Bayesian Terrain-Based Underwater Navigation Using an Improved State-Space Model

Kjetil Bergh Ånonsen
Department of Engineering Cybernetics,
Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
University Graduate Center, NO-2027 Kjeller, Norway
e-mail:anonsen@unik.no

Oddvar Hallingstad
Department of Engineering Cybernetics,
Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
University Graduate Center, NO-2027 Kjeller, Norway

Ove Kent Hagen
Norwegian Defence Research Establishment (FFI), NO-2027 Kjeller, Norway

Abstract— **This paper focuses on terrain aided underwater navigation as a means of aiding an inertial navigation system. It is assumed that a prior map is present and Bayesian methods are used to estimate the position of the vehicle. Traditionally this has been done using a crude low-dimensional model in the Bayesian filters. An improved state-space model is introduced, implemented in a particle filter/sequential Monte Carlo filter and tested on real AUV (autonomous underwater vehicle) data. Compared to conventional filter models, the new model yields smoother, slightly more accurate results, though problems with overconfidence occur.**

I. Introduction

This paper focuses on the development of an improved state-space model for underwater terrain based navigation compared to that used in [1], [2]. The focus is on AUVs (autonomous underwater vehicles), although the techniques presented can also be used by other underwater vehicles, like remotely operated vehicles (ROVs) and submarines.

Most AUV navigation systems are based on inertial navigation, see e.g. [3]. Inertial navigation systems drift off with time, even when velocity aiding is used. In order to allow extensive submerged operations, additional position fixes are needed. As GPS is not available under water, one is often dependent on acoustic aiding of the vehicle, either from a mother ship or from using underwater acoustic transponders. In order to increase the autonomy of the vehicle and avoid costly pre-deployment of underwater transponders, terrain-based navigation is a favorable alternative. As an AUV in many cases carries a bathymetric sensor, it is natural to utilize bathymetric information in the navigation of the vehicle. For the methods presented here we assume that there exists a prior bathymetric map database of the area. We use a multibeam echo sounder (MBE) as bathymetric sensor. MBEs are very well suited for terrain-based navigation, as a large area behind the vehicle is covered by the sensor.

Terrain-based navigation has been used for decades in aircraft and cruise missiles [4], [5], but for underwater vehicles the technique is rather new, though some papers have been published over the last few years, e.g. [6], [7], [2]. A variety of different terrain-based navigation methods have been proposed in the literature. Among the more sophisticated are the Bayesian methods, in which the position of the vehicle is estimated using a state-space model. Due to the strong nonlinearity of the measurements, Kalman filter-based methods have proven not well suited in most cases. Instead nonlinear Bayesian methods like point mass (PMF) and particle filters (PF) have been successfully applied to underwater navigation [5], [7]. Due to the high computational complexity of these methods, a crude, low-dimensional state-space model is used, often with a two or three dimensional state-space vector. In reality, the process and measurement equations are much more complicated than what is described in these low-dimensional models. This discrepancy has often proven to yield non-consistent estimate covariances [2]. The estimated standard deviations are typically smaller than the true estimation errors, which is unfavorable when the estimated position is to be integrated into the AUV navigation system.

In this paper we extend the state vector, incorporating more information from our knowledge of the underlying system. An improved state-space model is presented, and the model is tested using Bayesian methods on real AUV data collected by a HUGIN vehicle equipped with an MBE. The HUGIN AUVs are manufactured by Kongsberg Maritime AS.

II. The Estimation Methods

We here present the different models that we use for our terrain navigation algorithms. First we introduce the different coordinate systems used in the models.

A. Coordinate Systems

Table I gives an overview of the various coordinate systems used in this paper.

B. System Model

We first present the system (truth) model. We consider the following generic model for the motion of our AUV,

TABLE I

COORDINATE SYSTEMS USED IN THIS PAPER.

Symbol	Description
e	Earth centered, earth fixed coordinate system
n	Local north, east down (NED) coordinate system
b	Body fixed coordinate system
b'	Body fixed, roll and pitch compensated coordinate system
m	Map coordinate system (earth fixed)

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{v}_k + \boldsymbol{u}_k + \boldsymbol{v}'_k, \tag{1}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{g}(\boldsymbol{v}_k) + \boldsymbol{v}''_k, \tag{2}$$

where $\boldsymbol{x}_k = (x_{N,k}, x_{E,k})^T$ is the horizontal AUV position vector (north, east) decomposed in the $\{e\}$-frame, $\boldsymbol{u}_k$ is the position change from time step k to $k+1$, calculated from the inertial navigation system, and $\boldsymbol{v}'_k$ and $\boldsymbol{v}''_k$ are white noise sequences. Equation (2) models the strongly correlated error propagation of the inertial navigation system. Note that we do not specify the dimension of $\boldsymbol{v}_k$ nor the function $\boldsymbol{g}$, so this generic model can be made arbitrarily complex. The system measurement equation is given by

$$\boldsymbol{z}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{w}_k. \tag{3}$$

In [2] the problem of estimating the depth bias in the measurement was treated, and it was shown that by adding a separate state for the depth bias more accurate and robust methods were obtained in the presence of unknown depth biases. In this paper we assume that the dept bias has been properly compensated for in our data and estimate solely the horizontal AUV position. If necessary, a depth bias state can of course be added to the models presented here too, at the cost of slightly more computationally demanding algorithms.

