# TerrLab – a generic simulation and post-processing tool for terrain referenced navigation

Ove Kent Hagen
Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller, Norway
ove-kent.hagen@ffi.no

*Abstract-* **One of the challenges in underwater navigation for autonomous vehicles is to get position updates for the navigation system. For deep water applications or operations requiring a high degree of covertness, surfacing for GPS is not a solution. The conventional submerged position update techniques, like DGPS-USBL or LBL, requires external physical infrastructure in the operation area of the vehicle. Terrain referenced navigation is a promising technique for submerged position updates. It requires only a bathymetric map of the operation area and the use of a bathymetric sensor during the operation. Different algorithms for terrain referenced navigation have already successfully been tested for underwater applications. These algorithms include variants of the original TERCOM, different Kalman filter based techniques and non-linear Bayesian estimators like the point mass filter and particle filters. To asses the performance of the different algorithms and their robustness on both real and simulated data, FFI has developed a tool called TerrLab. The tool was originally used to qualify algorithms for the real-time terrain navigation system for the HUGIN AUVs. It is now also used to test different sensor and DTM error models for the algorithms.**

## I. INTRODUCTION

Since the early 90s the HUGIN autonomous underwater vehicles (AUV) have been developed jointly by the Norwegian Defence Research Establishment (FFI) and Kongsberg Maritime using a dual use philosophy for serving both the civilian and military market. The HUGIN 3000 vehicles are currently operated by the commercial survey industry all around the world. The Royal Norwegian Navy operates a military version of HUGIN called HUGIN 1000. In military operations the AUVs is tasked for mine countermeasure operations in addition to typical survey operations. There is also a higher need for covertness in military operations. This sets high demands of the navigation system of the AUVs.

The HUGIN vehicles all use an aided inertial navigation system (INS). This navigation system [1] can utilize a variety of aiding sensors, selected from a toolbox of aiding techniques, fitting the operational requirements. In many operations submerged position updates may be required. To meet these requirements FFI has both developed and evaluated several aiding techniques, among these terrain navigation, underwater transponder positioning (requires one or more transponders on the seafloor), and combining a mother ship's DGPS with an acoustic ultra short base line (DGPS-USBL). Terrain navigation offers a unique ability among these techniques in that it does not require any external physical infrastructure (transponders, mother ship etc). Basically it uses a pre-obtained digital terrain model (DTM) and by correlating or otherwise comparing bathymetric measurements against this DTM, it estimates the best matching position. When integrating terrain navigation with the INS a categorization can be made of the algorithms

1. *Loosely integrated:* The correlation between the DTM and the measurements is converted into a position measurement used by the INS.
2. *Tightly integrated:* The bathymetric measurement is used directly by the internal Kalman filter of the INS.

The first approach fits the typical global search algorithms: Terrain Contour Matching (TERCOM) [2], point mass filter (PMF) and small state particle filters (PF) [3]. The second approach fits the local gradient depending algorithms like TRIN [4] and SITAN [5]. In the real-time terrain navigation system developed for the HUGIN vehicles [6], only the first approach is currently supported.

To aid the evaluation and development of different terrain navigation algorithms, a generic simulation and post-processing tool, called TerrLab, has been developed at FFI. Today TerrLab is mainly a scientific tool, and is used to support the ongoing Underwater Navigation and Mapping (UnaMap) project at the University Graduate Centre (UniK) at Kjeller in Norway, from where several algorithms have been implemented and tested on real data from HUGIN within the TerrLab framework.

### A. Coordinate reference systems

Some of the different coordinate reference systems encountered in terrain navigation are described briefly here. The vehicle reference system is called $B$. The x-axis points forward, the y-axis points to starboard and the z-axis points down the vertical when the vehicle has zero roll and pitch. In this paper it is assumed to coincide with the navigation system's reference system. The different sensor $S$ reference systems are denoted by $B_s$ and their origin is at the sensor's location.

