Comparison of fix computation and filtering for autonomous acoustic navigation

Jérôme Vaganay†‡, James G. Bellingham† and John J. Leonard†§

Oceanographic data acquired by an autonomous underwater vehicle (AUV) must be correlated with accurate position information in order to be of value to scientists. Accurate navigation is therefore an essential requirement which can be fulfilled by use of acoustic long-baseline navigation systems and dead-reckoning sensors. Two approaches are presented in this paper. The fix computation approach consists of dead reckoning between fixes computed from a set of acoustic travel times. The filtering approach consists of correction of the vehicle dead-reckoned motion by taking into account the influence of the measured travel times. Fix computation is widely used for the positioning of manned submersibles and there seems to be some reluctance to switch to a filter-based approach, mainly because of the fear for divergence. After a detailed description of these two approaches, their advantages and drawbacks are compared by applying both algorithms to real data sets collected by the Odyssey II AUV developed at Massachusetts Institute of Technology Sea Grant. The filter presented in this paper is not subject to divergence. In the worst case, it would reject all the acoustic travel times measurements and proceed by dead reckoning only, so that re-initialization would be needed. This situation would also happen in the fix computation approach if the algorithm locked on an erroneous fix.

1. Introduction

One of the applications of survey-type autonomous underwater vehicles (AUVs) is the acquisition of oceanographic data for scientists. AUVs have indeed many advantages compared with existing systems. For instance, they are cheaper, easier to use, and faster and safer than towed systems. The critical problem to solve is that of navigation, since data have no significant value unless they can be associated with good estimates of the positions where they were acquired. Acoustic long-baseline (LBL) systems have the potential for accuracy. Owing to the low position update rate, however, these systems have to be associated with dead-reckoning sensors. LBL data are also affected by outliers due to noise and multiple propagation paths between emitter and receiver (Deffenbaugh 1994), so that outlier rejection procedures are required.

We consider two different approaches to vehicle positioning: fix computation and filtering. Fix computation consists of computing the least-squares solution corresponding to the intersection of spheres centered at the beacon locations with radii equal to the measured distances. The resulting fix is used to reset the dead-reckoned position. Outlier rejection is performed at the fix level. It consists of propagating the position uncertainty due to dead reckoning and defining validation regions for the validation of a new fix. When filtering, the vehicle position is predicted using dead-reckoning measurements and corrected using acoustic measurements by means of a Kalman filter, so that the estimated position is a weighted combination of acoustic and dead-reckoning data. Outlier rejection is performed by transforming the position uncertainty due to dead reckoning in the travel time domain.

Odyssey II is a low-cost deep-ocean-capable AUV developed at Massachusetts Institute of Technology (MIT) Sea Grant for scientific exploration of the

Accepted 16 October 1997.

†Massachusetts Institute of Technology, Sea Grant College Program, 292 Main Street, E38-300, Cambridge, MA 02139, USA.
‡Present address: Laboratoire d’Informatique, de Robtique et de Microelectronique de Montpellier, 161 rue Ada, 34392 Montpellier Cedex 5, France.
§Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.
oceans (Bellingham et al. 1994). As part of its sensor suite, the Odyssey II is equipped with a custom LBL navigation system (Atwood et al. 1995). During extensive field operations, the Odyssey II has collected a large amount of LBL data at various locations. Two of these data sets have been post-processed in order to illustrate the advantages and drawbacks of both approaches. The first experiment took place in Boca Raton, Florida, and the second in Buzzard’s Bay, Massachusetts.

The structure of the paper is as follows. Section 2 briefly describes the principle of LBL navigation. Section 3 presents the two algorithms with some comments on track initialization, outlier rejection and vehicle motion compensation. Section 4 discusses the advantages and drawbacks of each approach and presents some results obtained with the two above-cited data sets.

2. Long-baseline navigation principle

LBL navigation refers to an underwater positioning system employing an array of acoustic beacons separated by the order of hundreds of meters to a few kilometers. The array is usually calibrated through use of an additional acoustic transponder that is hung from a surface ship and interrogates the array from various locations.

The operation of our LBL system is as follows. In autonomous mode, the vehicle pings at a regular rate (every 5 s) at the ‘master’ frequency (9 kHz). Each acoustic beacon replies at its own frequency (10, 11, 12 and 13 kHz). If the average speed of sound is known, then the elapsed time between the initial ping and detection of the replies yields the range between the vehicle and the beacons. The navigation problem then consists of determining the vehicle position, given round-trip travel times measured from the vehicle to each beacon and the location of each beacon as determined in calibration.