Since we are dealing with MBEs, the bottom depth measurement $\boldsymbol{z}_k$ is a vector measurement. The function $\boldsymbol{h}(\boldsymbol{x}_k)$ denotes the true mean sea depth at position $\boldsymbol{x}_k$. The term $\boldsymbol{w}_k$ denotes the sensor measurement noise, which is assumed to be white.

In order to simplify the mathematical description of our algorithms, we will express our true position as an offset, $\delta\boldsymbol{x}_k$, from the position estimate $\tilde{\boldsymbol{x}}_k$ of the INS. Our process then becomes, using that $\tilde{\boldsymbol{x}}_{k+1} = \tilde{\boldsymbol{x}}_k + \boldsymbol{u}_k$,

$$\delta\boldsymbol{x}_k = \boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k, \tag{4}$$

$$\delta\boldsymbol{x}_{k+1} = \delta\boldsymbol{x}_k + \boldsymbol{v}_k + \boldsymbol{v}'_k, \tag{5}$$

$$\boldsymbol{v}_{k+1} = \boldsymbol{g}(\boldsymbol{v}_k) + \boldsymbol{v}''_k, \tag{6}$$

$$\tag{7}$$

with the measurement equation

$$\boldsymbol{z}_k = \boldsymbol{h}(\tilde{\boldsymbol{x}}_k + \delta\boldsymbol{x}_k) + \boldsymbol{w}_k. \tag{8}$$

C. Conventional Filter Model

Due to the computational requirements of the Bayesian methods we want to use for solving our terrain navigation problem, we have to restrict ourselves to a simpler model than that described above. This is particularly the case for the point mass filter [5], in which a maximum of 3 states is practically feasible. For the particle filter, the dimension is of less importance, but the number of particles needed for convergence increases with the number of states [8].

Conventionally, the filter model used in terrain navigation is formulated directly in the $\{e\}$ coordinate frame, and the drift is assumed independent in the north and east directions, respectively. This is of course a major assumption, as the drift is more likely to be independent along the surge and sway directions of the vehicle.

The fact that we use a different model in the filter from that in the true system is in itself a major source of error in the terrain navigation estimates. To distinguish between the system and filter models, we attach asterisks to variables in the filter model. Our filter model reads, using the same delta notation as in (4),

$$\delta\boldsymbol{x}^*_k = \boldsymbol{x}^*_k - \tilde{\boldsymbol{x}}_k, \tag{9}$$

$$\delta\boldsymbol{x}^*_{k+1} = \delta\boldsymbol{x}^*_k + \boldsymbol{v}^*_k, \tag{10}$$

$$\begin{aligned}\boldsymbol{z}_k &= \boldsymbol{h}^*(\boldsymbol{x}^*_k) + \boldsymbol{w}^*_k \\ &= \boldsymbol{h}^*(\tilde{\boldsymbol{x}}_k + \delta\boldsymbol{x}^*_k) + \boldsymbol{w}^*_k,\end{aligned} \tag{11}$$

where $\delta\boldsymbol{x}^*_k = (\delta x_{N,k}, \delta x_{E,k})^T$ and $\boldsymbol{x}_k = (x_{N,k}, x_{E,k})^T$, with the assumptions

$$p(\boldsymbol{v}^*_k, \boldsymbol{v}^*_l) = p(\boldsymbol{v}^*_k)p(\boldsymbol{v}^*_l), \tag{12}$$

$$p(\boldsymbol{w}^*_k, \boldsymbol{w}^*_l) = p(\boldsymbol{w}^*_k)p(\boldsymbol{w}^*_l), \tag{13}$$

giving noise sequences that are statistically independent from time step to time step. In (10) $\boldsymbol{v}_k$ and $\boldsymbol{v}'_k$ have been replaced by the white noise sequence $\boldsymbol{v}^*_k$. We also need to specify the densities of the noise sequences and the initial position offset, $\delta\boldsymbol{x}^*_0$. A convenient, but not necessary, assumption is to assume Gaussian densities,

$$p(\delta\boldsymbol{x}^*_0) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}^*_0), \tag{14}$$

$$p(\boldsymbol{v}^*_k) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^*_k), \tag{15}$$

$$p(\boldsymbol{w}^*_k) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}^*_k). \tag{16}$$

