}(i,j)|\mathbb{Z}_k) = \sum_{m=1}^{M}\sum_{n=1}^{N} p_{\mathbf{v}^*_k}(\delta\mathbf{x}^*_{k+1}(i,j) - \delta\mathbf{x}^*_k(m,n))p(\delta\mathbf{x}^*_k(m,n)|\mathbb{Z}_k)\Delta^2. \tag{18}$$

This equation can be viewed as a two-dimensional convolution, which in general is computationally burdensome. As suggested in [3], we here assume the process noise to be independent and identically distributed in both directions, such that the two-dimensional convolution may be computed from two one-dimensional convolutions. For AUVs, however, this assumption is not necessarily true, and this model should be improved upon in future work.

2.3 The TERCOM Algorithm

The Terrain Contour Matching Algorithm was one of the first algorithms developed for terrain navigation, and it is still used in several applications today, such as in cruise missiles. The TERCOM algorithm is well-suited for batch processing, where the measurements from a certain profile along the trajectory are processed in batch, but it may also be run recursively. For multiple-measurement sensors, such as MBE, batch processing is natural. The method implicitly assumes constant position offsets from the INS position each time it is started. The algorithm should therefore be run continuously for limited time periods only and frequently be restarted. This procedure is well-suited for calculating position fixes for the navigation system.

We divide our search area into a candidate grid $\{\delta\mathbf{x}^*(i,j)\}_{i=1,\dots,M,j=1,\dots,N}$ and seek the position offset candidate that minimizes a certain criterion related to the difference between the measured depth and the map depth for the candidate position. We here use the classical mean absolute distance (MAD) criterion. For vector measurements, the TERCOM estimate is given by

$$\delta\hat{\mathbf{x}}_k = \arg \min_{\delta\mathbf{x}^*(i,j)} \frac{1}{kN_b} \sum_{l=1}^{k} \sum_{m=1}^{N_b} \left| \mathbf{z}_{l,m} - \mathbf{h}_m^*(\tilde{\mathbf{x}}_l + \delta\mathbf{x}^*(i,j)) \right|, \tag{19}$$

where $\mathbf{z}_{l,m}$ is the measurement of beam number m at time step l, N_b is the number of beams and $\mathbf{h}_m^*$ the map depth corresponding to this beam. By plotting the value of the double sum in (19) for all i, j, we get the so-called correlation surface.

The TERCOM algorithm in its original form has no intrinsic uncertainty measure or variance, which is needed if the estimate is to be used in a Kalman filter. It is possible to construct an artificial variance, for instance from the correlation surface, [2]. Also note that the TERCOM algorithm does not utilize the statistical properties of the problem. TERCOM is also less computationally demanding than the PMF.

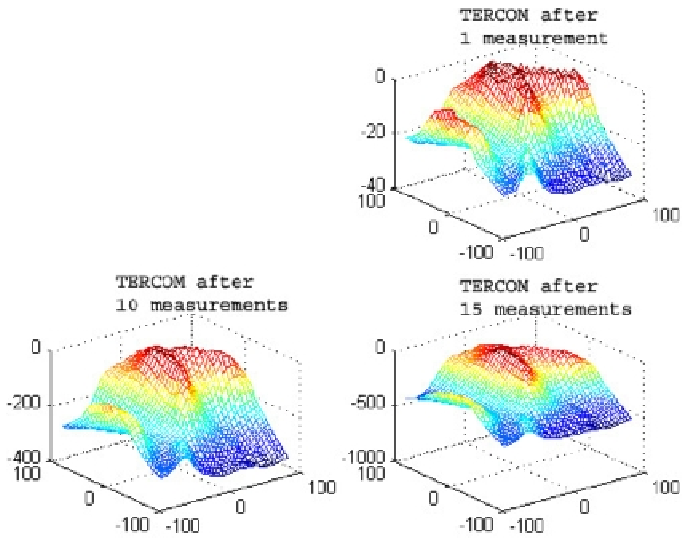

Figure 2: Examples of TERCOM correlation surfaces.

Figures 2 and 3 show examples of the TERCOM correlation surface and the approximated posterior PMF distribution. The same measurement data were used for both figures. The TERCOM correlation surface has been sign-shifted, such that higher values correspond to better matches.