There is another semi-body-fixed reference system in use which has been locally aligned with the horizontal plane (compensated for roll and pitch) such that the z-axis always points down (approximated by the inward normal of the earth spheroid model used). This reference system is denoted by $M_B$. Finally there is a local reference system $M_N$ with x-axis north, y-axis east and the z-axis along the inward normal of the earth spheroid model. The origin is at the vehicle's reference point.

## II. System Overview

### A. Modular design

To enable fast prototyping and easy sensor data and algorithm evaluation, TerrLab is implemented in MatLab™. The original use of TerrLab was to qualify different algorithms and their implementations for use on the real-time system. The design requirements of TerrLab therefore pretty much follow that of the real-time system. Besides the availability of a bathymetric sensor and a DTM, it is assumed that there exists an external system providing best known position and attitude, along with their expected accuracy. The main design requirements for TerrLab are as follows:

- The DTM representation is hidden from the algorithms
- The algorithms use a shared interface on input and output
- It is easy to integrate bathymetric sensors
- It is easy to integrate external navigation systems

The handling of the DTM in TerrLab is organized as a plug-in. The plug-in has a free choice of DTM representation, and is responsible for all file I/O operations, and also generation of the DTM from scattered XYZ soundings (optional). This free structure is called *Map* within the TerrLab system. An important requirement is that the Map plug-in is able to provide random access to the depth value, its accuracy and gradients at any location. The location input is given in latitude and longitude coordinates. There is a default Map plug-in already implemented that uses a regular grid combined with bilinear interpolation. This design makes it easy to evaluate different DTM representations, but on the other hand it does not support algorithms that are tightly integrated with a particular DTM representation.

A unified interface towards the external navigation system is established by converting navigation system specific data to an internal data structure *Nav* . This data structure contains the best estimate of the platforms position (latitude, longitude, depth in the same datum as *Map*) and attitude (roll, pitch, heading), along with the estimate's standard deviation, in addition to the horizontal position covariance. If position and attitude data comes from separate sources, these must be synchronized during data importation.

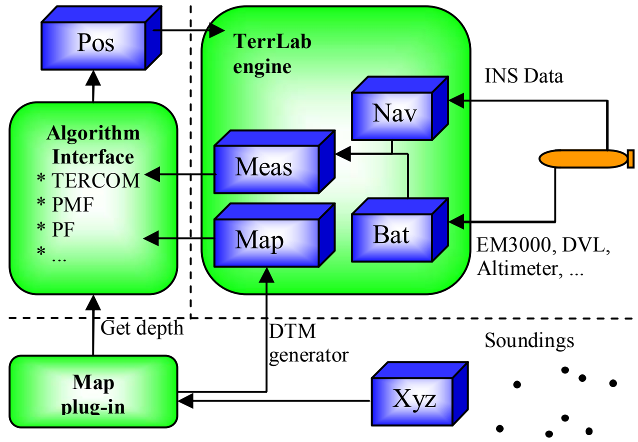


Figure 1. An overview of the TerrLab system.

For underwater vehicles the depth is usually calculated from pressure measurements. The depth component is very important both for terrain navigation and bathymetric mapping surveys in general. A dedicated study on the problem of calculating high accuracy estimates of depth from pressure measurements, completed with the implementation of a plug-in for pressure to depth conversion, following the UNESCO standard [8]. This plug-in is optional and not needed if the external navigation system already performs high quality pressure to depth conversions.

The modularity in design of the integration of bathymetric sensors is also kept by a unified data structure called *Bat*. The *Bat* structure handles arbitrary many beams from a bathymetric sensor, and for each beam there is a coordinate triple describing the sensor measurement. The coordinate triple is interpreted according to in which reference system the sensor measurements were made; see section III for more details.

The main purpose of TerrLab is evaluation of algorithms and their different implementation variants, and for testing different bathymetric sensors and error models for these sensors. To aid this, an engine is provided that performs all the necessary processing of sensor data needed to fit the common algorithms interface.

TerrLab also contains tools to aid the evaluation of bathymetric sensors by comparison towards the DTM.