Equation (14) indicates that the initial position has a Gaussian density centered around the estimated INS position $\tilde{\boldsymbol{x}}_0$. We further assume that the process noise, measurement noise and initial position are uncorrelated. The function $\boldsymbol{h}^*(\boldsymbol{x}^*_k)$ indicates the depth at position $\boldsymbol{x}^*_k$ and is given by the digital terrain map. We use terrain maps consisting of gridded nodes, and the depth values given by $\boldsymbol{h}^*$ are found by bilinear interpolation over the grid. The noise sequence $\boldsymbol{w}^*_k$ in (11) therefore models both map noise (including interpolation errors) and the sensor noise. The measurement $\boldsymbol{z}_k$ is the total sea depth at the current AUV position, and it is computed as the sum of the AUV depth, given by a pressure sensor, and the AUV altitude above the MBE footprint on the sea floor for each MBE beam. The noise sequence $\boldsymbol{w}^*_k$ therefore contains contributions from map errors, pressure sensor noise and bathymetric sensor noise. For a detailed analysis of the depth accuracy of the HUGIN class AUVs, we refer to [9].

D. 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. [10]) and our filter model, (9)–(11), we have

$$\begin{aligned} p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_k) &= \frac{p(\boldsymbol{z}_k|\delta\boldsymbol{x}_k^*,\mathbb{Z}_{k-1})p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_{k-1})}{p(\boldsymbol{z}_k|\mathbb{Z}_{k-1})} \\ &= \alpha_k^{-1} p_{\boldsymbol{w}_k^*}(\boldsymbol{z}_k - \boldsymbol{h}^*(\tilde{\boldsymbol{x}}_k + \delta\boldsymbol{x}_k^*)) \\ &\quad \cdot p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_{k-1}) \end{aligned} \tag{17}$$

where

$$\begin{aligned} \alpha_k = \int_{\mathbb{R}^2} \big[& p_{\boldsymbol{w}_k^*}(\boldsymbol{z}_k - \boldsymbol{h}^*(\tilde{\boldsymbol{x}}_k + \delta\boldsymbol{x}_k^*)) \\ & \cdot p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_{k-1})\big] \mathrm{d}\delta\boldsymbol{x}_k^*. \end{aligned}$$

The minimum mean square error (MMSE) [10] estimate is then given by

$$\begin{aligned} \delta\hat{\boldsymbol{x}}_k &= E\{\delta\boldsymbol{x}_k^*|\mathbb{Z}_k\} \\ &= \int_{\mathbb{R}^2} \delta\boldsymbol{x}_k^* p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_k)\mathrm{d}\delta\boldsymbol{x}_k^*, \end{aligned} \tag{18}$$

with the covariance matrix

$$\begin{aligned} \hat{\boldsymbol{P}}_k = \int_{\mathbb{R}^2} \big[& (\delta\boldsymbol{x}_k^* - \delta\hat{\boldsymbol{x}}_k^*)(\delta\boldsymbol{x}_k^* - \delta\hat{\boldsymbol{x}}_k^*)^T \\ & \cdot p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_k)\big] \mathrm{d}\delta\boldsymbol{x}_k^*. \end{aligned} \tag{19}$$

For the time update of our position density we have,

$$\begin{aligned} p(\delta\boldsymbol{x}_{k+1}^*|\mathbb{Z}_k) &= \int_{\mathbb{R}^2} p(\delta\boldsymbol{x}_{k+1}^*, \delta\boldsymbol{x}_k^*|\mathbb{Z}_k)\mathrm{d}\delta\boldsymbol{x}_k^* \\ &= \int_{\mathbb{R}^2} p(\delta\boldsymbol{x}_{k+1}^*|\delta\boldsymbol{x}_k^*, \mathbb{Z}_k) \\ &\qquad \cdot p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_k)\mathrm{d}\delta\boldsymbol{x}_k^* \\ &= \int_{\mathbb{R}^2} p_{\boldsymbol{v}_k^*}(\delta\boldsymbol{x}_{k+1}^* - \delta\boldsymbol{x}_k^*) \\ &\qquad \cdot p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_k)\mathrm{d}\delta\boldsymbol{x}_k^*. \end{aligned} \tag{20}$$

Given the density of the initial position, $p(\delta\boldsymbol{x}_0^*)$, (17) and (20) can now be used recursively to obtain the density of the position offsets for each time step. However, the integrals in the equations are not analytically solvable, and we therefore need to evaluate these integrals numerically. We will here use a particle filter (sequential Monte Carlo filter) for the evaluation of these equations. The particle filter gives estimates both of the vehicle position and its covariance. If the estimates are to be integrated into the inertial navigation system, it is crucial that the estimated covariance is accurate as possible to avoid inconsistencies in the system.