3 Computational Results

We here present computational results obtained from the PMF and TERCOM algorithms, using real HUGIN data, collected by a HUGIN 1 vehicle on November 29, 2001, in an area in the Oslo Fjord. We will use bathymetric measurements from a Kongsberg Maritime EM3000 (300 kHz) MBE, combined with AUV depth measurements from a Digiquartz pressure sensor. MBEs are ideal sensors for terrain referenced navigation, as they measure the depth in a wide swath beneath the vehicle, [9]. The EM3000 uses up to 127 beams in each ping, but in this particular data set only 92 beams were used.

Figure 4 shows a contour map of the area in local coordinates, together with the AUV trajectory computed from NavLab post-processing (smoothing), [4], of the real-time DVL, IMU, and DGPS/USBL data, [8]. The accuracy of this trajectory is expected to be within 1 m (1σ),

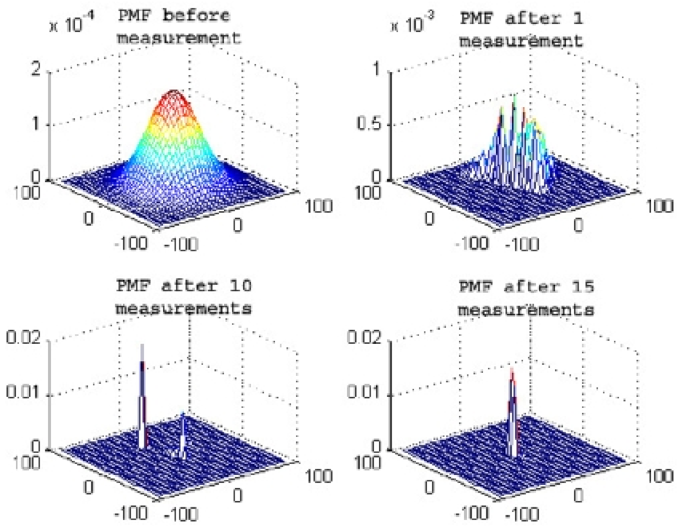

Figure 3: Examples of approximated posterior PMF densities.

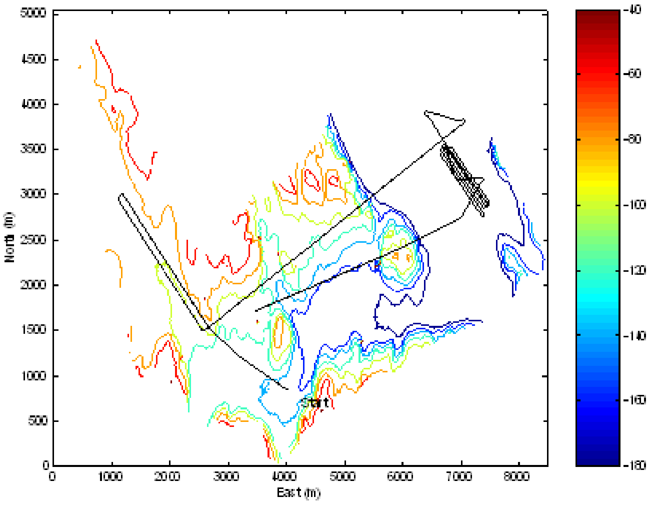

Figure 4: Map of the area and AUV trajectory.

and it will be used as ground truth throughout this paper. The map was constructed from depth measurements from an EM1002 MBE, collected by a surface vessel, and it is therefore statistically independent of the bathymetric measurements from our AUV. The EM1002 measurements were used to build a 10 m gridded UTM depth map. The AUV moves from an area with relatively large depth variations, to a relatively flat area, where it travels in a lawn mower pattern, and back to the rougher area again. Figure 5 shows the sea depth along the vehicle trajectory, as measured from the AUV's DVL and pressure sensor. Extensive MATLAB tests of the two algorithms were done. In our implementation of the PMF, we used an adaptive grid approach, [3], utilizing the fact that many of the point masses will be very close to zero. By setting all point masses below a certain threshold to zero at each time step, we can effectively reduce the number of computations needed, using sparse matrices. Each time the number of non-zero point masses gets above or below certain limits, the PMF grid is decimated or refined, such that a fine grid is used in areas with low estimate uncertainty and a coarser grid is used in areas where the uncertainty is higher. However, as the map has a resolution of 10 m, the PMF grid resolution is not allowed to get below a certain limit. The adaptive grid approach is very useful when performing terrain referenced navigation for large areas with high initial uncertainty, but for smaller areas, like the one we treat here, it is not strictly necessary. A real-time version of the