### B. Algorithm interface and restrictions

TerrLab supports only the loosely coupled approach to aiding an INS. In TerrLab the *Nav* data are therefore fused with the *Bat* data into a *Meas* data structure that contains the 3D location of the vehicle reference point and the bathymetric measurement (3D vector per beam) in the $M_N$ coordinate system. The algorithms are then fed with the *Meas* data and the *Map* structure. To compare these measurements with the DTM the *Bat* part of *Meas* are converted to angles by dividing with the spheroid curvatures in the meridian (northern direction) and the prime vertical (eastern direction). The algorithms are required to estimate the vehicle's *Pos* data (3D position and covariance) after each call. There are no restrictions on the internal data structure and the internal implementation of the algorithms. This makes it very easy to both implement new algorithms and to test out different implementation strategies for already existing algorithms.

This design restricts however the algorithms from estimating other states like e.g. heading, and does not support tightly coupled algorithms like SITAN and TRIN. FFI has already developed a well established commercially available simulation and post-processing tool for inertial navigation systems, called NavLab [7]. The solution to the tightly integrated approach is planned as a module complying with the interfaces of the next version of NavLab. This version will have the flexible sensor support required by the tightly integrated approach to terrain navigation.

## III. Generic sensor and map processing

### A. Bathymetric measurement geometry

In TerrLab the bathymetric sensor measurements are classified according to their measurement configuration geometry. Two configurations have been needed so far:

- The basic measurement configuration geometry is *range* along a *unit beam vector (described by two angles)* relative to the vehicle body fixed system $B$ . Examples: altimeters, single beam echo sounders, Doppler velocity logs (typically constant beam vectors), but also the close to raw format of multi beam echo sounders (MBE) (the beam vector varies because beam forming depends on sound velocity at the transducer location).
- In the second and more elaborated measurement configuration, the measurement is delivered as a vector in body fixed, but this time roll and pitch compensated, coordinate system $M_B$. The measurement is *along*, *across*, *relative depth*. Measurements from MBEs are often given in this format.

The measurements are then simply transformed into the $M_N$ system used by the *Meas* data structure. As an example; assume a single beam echo sounder (SBE) is installed at position $\mathbf{p}^{B}_{BB_{sbe}}$ on the vehicle, with a misalignment of the sensor's reference system with respect to the vehicle's reference system, typically found from calibration, and described by the rotation matrix $\mathbf{R}_{BB_{sbe}}$. Let the vehicle's current direction cosine matrix be given by $\mathbf{R}_{M_N B}$. For the case of a SBE the beam vector is fixed in the sensor's reference system. Let $N_{bottom}$ be the center of the SBE footprint on the seafloor. The unit beam vector is denoted by $\mathbf{b}^{B_{sbe}}_{B_{sbe}N_{bottom}}$ and the range measurement is denoted by $\hat{r}$ . The transformed measurement in $M_N$ is then given by

$$\tilde{\mathbf{p}}^{M_N}_{BN_{bottom}} = \mathbf{R}_{M_N B}\left(\mathbf{p}^{B}_{BB_{sbe}} + \mathbf{R}_{BB_{sbe}}\hat{r}\mathbf{b}^{B_{sbe}}_{B_{sbe}N_{bottom}}\right) \qquad (1)$$

Since mostly acoustic sensors are used for measuring bathymetry, some comments must be made on the simplified approach above. First assume that the sound velocity $c_0$ is constant. An acoustic sensor transmits sound waves, and receives the bottom reflected waves within a cone. The direction of the center axis of this cone is then defined as the beam vector above. The opening angles of this cone define the sensor's beam width $\Delta\theta_{Beam}$ (assumed circular here). The acoustic sensor does however not measure the range itself but detects the reception time of the bottom echo, and thereby the two-way travel time $\tilde{\tau}$ of the signal. The range measurement is then simply found as $\tilde{r} = c_0\tilde{\tau}/2$.

If however the sound velocity profile is varying through the water column, the effective range and beam vector to the actual foot print center on the seafloor must be calculated from ray bending calculations to fit the first model. MBEs usually performs this calculation, but it is then natural to receive the measurement in the $M_B$ system instead, since the vehicle's roll and pitch are needed to initialize the ray bending calculations.