E. The Bayesian Bootstrap Particle Filter

Particle filtering has gained attention the recent years for estimation in nonlinear systems, in areas such as signal processing, tracking and navigation, to name a few. Introductions to the principles of particle filtering can be found in [11], [12]. We will here use the Bayesian Bootstrap particle filter, which is one of the simplest particle filter algorithms, first presented in [13]. The algorithm estimates the posterior density $p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_k)$ at time step k by a set of particles, $\{\delta\boldsymbol{x}_k^i\}_{i=1,\ldots,N}$, where N denotes the number of particles. The posterior density can be written as

$$p(\delta\boldsymbol{x}_k^*|\mathbb{Z}_k) \approx \frac{1}{N}\sum_{i=1}^{N} \delta\left(\delta\boldsymbol{x}_k^* - \delta\boldsymbol{x}_k^i\right), \tag{21}$$

where $\delta(\cdot)$ is the (2-dimensional) Dirac delta function. At each time step, starting with the particle set $\{\delta\boldsymbol{x}_k^i\}_{i=1,\ldots,N}$, each of the particles are first predicted independently, by propagating them through the filter process equation (10). In our case this means adding a noise sampled from the process noise distribution, yielding the new particle set $\left\{\delta\boldsymbol{x}_{k+1}'^i\right\}_{i=1,\ldots,N}$. For each particle, a likelihood weight $w_{k+1}^i \propto p(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}'^i)$ is computed, using the measurement $\boldsymbol{z}_{k+1}$. The weights are then normalized, such that $\sum_{i=1}^N w_{k+1}^i = 1$. The weighted particle set $\left\{\delta\boldsymbol{x}_{k+1}'^i, w_{k+1}^i\right\}_{i=1,\ldots,N}$ is now resampled with replacement N times, with the probability w_{k+1}^j of resampling particle $\delta\boldsymbol{x}_{k+1}'^j$. We here use the systematic resampling procedure described in [11]. This completes the cycle, and we now have a particle set approximately distributed according to $p(\delta\boldsymbol{x}_{k+1}^*|\mathbb{Z}_{k+1})$.

The reason why we have chosen the Bayesian Bootstrap algorithm, is that it is one of the simplest particle filters to implement. Other algorithms, like the Sequential Importance Resampling (SIS) algorithm [12] have also been tested on our data, with very similar results to those from the Bayesian Bootstrap. We therefore concentrate on the Bayesian Bootstrap algorithm in this paper.

F. Improved State-Space Filter Model

The model presented in Section II-C is very simple in several aspects. Firstly, the errors from time step to time step are assumed uncorrelated, which is unrealistic. Secondly, the drift in the INS are assumed independent in the north and east directions. In reality, when the INS is run with velocity aiding, e.g. from a Doppler velocity log, the drift is typically larger along the vehicle track than across the vehicle track [14]. It is therefore more natural to develop the model in the vehicle's body frame, the $\{b\}$-frame, and then later transform the result to the $\{m\}$-frame or $\{e\}$-frame. For convenience, we will in fact develop our model in the $\{b'\}$-frame, which is a roll and pitch compensated body frame. This frame will have the same heading angle ψ as the $\{b\}$-frame, whereas the roll and pitch angles are zero, compared to the $\{n\}$-frame. The reason for using this frame is that the MBE measurements are automatically given in this frame from the MBE system.

In the conventional filter model a white noise model was used for the drift. A natural extension to this model is to use a first-order Markov model, see [10], for the drift in

the $\{b'\}$-frame. The position offset will be described in the $\{m\}$-frame, which is a local map frame. Using a flat earth approximation, the orientation of the $\{m\}$-frame coincides with that of the $\{n\}$-frame, the only difference being that the origin of the $\{n\}$-frame is moving with the vehicle. The filter model now becomes

$$\begin{bmatrix} \delta \boldsymbol{x}_{k+1}^m \\ \boldsymbol{v}_{k+1}^{b'} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{2\times 2} & \boldsymbol{R}_{b'}^m \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{G}_v \end{bmatrix} \begin{bmatrix} \delta \boldsymbol{x}_k^m \\ \boldsymbol{v}_k^{b'} \end{bmatrix} + \begin{bmatrix} \boldsymbol{R}_{b'}^m & \boldsymbol{0}_{2\times 2} \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_{2\times 2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_k^{b'} \\ \boldsymbol{\zeta}_k^{b'} \end{bmatrix}, \quad (22)$$

where $\boldsymbol{\zeta}_k^{b'}$ and $\boldsymbol{\gamma}_k^{b'}$ is a white noise sequence with

$$E[\boldsymbol{\zeta}_k^{b'}(\boldsymbol{\zeta}_k^{b'})^T] = \boldsymbol{Q}_k^{b'}\delta_{kl}, \quad (23)$$

$$E[\boldsymbol{\gamma}_k^{b'}(\boldsymbol{\gamma}_k^{b'})^T] = \boldsymbol{\Gamma}_k^{b'}\delta_{kl}. \quad (24)$$

The superscripts on the vectors indicate which coordinate system they are represented in. The matrix $\boldsymbol{G}_v$ is the transition matrix for the discrete 1. order Markov process $\boldsymbol{v}_k^{b'}$. This process is the discretization of the continuous 1. order Markov process

$$\dot{\boldsymbol{v}} = \begin{bmatrix} -\frac{1}{\tau_1} & 0 \\ 0 & -\frac{1}{\tau_2} \end{bmatrix} \boldsymbol{v} + \boldsymbol{\zeta}, \quad (25)$$

where τ_1 and τ_2 are positive time constants. The matrix $\boldsymbol{G}_v$ is therefore given by

$$\boldsymbol{G}_v = \begin{bmatrix} \exp(-\frac{T}{\tau_1}) & 0 \\ 0 & \exp(-\frac{T}{\tau_2}) \end{bmatrix}, \quad (26)$$

where T is the time step in the discrete-time Markov process.