Transponder-based navigation of AUVs may appear to be straightforward since each acoustic beacon can be uniquely identified from the frequency of its reply. Because of multipath propagation, noise and other error sources, however, the underwater LBL navigation problem can be difficult owing to the challenge of defining efficient autonomous outlier rejection procedures.

3. Presentation of the two approaches

Let us first define the following parameters. The local Earth frame origin is at the water surface. The $x$ axis points north, the $y$ axis points east, and the $z$ axis points down. The vehicle position in this frame is $(x, y, z)$, and its orientation is defined by its yaw, pitch and roll angles $(\psi, \theta, \phi)$. The coordinates of beacon $i$ are $(x_i, y_i, z_i)$. The measured round-trip travel times between the vehicle and beacon $i$ is $t_i$ and the associated distance is $d_i$. The distances are computed from the travel times by the approximate relation $d_i = c(t_i - \tau_i)/2$, where $c$ is the average speed of sound and $\tau_i$ is the turn-around time of beacon $i$ (typically 15 ms).

3.1. Fix computation approach

The fix computation approach consists of dead reckoning from the last acoustic fix and resetting the vehicle position as soon as a new fix is obtained. The dead-reckoned vehicle position $X = [x \ y]^T$ is calculated by means of the classical equation

$$X_{k+1} = X_k + \delta A_k(\psi_k, \theta_k, \phi_k) V_k \Delta t,$$

where the number and nature (water or bottom referenced) of the components of $V_k$ depend on the speed sensor type, $\delta A_k(\psi_k, \theta_k, \phi_k)$ is the rotation matrix allowing one to transform $V_k$ from the vehicle frame to the earth reference frame, and $\Delta t$ is the sampling interval.

Ideally, a new fix is obtained at every LBL cycle, although the number of returns may sometimes not be sufficient to compute a fix (0 or 1 return). Spurious travel times, however, can lead to inaccurate fixes which have to be discarded. A validation procedure is needed to determine whether a calculated fix is accurate enough to be used for a reset. Such a procedure is referred to as a ‘fix domain’ outlier rejection procedure because it runs on calculated fixes rather than on the travel times themselves. The details of the procedure are given in section 3.4.

We now present the least-squares algorithm allowing one to compute a fix from two or more travel times. It only optimizes the $x$ and $y$ position components and makes use of the vehicle depth $z$ which is directly and accurately measured by the on-board pressure sensor.

The vector $\mathbf{d} = [d_1 \cdots d_n]^T$ of the distances measured between the vehicle and the $n$ beacons ($n \geq 2$) during an LBL cycle can be related to the vehicle position by the following equation:

$$\mathbf{d} = f(\mathbf{X}) + \mathbf{n},$$

with

$$f(\mathbf{X}) = \begin{bmatrix} [(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2]^{1/2} \\ \vdots \\ [(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2]^{1/2} \end{bmatrix},$$

where $\mathbf{n}$ is a Gaussian noise vector, with covariance matrix $\mathbf{N}$, assumed to encompass error sources such as errors on the beacons’ locations, on the travel time measurements, on the speed of sound and on the vehicle depth. The residual $Q(\mathbf{X})$ to minimize under the Gaussian noise assumption is given by

$$Q(\mathbf{X}) = [\mathbf{d} - f(\mathbf{X})]^T \mathbf{N}^{-1} [\mathbf{d} - f(\mathbf{X})].$$

The function $f(\mathbf{X})$ being nonlinear, it is linearized about a reference position $\mathbf{X}_k$ by a first-order Taylor expan-
Running a good initialization procedure all along the vehicle motion can also provide valuable reference fixes and serve as a back-up for the filtering approach, or as a reference to check the good behavior of the filter.

Although filtering has several advantages compared with fix computation, it is not an easy solution to implement in an AUV. The filter behavior is highly affected by errors on the models and on the parameters. The risk of systematic rejection of travel times (because the filter used erroneous acoustic data) is always a concern, but it could be monitored to re-initialize the filter when needed. It has to be noted that systematic rejection of fixes can also happen when using the fix computation approach if the algorithm locks on an erroneous fix.