### B. Measurement error models

In terrain navigation there is a problem related to the modeling of the measurements. To illustrate this problem, terrain navigation in two dimensions with a single sharp beam pointing nadir is considered.

Let $(\hat{x},\hat{z})$ denote the vehicle's estimated position, and let $\mathrm{h}(x)$ denote the true terrain profile. The ideal measurement situation, assuming additive noise $w_d$ , is then described by the equation (see Figure 2)

$$\tilde{z}_d = \mathrm{h}(\hat{x}) - \hat{z} + w_d. \qquad (2)$$

The problem is however that true terrain is unknown. The terrain profile function is only known as a digital model, described by a digital representation and a method for interpolation. Let us assume that the DTM is a regular grid of nodes $\left\{(x_i,h_i) \mid x_{i+1}-x_i=\Delta x\right\}$ and that linear interpolation is used to find depth values, so

$$\hat{\mathrm{h}}_I(x) = \sum_i (1-s_i(x))\hat{h}_i + s_i(x)\hat{h}_{i+1},$$
$$s_i(x) = \begin{cases} \dfrac{x-x_i}{\Delta x} & , x\in\left[x_i,x_{i+1}\right] \\ 0 & , x\notin\left[x_i,x_{i+1}\right] \end{cases} \qquad (3)$$

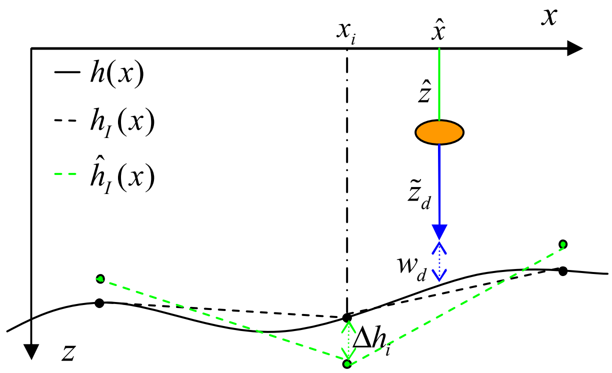


Figure 2 The bathymetric measurement $\tilde{z}_d$ is made of the true terrain $h(x)$, and not of the DTM. In addition the depths of the DTM nodes are only estimates and not the true values.

The usage of a DTM in terrain navigation will introduce additional errors $w_I$ from both the estimation error of the nodal values $\Delta h_i$ (stochastic) and the interpolation error $\Delta h_I(s_i)$ (deterministic). Let the total measurement noise be denoted by $w = w_d + w_I$. The actual measurement equation is then given by

$$\tilde{z}_d = \hat{\mathrm{h}}_I(\hat{x}) - \hat{z} + w. \qquad (4)$$

In a real measurement situation the physical beam width must also be considered, so the whole beam footprint is actually part of the measurement. The detected bottom in the return signal may not lie on the beam center axis, leading to an additional depth measurement error [9].

*Error model of the DTM*

Assuming the vehicle's depth estimate, the DTM errors and the bathymetric measurement are all independent of each other, it follows that

$$\mathrm{E}[w^2] = \mathrm{E}[\hat{z}^2] + \mathrm{E}[w_d^2] + \mathrm{E}[w_I^2] \qquad (5)$$

By further using (3), the last term of (5) becomes

$$\begin{aligned}\mathrm{E}[w_I^2] &= (1-s_i)^2\,\mathrm{E}[\Delta h_i^2] + 2s_i(1-s_i)\,\mathrm{E}[\Delta h_i \Delta h_{i+1}] \\ &\quad + s_i^{\,2}\,\mathrm{E}[\Delta h_{i+1}^2] + \Delta h_I(O(s_i^4))\end{aligned} \qquad (6)$$

Equation (6) shows how the covariance between the neighbor nodes in the DTM enters into the measurement equation of terrain navigation if linear interpolation is used. The argument is easily extended to the 3D case, and for higher order interpolation requiring more than the nearest neighbor nodes [9]. Obviously, if nearest neighbor interpolation is used, only the variance of the nearest node needs to be used.