The transformation matrix $\boldsymbol{R}_{b'}^m$ is simply a heading compensation:

$$\boldsymbol{R}_{b'}^m = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}, \quad (27)$$

where ψ is the heading estimate taken from the real-time navigation system.

This concludes the development of our new state-space model. Letting $\boldsymbol{\xi}^* = \begin{bmatrix} (\delta \boldsymbol{x}_k^m)^T & (\boldsymbol{v}_k^{b'})^T \end{bmatrix}^T$, where the asterisk again indicates that this is a filter model, our model can be written on the form

$$\boldsymbol{\xi}_{k+1}^* = \boldsymbol{F}^*\boldsymbol{\xi}_k^* + \boldsymbol{\nu}_k^*, \quad (28)$$

$$\boldsymbol{z}_k^* = \boldsymbol{h}^*(\boldsymbol{\xi}_k^*, \tilde{\boldsymbol{x}}_k) + \boldsymbol{w}_k^*, \quad (29)$$

which is standard state-space form with white noise

$$\boldsymbol{\nu}_k^* = \begin{bmatrix} \boldsymbol{R}_{b'}^m & \boldsymbol{0}_{2\times 2} \\ \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_{2\times 2} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_k^{b'} \\ \boldsymbol{\zeta}_k^{b'} \end{bmatrix}. \quad (30)$$

We will again use a particle filter for this model, this time with 4 states in the state vector.

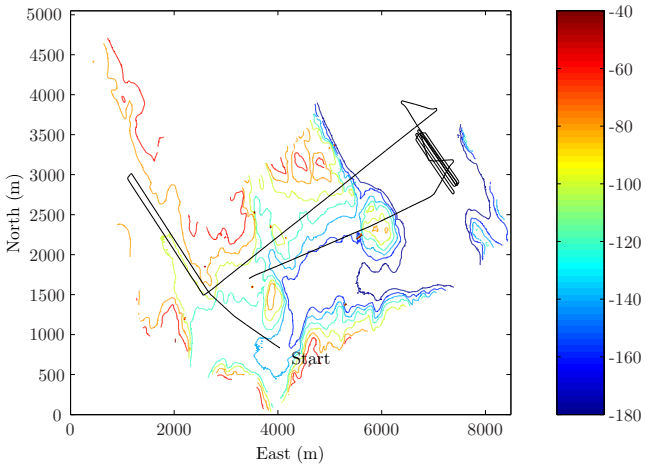

Fig. 1. Contour map and AUV trajectory.

III. Computational Results

A. Experimental Setup

In this section we present results from the aforementioned methods, using real data from a HUGIN AUV. See [3] for a description of the HUGIN class AUVs. The data were collected by a HUGIN 1 vehicle in November 2001, in an area in the Oslofjord. The vehicle was equipped with a Kongsberg Maritime EM3000 MBE (300 kHz), which together with a pressure sensor yields the total sea depth along a line across the track of the vehicle. The MBE has 127 beams, but here only up to 92 beams were used. As in [2], a sub-sampling procedure was used on the MBE beams in order to counter for unmodeled correlations within and between MBE pings, stemming from the fact that some of the MBE beams will hit the sea floor within the same map grid cell. We used our algorithms to estimate the position offset from the real-time navigation solution $\tilde{\mathbf{x}}_k$. To investigate the robustness of the estimators to errors in the initial position, the methods were initialized with errors in the magnitude of 50–100 meters. For ground truth we used a smoothed trajectory obtained using NavLab post-processing of the real-time DVL, IMU and DGPS/USBL data [15]. The accuracy of this solution is approximately 1 meter (1σ) [16]. A gridded terrain map, with 10 meter grid resolution, was used. This map was constructed using measurements from a Kongsberg Maritime EM1002 MBE (100 kHz) from a surface vessel, and the map data are therefore statistically independent from the EM3000 measurements. Fig. 1 shows a contour map of the area together with the AUV trajectory. The AUV travels from an area with a relatively rough terrain to a flat area, where it moves in a lawn mower pattern, before returning to the rough area again. As all terrain navigation algorithms require some terrain variation to work, we expect poor terrain navigation performance in the flat area, which was seen in [2], in which the same data were used.

B. Results

In order to test the behavior of the algorithms on real AUV data, the two different state-space models were implemented in Matlab using the Bayesian Bootstrap particle

TABLE II

FILTER DRIFT NOISE PARAMETERS USED IN MONTE CARLO RUNS.