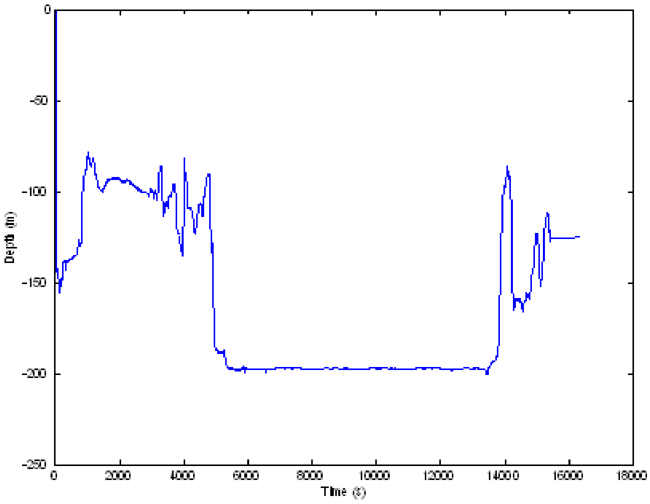

Figure 5: Total sea depth along trajectory, from DVL and pressure sensor.

Time/ s	Terrain properties	PMF error/ m	TCM error/ m
644-1644	Rough, valley	6-13	8-13
1644-2843	Valley	15-82	15-86
2843-5443	Rough	6-14	6-25
5443-13440	Flat	30-120	30-120
13440-1441	Rough	7-15	7-20

Table 1: Summary of results from MBE data in different parts of the run.

PMF, without adaptive grid, has been implemented for the Hugin class vehicles, [9]. In real-time applications the algorithms are typically restarted frequently, and an estimate is computed from a series of MBE pings, so we followed this approach in our tests. To test the sensitivity of the algorithms to errors in the a priori estimates, an artificial error was added to our initial INS estimate.

The measurement at each MBE ping is a vector $\mathbf{z}_k$, consisting of measurements from each beam. As each of these measurements are done using the same sound velocity profile, and they are all added to the same AUV depth estimate from the pressure sensor, there will be correlations between the beams within each ping. Some of the beams will also use the same map nodes to calculate the expected depth, leading to correlations both within and between pings. To model the within-ping correlations, cross terms were added to the measurement covariance matrix in the PMF. There are also other correlations between subsequent pings, e.g. from tidal effects and pressure-to-depth conversion, but since we only deal with a two-dimensional PMF, these effects are disregarded. One possible solution for dealing with such time correlations is to operate the algorithms on relative depth measurements only, by subtracting the mean from all the measurements, [10]. In self-similar terrain, however, this is unfavourable, and we have not used this approach here.

Table 1 shows a summary of the results obtained by the two algorithms in different areas of the run. We used a measurement update frequency of 0.5 Hz, and the algorithms were run on terrain profiles consisting of 100 pings each time they were restarted, corresponding to a profile length of around 400 m. Position estimates from the real-time navigation system, with an added error of 100 m in each direction were used as initial estimates. A 5 m grid resolution was used for TERCOM, whereas the minimum resolution of the adaptive PMF grid was set to 2 m. The final error after 100 time steps was computed as the total horizontal error of the estimate compared to ground truth. The errors listed in Table 1 are the minimum and maximum errors obtained in the different areas. In most of the runs, except in the flat area, the errors were closer to the minimum values. There were, however, occasional false fixes (wild points), with low estimated PMF variance (respectively stable TERCOM estimates) but large errors, even in the well-suited areas.

Both algorithms performed quite well in the rough areas, with typical final errors around the map resolution (10 m), and they converged rather quickly to a stable estimate, typically after 30-40 time steps, sometimes as few as 2-3 time steps. In this area the estimates from each method after convergence were quite similar, though the nature of their estimates was very different. Figure 6 shows the errors from one of the tests in the rough terrain. Notice that the PMF estimate is much smoother than the TERCOM estimate, which has sudden jumps even after 90 time steps.