*Sensor error model*

Either sensor specific error models or a generic default error model can be used in TerrLab. After importing sensor data, all that is required for the sensor error model in TerrLab is to produce the covariance matrix of the measurement in its own geometry configuration. As an example, the EM3000 can output measurements in the $M_B$ system. There is already developed a model of the EM3000 accuracy in this reference frame that the user may include [10]. For bathymetric sonars in general, one can also refer to [11], where a thorough treatment is given for four different methods of bottom detection.

There is also a default generic model in TerrLab, which models the error in a purely geometric fashion. Assuming the sensor reference system is perfectly aligned with $B$, the bathymetric measurement can then be written as

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{z}\end{bmatrix}^B = \mathbf{f}_S(\tilde{r},\tilde{\alpha},\tilde{\beta}) = \begin{bmatrix}\tilde{r}\sin\tilde{\beta}\cos\tilde{\alpha}\\ \tilde{r}\sin\tilde{\beta}\sin\tilde{\alpha}\\ \tilde{r}\cos\tilde{\beta}\end{bmatrix}. \qquad (7)$$

Here $\tilde{r} = r + \Delta r$ is the range measurement, as the sum of the true value and the range error. The beam pointing angles are treated the same way; beam azimuth $\tilde{\alpha} = \alpha + \Delta\alpha$ (the angle between the beam and the x-axis) and the beam zenith $\tilde{\beta} = \beta + \Delta\beta$ (the angle between the beam and the z-axis). By using the Taylor series expansion of $\mathbf{f}_S$ to first order with respect to the error about the true values, we get

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\\ \tilde{z}\end{bmatrix}^B = \begin{bmatrix}x\\ y\\ z\end{bmatrix}^B + \mathbf{A}_{r\alpha\beta}\begin{bmatrix}\Delta r\\ \Delta\alpha\\ \Delta\beta\end{bmatrix} \quad, \mathbf{A}_{r\alpha\beta} = \frac{\partial \mathbf{f}_S}{\partial[r,\alpha,\beta]} \qquad (8)$$

The last term is then recognized as the measurement error in B. Let $\mathbf{C}_S$ denote the sensor measurement error covariance matrix. The default generic model assumes a Gaussian distribution in range and angular errors. The range standard deviation is a tunable per cent of range, and the angular standard deviations are a tunable per cent of the beam width. Since the transformation in (8) is now linear in the measurement error, the covariance matrix in $B$ is then given by

$$\mathbf{C}_B = \mathbf{A}_{r\alpha\beta}\mathbf{C}_S\mathbf{A}_{r\alpha\beta}^T. \qquad (9)$$

In the same way the covariance matrix in $B$ is further transformed into $M_N$, taking into account also the attitude estimate errors as well [9]. Another approach on this part of the process is given in [12].

### *C. Measurement decimation*

There is another subtle interaction between sensor error and the DTM error in the terrain navigation measurement model. Say the vehicle is traveling level at low height 3 m above a flat sea floor. Consider a vehicle equipped with a DVL with

all the four beams forming beam angles to the vertical of 30º. The footprint of the four beams will then be the corners of a square with 3 m diagonals on the seafloor. If the DTM resolution is 10 m, all the DVL beams will frequently hit the same DTM grid cell. A similar situation arises if the vehicle frequently pings a SBE, say 1 Hz at low speed 2 m/s. If the vehicle starts at a DTM grid cell boundary, it will take the SBE beam at least 5 s to leave the DTM grid cell.
This effect will introduce an additional covariance between the measurements both within each ping and in between consecutive pings. If not compensated for, it may typically make the estimators overconfident in their own solution, a problem reported with the PMF in [13]. Later in [14] it was found that decimation of the measurements both spatially and in time helps against the estimator overconfidence of the PMF. TerrLab supports measurement decimation today, but it is planned to rather exploit this effective measurement redundancy in a combined wild point and decimation filter for the bathymetric measurements.