Method ID	τ_1[s]	τ_2[s]	White noise parameters $[\sqrt{q_{11}}\sqrt{q_{22}}]$ [m/s]
PF2D	–	–	[0.5 0.5] (north, east)
PF4	10	10	[0.001 0.0005] (surge, sway)
PF4	100	100	[0.001 0.0005] (surge, sway)

filter. Each method was extensively tested, in order to find the optimal parameter settings. The algorithms were started with an offset from the true solution of 50 meters in each horizontal direction, and both variants were able to relatively quickly obtain good estimates in terrain suited for terrain navigation, when compared to the ground truth described above. Sequences of 1000 seconds of navigation data were used each time the algorithms were started, in order to simulate a real scenario in which the terrain navigation system is started whenever a position fix is needed. As soon as a position estimate is obtained, this should be fed back to the navigation system. Since we are dealing with sequential Monte Carlo methods, the estimates will be slightly different each time the algorithms are run. It is therefore necessary to conduct Monte Carlo runs, where the algorithms are run a number of consecutive times using the same navigation data, to assess the quality of the two variants. Table II shows the drift noise parameters used in the filters in the various runs. Notice the low white noise parameters used in the improved model.

Fig. 2 shows the average horizontal errors from 25 Monte Carlo runs 1000–2000 seconds after the start of the run, when the vehicle is in an area well suited for terrain navigation. The red line, labeled 'PF2D', shows the results using the 2-dimensional model, whereas the blue and black lines, labeled 'PF4', are the results obtained from the improved state-space model, with drift parameters $\tau_1 = \tau_2 = 10\,\mathrm{s}$ and $\tau_1 = \tau_2 = 100\,\mathrm{s}$, respectively. These parameters turned out to yield the best accuracy in the estimates. As can be seen in Fig. 2, the improved filter models seem to yield a more stable solution than the conventional one, which has a tendency of fluctuations. This is particularly evident in the interval between 200 and 400 seconds, where the estimate error from the conventional model suddenly increases for a while, before returning to around 5 meters after around 500 seconds. This behavior is also seen to some extent in the results from the improved model, but especially in the $\tau = 10\,\mathrm{s}$ case the fluctuations are of much smaller magnitude, yielding smoother estimates. The reason for this fluctuating behavior is mainly that one has to use a much higher white noise parameter for the drift in the particle filter for the conventional model than what is the case in the improved model, which has some time correlation in the process noise. This effect was evident throughout all our tests in the suited areas; the improved filter model estimates are smoother and have a lower tendency of fluctuations and hence lower probability of filter divergence.

Similar results can be seen in Fig. 3, where corresponding results 3000–4000 seconds from the start of the run are shown. Here the tendency of fluctuations in the conventional results are even more evident. Though all of the variants yield estimates of high accuracy, within 10

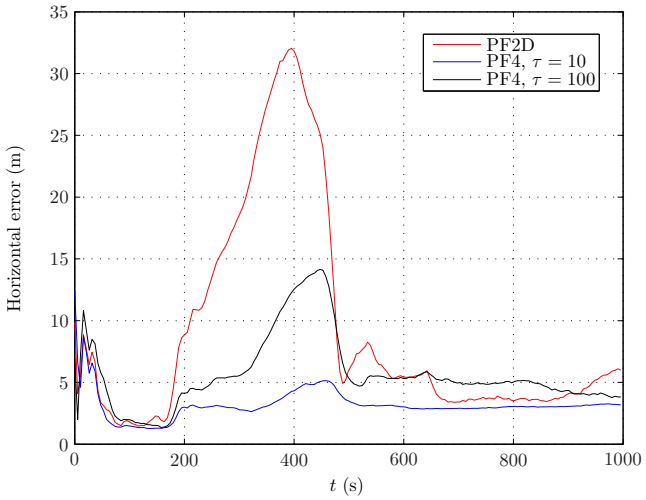

Fig. 2. Computational results from 25 MC runs, 1000–2000 seconds from start of run.

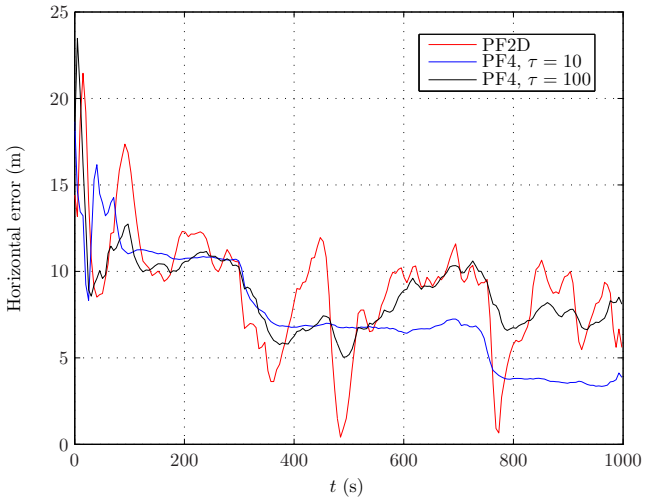

Fig. 3. Computational results from 25 MC runs, 3000-4000 seconds from start of run.

meters, which is the horizontal map resolution, the results from the improved models are superior when it comes to smoothness of the estimates. The estimates from the improved model with drift parameter $\tau = 10\,\mathrm{s}$ are the most accurate and stable.