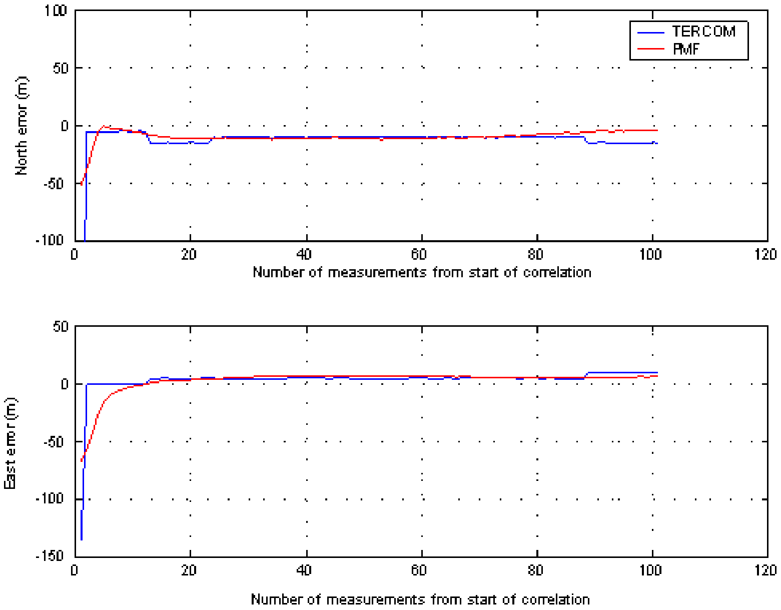

Figure 6: Errors, 3797-3997 s from start of run.

As expected, both algorithms performed poorly in the flat area, see Figure 7. Some of the major disadvantages of the TERCOM algorithm were revealed here. The TERCOM estimate is very unstable, and it has no mechanism for indicating that something is wrong. In contrast, PMF shows a much smoother behavior, and the covariance matrix reflects the uncertainty in the estimate. Some of the same problems were also evident in the 'valley' area, where the error in the principal direction along the valley was large, a known phenomenon for terrain navigation in valleys or near beaches, [10]. Again, the PMF was able to detect this uncertainty through its covariance matrix.

A problem with the PMF throughout all of our tests, is that the estimate is overconfident. Though it is able to reflect uncertainty in the estimate in different areas, as described above, the estimated variance in each direction is generally too low. For example, in the rough area, where the accuracy of the estimates are typically around 10 m, the estimated standard deviation in each direction is typically as low as 2 m. This is a serious problem if the PMF estimate is to be integrated in a Kalman filter. Simulations, where process noise and measurement noise were sampled from known distributions, suggest that this inconsistency stems from mismatch between our filter model and the true system. In particular, map noise and sensor noise from the MBE is very complicated, consisting of a number of different contributions, and this is subject to ongoing research.

As mentioned above, we used the adaptive grid approach in our PMF implementation. Tests done without grid adaptation show that the quality of the estimates does not suffer significantly from this approach, as long as the truncation threshold is not chosen too high. There is, however, a slightly higher risk of converging to a wrong estimate, and in real system this approach should be used with care.

We emphasize that similar tests to the ones described here have been done with other data sets from the same test area, yielding very similar results. Both methods are robust to errors in the initial position given by the navigation system, as long as the correct position is within the search area.