### D. *Batch or recursive processing*

The algorithms implemented in TerrLab can process the measurements in batch mode (collection of a configurable number of pings) or recursive mode (processing each ping by itself). This is done outside the algorithm functions, so they do not know if the measurements are actually a batch of terrain profiles from different pings or a single profile from current ping. In order to do this the measurements are as usual transformed into $M_N$ for each ping, and then translated along the vehicle's displacement vector in between the pings. The *Meas* data structure will then always contain the bathymetric measurements relatively to current navigation solution.

The advantage of batch processing is that the measurement covariance between pings (caused by e.g. tidal errors and sound velocity profile errors) may be modeled without extending the state vector. An underlying assumption is however that the navigation solution does not drift far during the collection of the batch of profiles.

## IV. SIMULATIONS

The experience so far has shown realistic data to be invaluable in algorithm testing, especially in tests for their robustness. However, simulations are useful for validation of algorithm implementation and for detection of algorithm artifacts and their possible numerical problems. In order to enable simulations of a realistic navigation system, an interface towards NavLab is planned for. NavLab is able to simulate a variety of navigation sensors, so TerrLab will then add a range-to-bottom simulator and a DTM simulator to the complete simulator capability. The range simulator takes as input the vehicles true position, the true terrain, and the sensors beam vector and beam width. It basically traces along the beam vector until it hits the sea floor, and noise is added according to the sensor error model specified. If the beam width is very wide, it can trace several sectors of the beam cone, and the range is then given either a closest range (simulates bottom detection by edge detection) or a mean range (simulates bottom detection by center of gravity calculation). There are two ways to simulate in TerrLab:

1. A basic simulator: An existing DTM is considered as true terrain, and a straight line trajectory is specified. The range simulator generates bathymetric measurements, and a navigation solution is simulated with fixed heading, depth and surge bias. A DTM for the terrain navigation algorithms is then simulated by adding Gaussian white noise to the nodes.
2. A simulator that combines real and simulated data. If the vehicle received position updates during mission, navigation can be post-processed, and the result can be used as true trajectory. The navigation solution used as input in TerrLab may then be simulated by a pure 3D translation of the real-time solution.

The simulators implemented today do not take into account ray bending, the bottom echo strength influence (typically depending on incident angle and seafloor type) on the bottom detection algorithm, nor wild points.

## V. ALGORITHMS IN TERRLAB

### A. *Point Mass Filters*

The PMF is a non-linear Bayesian estimator that actually estimates the whole probability density function (PDF) of the state vector over a grid in state space. The method was first introduced in terrain navigation for aircrafts [3]. The algorithm fits directly into the algorithm interface in TerrLab, and can be configured to yield both the MMSE and the ML estimates from the PDF, along with the covariance estimate of the PDF. The filter can use as input an a priori position error PDF, and the measurement error can be modeled by an arbitrary measurement error PDF as well. Navigation system errors are also modeled by specifying a PDF for the navigation position error drift between measurement updates.

In TerrLab both a regular grid (see Figure 3) and an adaptive grid version in 2D (the state vector is horizontal position error) is implemented. The adaptive grid implementation [13] follows the method suggested in [3]. In addition a 3D regular grid implementation, with the vehicle depth error as the additional state, has been implemented [14] as part of the UnaMap project.

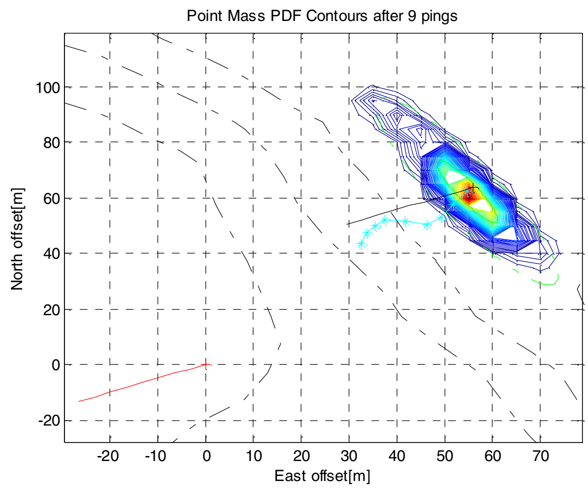


Figure 3 An example of a PDF resulting from processing 9 pings from a DVL. Red line is a priori navigation solution, black line is true position and cyan marks position estimates from the PMF.