As seen before in [1] and [2] the covariance estimates in Bayesian terrain navigation have proven to have a tendency of overconfidence, i.e. the estimated covariances are too low compared to the true uncertainties in the estimates. This was one of the rationales for deriving the improved state-space model. It turns out, however, that the covariance problem is not solved by the improved model. In fact, the estimated covariance is much lower when the improved filter model is used. Fig. 4 and 5 show the estimated north and east offsets and their standard deviations for the conventional and improved models, respectively. These results are from one single run with the particle filters. For a consistent estimate, the estimate should lie within the 1 σ bounds around 68% of the time, if we assume a Gaussian distribution. This is achieved for the conventional model, but for the improved model it is far from true. This means

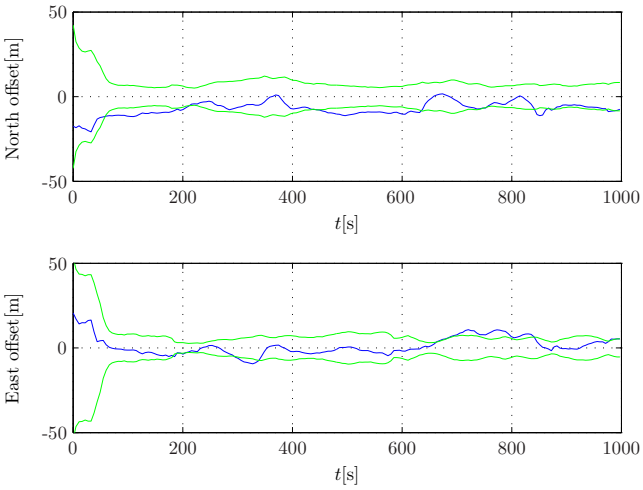

Fig. 4. Estimated north and east offsets (blue lines) and standard deviations (green lines), conventional filter model (single run).

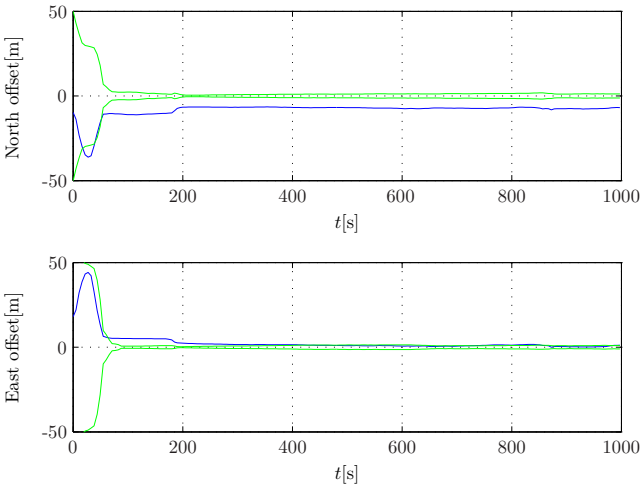

Fig. 5. Estimated north and east offsets (blue lines) and standard deviations (green lines), improved filter model with $\tau_1 = \tau_2 = 10\,\mathrm{s}$ (single run).

that there is still a discrepancy between the true system and our model. The improved smoothness of the estimates when using the improved model therefore comes at the cost of poorer covariance estimates. A higher covariance in the improved model can be attained by using a higher white noise parameter, but this severely degrades the stability and smoothness of the estimates.

The behavior of the two models in flat terrain is shown in Fig. 6. Here the vehicle is traversing the flat terrain in a lawn-mower pattern, as can be seen in Fig. 1. As expected, neither the conventional nor the improved model does well here, as the terrain does not have enough variation for the terrain navigation algorithms to obtain distinct fits. None of the results are accurate in this area, with typical horizontal uncertainties of 50–100 meters. It should be noted, however, that to some extent the covariance matrices estimated by the particle filter are able to reflect the uncertainties in the estimates both for the conventional and the improved model. However, there is a tendency of overconfidence in both cases. From these results we conclude that nothing is

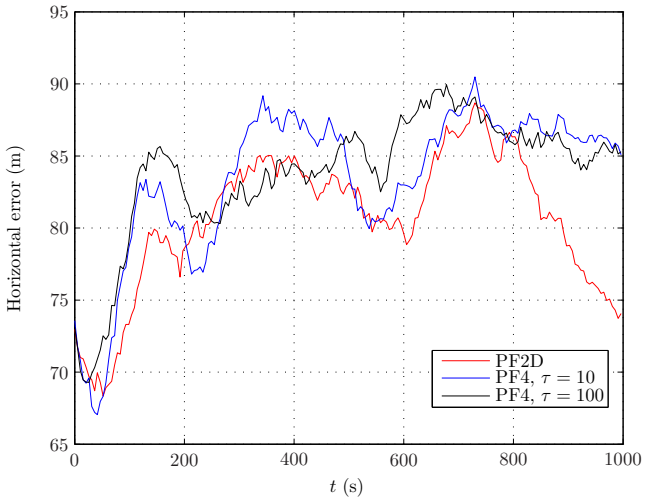

Fig. 6. Computational results from 25 MC runs, 9000-10000 seconds from start of run.

gained in the flat area by using the improved model. On the other hand, nothing is lost either. The terrain simply does not contain enough information within the current sensor and map accuracy for terrain navigation to work satisfactorily.