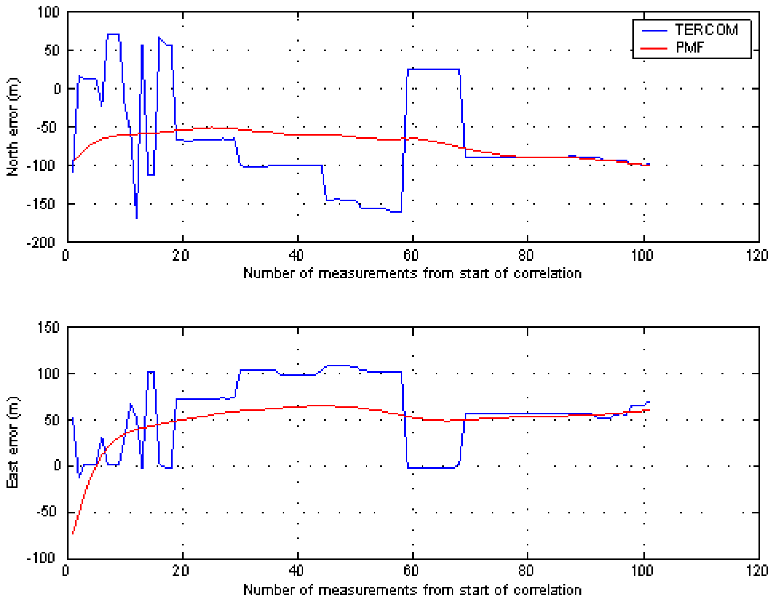

Figure 7: Errors, 5397-5597 s from start of run.

4 Conclusions and Future Work

We have seen how the Point Mass Filter and Terrain Contour Matching algorithms can be used for terrain referenced underwater navigation. In suitable terrain, both methods yield position estimates with an accuracy around that of the map resolution. The PMF has an overall better performance than TERCOM, especially in less suitable terrain, since it is able to indicate its own uncertainty through its covariance matrix. However, the PMF has a tendency of overconfidence, and this should be further investigated in future work. Another natural extension is to extend the PMF to three dimensions by including depth errors, e.g. due to tidal effects and pressure to depth conversion. This can also be done using particle filters, or a combination of particle filters and Kalman filtering, through Rao-Blackwellization, [11].

References

[1] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, New York, 2001.

[2] O. Bergem. *Bathymetric Navigation of Autonomous Underwater Vehicles Using a Multibeam Sonar and a Kalman Filter with Relative Measurement Covariance Matrices*. PhD thesis, Department of Informatics and Computer Science, College of Arts and Science, University of Trondheim, Norway, 1993.

[3] N. Bergman. *Recursive Bayesian Estimation - Navigation and Tracking Applications*. PhD thesis, Department of Electrical Engineering, Linköping University, Sweden, 1999.

[4] K. Gade. NAVLAB – Overview and User Guide. Technical Report FFI/RAPPORT-2003/02128, Norwegian Defense Research Establishment, 2003.

[5] J.P. Golden. Terrain contour matching (TERCOM): A cruise missile guidance aid. In T.F. Wiener, editor, *Image Processing for Missile Guidance*, volume 238, pages 10–18, San Diego, CA, 1980. The Society of Photo-Optical Instrumentation Engineers.

[6] O.K. Hagen and P.E. Hagen. Terrain Referenced Integrated Navigation System for Underwater Vehicles. In E. Bovio, R. Tyce, and H. Schmidt, editors, *SACLANTCEN Conference Proceedings CP-46*, pages 171–180, La Spezia, Italy, 2000. NATO SACLANT Undersea Research Centre.

[7] B. Jalving. Depth Accuracy in Seabed Mapping with Underwater Vehicles. In *Proceedings from Oceans '99*, Seattle, WA, 1999.

[8] B. Jalving, K. Gade, O.K. Hagen, and K. Vestgård. A Toolbox of Aiding Techniques for the HUGIN AUV Integrated Inertial Navigation System. In *Proceedings from Oceans 2003*, San Diego, CA, 2003.

[9] B. Jalving, M. Mandt, O.K. Hagen, and F. Pøhner. Terrain referenced navigation of AUVs and submarines using multibeam echo sounders. In *Procedings from UDT Europe 2004*, Nice, France, 2004.

[10] M. Mandt. Terrengreferert posisjonering av undervannsfarkoster. Technical Report FFI/RAPPORT-2001/05900, Norwegian Defense Research Establishment, 2001.

[11] Per-Johan Nordlund. *Sequential Monte Carlo Filters and Integrated Navigation.* PhD thesis, Department of Electrical Engineering, Linköping University, Sweden, 2002.

[12] I. Nygren and M. Jansson. Terrain Navigation Using the Correlator Method. In *Position Location and Navigation Symposium, PLANS 2004*, Monterey, CA, April 2004.