### *B. Terrain Contour Matching filters*

TERCOM [2] was the first terrain navigation algorithm to be published and used in real systems. The algorithm is very easy to implement, and the results are actually quite good [13]. It simply sums the absolute difference (variants may use quadratic) between the bathymetric measurement and the DTM lookup value on a grid about current position, forming the correlation surface. The position solution is the grid point of the correlation surface's minimum. The main problem with the method is that it does not have an inherent accuracy description of its solution, which is desirable when integrating the resulting position measurement with an INS. Several methods have been proposed to address this problem. In [18] the concept of a matching strength surface within a validation gate about the navigation solution is introduced. In [9] it is shown that this surface is actually the position PDF if TERCOM, with all its underlying assumptions, is modeled within the PMF framework.

The assumptions are:

- A priori PDF is uniform over the search area
- Measurement errors are uncorrelated and Gaussian distributed
- There is no drift in the navigation solution between pings in the correlation sequence
- Use the ML estimate of the PDF

In [16] convergence is assumed by using large enough batches of measurements (or the beams from a single ping of a 3D sonar), and covariance is then calculated according to the theoretical accuracy of a ML estimator.

Both these variants are implemented in TerrLab.

### *C. Particle filters*

Another way to implement a non-linear Bayesian estimator is through Monte Carlo simulation methods, called particle filters (PF). An initial set of particles are drawn from the initial PDF. This set of particles now represents the PDF. The particles are then propagated and simulated according to the system dynamics model. During the measurement update the particles are assigned weights according to their measurement likelihood. In order to prevent degeneration of the PFs, the particles frequently need to be resampled from the recently updated PDF. Different strategies to solve the degeneration problem lead to different PF variants.

In TerrLab there is a 2D (state vector is horizontal position error, see Figure 4) bootstrap version implemented [17][3] and a 3D bootstrap [14] version with an additional vehicle depth error state.

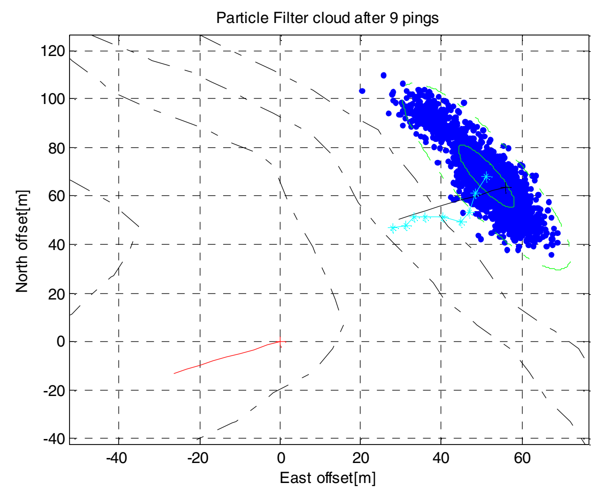


Figure 4 The PF cloud after processing 9 pings from a DVL. Red line is a priori navigation solution, black line is true position and cyan marks position estimates from the PF.

### *D. Relative profile or absolute depth*

All the algorithms in TerrLab may be configured to perform the profile difference calculations in an absolute or relative mode. The absolute profile difference mode uses the *Meas* data structure directly and forms the measurement differences in each offset position. In the relative profile difference mode a representative depth level (mean or median) is first found from the beams in the *Meas* data structure. For each offset position the expected DTM profile is then found, and a representative DTM depth level (mean or median) is then found. The measurement differences are then calculated between these two relative depth measurements. The motivation behind this is the assumption that the largest error component in depth is an almost constant depth bias (typically due to tidal elevation). While this has proven to be wise in terrain very suitable for terrain navigation, like the cells used in original TERCOM algorithm [2], it is not wise to do in the case of self-similar terrain [15]. Self-similarity may seem like a rare case, but large portions of the underwater topography, for instance near beaches or inside fjords, there is often self-similarity in one direction within a typical correlation sequence.