IV. Conclusions and Future Work

In this paper we have developed a new model for Bayesian terrain navigation of AUVs by incorporating more information about the inertial navigation system than what is done in conventional terrain navigation. The new model was developed in the vehicle body frame, using a first order Markov model for the drift error, before transforming the variables into a locally flat earth frame.

Tests done with a particle filter on real AUV data show that the new model yields smoother and more accurate estimates in suited terrain, but the estimated covariances are far to small. In other words, the estimates from the proposed model are highly overconfident. The improved smoothness and accuracy therefore come at the cost of poorer covariance estimates. The difference from the conventional model is insignificant in flat terrain.

The inconsistency is a major problem with the new model and needs to be solved in future work. The main reason for the inconsistency is that the relatively simple filter model does not model the true system accurately enough. One therefore has to include more states in the state vector in an effort to obtain a more realistic drift model in the filter. As high dimension particle filters require a very large number of particles, one may have to resort to Rao-Blackwellization [17], [18], a combination of Kalman and particle filtering.

Acknowledgments

This work is part of the UNaMap project, funded by the Norwegian Research Council, Kongsberg Maritime AS, and Kongsberg Defence & Aerospace AS. The work is carried out in close cooperation with the Norwegian Defence Research Establishment (FFI).

REFERENCES

[1] K. B. Ånonsen, O. Hallingstad, O. Hagen, and M. Mandt, "Terrain aided AUV navigation - a comparison of the point mass filter and terrain contour matching algorithms," in *Proceedings from UDT Europe 2005*, Amsterdam, the Netherlands, June 2005.

[2] K. Ånonsen and O. Hallingstad, "Terrain aided underwater navigation using point mass and particle filters," in *Proceedings of the IEEE/ION Position Location and Navigation Symposium 2006*, San Diego, CA, USA, April 2006.

[3] B. Jalving, K. Gade, O. Hagen, and K. Vestgård, "A toolbox of aiding techniques for the HUGIN AUV integrated inertial navigation system," in *Proceedings from IEEE Oceans 2003*, San Diego, CA, 2003.

[4] L. Hostetler, "Optimal terrain-aided navigation systems," in *AIAA Guidance an Control Conference*, Palo Alto, CA, 1978.

[5] N. Bergman, "Recursive bayesian estimation - navigation and tracking applications," Ph.D. dissertation, Department of Electrical Engineering, Linköping University, Sweden, 1999.

[6] I. Nygren and M. Jansson, "Terrain navigation using the correlator method," in *Position Location and Navigation Symposium, PLANS 2004*, Monterey, CA, April 2004.

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

[8] D. Crisan and A. Doucet, "A survey of convergence results on particle filtering methods forpractitioners," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 50, no. 3, pp. 736–746, Mar. 2002.

[9] B. Jalving, "Depth accuracy in seabed mapping with underwater vehicles," in *Proceedings from Oceans '99*, Seattle, WA, 1999.

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

[11] M. 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, no. 2, pp. 174–188, 2002.

[12] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer Verlag, 2001, ch. An Introduction to Sequential Monte Carlo Methods.

[13] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, 1993, pp. 107–113.

[14] B. Jalving, K. Gade, K. Svartveit, A. Willumsen, and R. Sorhagen, "DVL velocity aiding in the HUGIN 1000 integrated inertial navigation system," *Modeling, Identification and Control*, vol. 25, no. 4, pp. 223–235, October 2004.

[15] K. Gade, "Navlab, a generic simulation and post-processing tool for navigation," *European Journal of Navigation*, vol. 2, no. 4, pp. 51–59, November 2004. [Online]. Available: www.navlab.net

[16] K. Vestgård, R. Hansen, B. Jalving, and O. Pedersen, "The HUGIN 3000 survey AUV," in *Proceedings from ISOPE-2001: Eleventh International Offshore and Polar Engineering Conference*, Stavanger, Norway, June 2001, pp. 679–684.

[17] R. Chen and J. Liu, "Mixture kalman filters," *Journal of the Royal Statistical Society*, vol. 62, no. 3, pp. 493–508, 2000.

[18] C. Andrieu, A. Docet, and E. Punskaya, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001, ch. Sequential Monte Carlo Methods for Optimal Filtering, pp. 79–95.