All the 2D algorithms will actually also deliver an estimate of the error of the vehicle reference depth with respect to the DTM. If the DTM is globally referenced, this depth error estimate may be used to globally reference INS depth. The assumption is then that the global depth error is slowly varying. In addition the terrain navigation solution must have converged to correct position and no depth bias in the bathymetric sensor processing chain must be present (SVP errors, lever arm errors, etc.). If this is the case, any bias between measured depth profile and DTM depth profile can be interpreted as global vehicle depth error.

## VI. Summary

Terrain referenced navigation is a promising technique for submerged position updates. TerrLab is a tool for the evaluation of different terrain navigation algorithms, and has been presented in this paper. It can be used on both simulated and real data from the HUGIN vehicles, and is flexible with respect to integration of new bathymetric sensors and their error models.

Today TerrLab is mainly a scientific tool and is still under development. It is used to support terrain navigation research under the UnaMap project at UniK. The UnaMap project has made major contributions to the available algorithms in TerrLab.

## References


[1] 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 IEEE Oceans 2003,* San Diego, CA, 2003.

[2] J. Golden, "Terrain contour matching (TERCOM): A cruise missile guidance aid," in *Image Processing for Missile Guidance,* T. Wiener, Ed., vol. 238, The Society of Photo-Optical Engineers, 1980, pp. 10-18.

[3] N. Bergman, L. Jung and F. Gustafsson, "Terrain navigation using Bayesian statistics," in *IEEE Control System Magazine,* vol. 19, no. 3, 1999, pp. 33-40.

[4] O.K. Hagen and P.E Hagen, "Terrain referenced integrated navigation system for underwater vehicles," in *SACLANTCEN Conference Proceedings CP-46,* E. Bovio, R. Tyce and H. Schmidt, Eds. La Spezia, Italy: NATO SACTLANT Undersea Research Centre, 2000, pp. 171-180.

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

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

[7] K. Gade, "NavLab, a generic simulation and post-processing tool for navigation," in *European Journal of Navigation*, vol. 2, no. 4, 2004.

[8] N.P. Fofonoff and R.C. Millard Jr, "Algorithms for computation of fundamental properties of seawater," in *Unesco Technical Papers in Marine Science*, vol. 44, 1983

[9] O. Hagen, "TerrLab – a generic simulation and post-processing tool for terrain navigation," Norwegian Defence Research Establishment, Tech. Rep., unpublished.

[10] E. Hammerstad, "Multibeam echosounder accuracy," *EM Technical Note,* Kongsberg Simrad, Horten, Norway, 1998.

[11] X. Lurton, "Measurement precision of bathymetric sonars as a function of signal/noise ratio," in *Traitement du signal*, vol. 18, no. 3, 2001

[12] Hare R., "Depth and position error budgets for multibeam echosounding," in *International Hydrographic Review*, LXXII(2), Monaco, 1995, pp. 35-69

[13] 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.

[14] K.B. Ånonsen and O. Hallingstad, "Terrain aided underwater navigation using point mass and particle filters," in *Proccedings from PLANS 2006*, San Diego, CA, April 2006.

[15] M. Mandt, "Terrengreferert posisjonering av undervannsfarkoster," Norwegian Defence Research Establishment, Tech. Rep. FFI/RAPPORT-2001/05900, 2001.

[16] I. Nygren and M. Jansson, "Terrain navigation for underwater vehicles using the correlation method, in *IEEE Journal of Oceanic Engineering,* vol. 29, no. 3, 2004.

[17] 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.

[18] O. Bergem, "Bathymetric navigation of autonomous underwater vehicles using a multibeam sonar and a Kalman filter with relative measurement covariance matrices," Dr. Scient thesis, University of Trondheim, Norway, 1993.