With appropriate sensors, a solution could be to use several filters, corresponding to different navigation conditions (Vaganay and Rigaud 1995), in cooperation with a fix computation approach. A bank of filters could allow estimation of model parameters. Fixes could be calculated in parallel and serve as a reference to check the good behavior of the filters every time that a fix is considered valid. Even if several consecutive LBL cycles are necessary to validate a fix, this information could still prevent the filter from malfunctioning. This highlights the need for a navigation supervisor which would manage the bank of filters, determine when a fix is valid, check the good behavior of the filters and determine whether a fix must be used to re-initialize the filters.

References


necessary to obtain a fix. If dead reckoning can be trusted long enough, this is, however, a viable approach.

5. Conclusion
Filtering appears to be a more attractive solution essentially because of the smoother tracks that it provides. Another great advantage of the filtering approach is its ability to process travel times one by one as they arrive, so that cycles involving only one or two valid travel times can be used to update the vehicle position. Furthermore, as opposed to the fix computation approach, using a pair of travel times does not result in a position jump, since the effects of baseline miscalibration are filtered out. On the other hand, the computation of fixes is still required to initialize the vehicle track.
threshold of the normalized innovation squared, which determines the validity of each travel time.

For fix computation, outlier rejection has to be added. One possibility has been presented in this paper. Fix validation can also be viewed as an extension of the initialization problem, since a procedure that works to determine an initial fix should work just as well along the complete track. Figure 13 shows the fixes that were selected by the initialization procedure (see section 3.3) iterated over the complete vehicle trajectory in Buzzard's Bay. These fixes are interesting in that they depend only very little on the dead-reckoning sensor measurements. The only drawback is that the position update rate is lower because several LBL cycles can be
respectively. When four beacons are deployed, precluding the use of two-beacon fixes is not too restrictive and provides a smoother track. When navigating in a more common three-beacon array, however, there exists a trade-off between the desired track smoothness and the number of position updates.

4.3. Outlier rejection procedures

Outlier rejection can be performed at almost no extra computational cost when using the filter. Since the propagation of the position uncertainty during dead reckoning is already part of the normal computations of the filter, the only calculation that needs to be added is the
single travel time to correct the state. The fourth and fifth columns contain the percentages of use of the total number of returns for the filter and the fix computation approach.

In the tracks presented in figures 4 and 6, two-beacon fixes were allowed in order not to penalize the fix computation approach regarding the percentage of use of acoustic data. Because of their greater sensitivity to baseline miscalibration, however, these fixes largely participate to the track jitter. In fact the less accurate fixes in figures 4 and 6 result from the use of only two travel times. Figures 11 and 12 show the tracks obtained by precluding the use of two-beacon fixes. It can be seen that these tracks are a little smoother, but the percentages of use of acoustic data are also reduced to 86.7% and 67.4% for Florida and Buzzard’s Bay experiments.
Comparison of fix computation and filtering for navigation

Figure 3. Vehicle track by filtering (Florida).

Figure 4. Vehicle track by fix computation (Florida).

times validated by the filter during each experiment are represented in figures 9 and 10. These can be compared with the raw travel times in figures 1 and 2. Results concerning the use of the measured travel times for the Florida and Buzzard's Bay experiments are represented in table 2. The second column contains the total number of LBL cycles after initialization. The third column contains the number of cycles during which the filter used a

Table 2. Results using measured travel times

<table>
<thead>
<tr>
<th></th>
<th>Total number of cycles</th>
<th>Number of cycles with single travel time</th>
<th>Use of travel time (%)</th>
<th>Use of travel time (fix computation) (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Florida</td>
<td>861</td>
<td>18</td>
<td>92.6</td>
<td>90.9</td>
</tr>
<tr>
<td>Buzzard's Bay</td>
<td>630</td>
<td>52</td>
<td>87.3</td>
<td>74.0</td>
</tr>
</tbody>
</table>
$n_\epsilon$ (13) helps the filter to adapt to sudden changes in the drifts due to turns for instance. Unfortunately, it also decreases the confidence in the position prediction, which leads to a less smooth track by validation and use of ‘not-so-good’ travel times. In the worst case, decreasing the confidence in the prediction could facilitate the validation of erroneous travel times and induce the rejection of all the following acoustic data. The vehicle would then end up dead reckoning and the filter would have to be re-initialized. In addition, the noise variances are generally set to fixed values, although they may vary in reality. With more sensor information it is possible to try to estimate the various errors source separately by use of algorithms adapted to the navigation conditions. This idea has been addressed by Vaganay and Rigaud (1995).

The smoothness of the filtered track compared with the track obtained by fix computation is clearly shown in figures 3 and 4 and in figures 5 and 6. In figure 5, the motion started in the upper left corner. The sudden change that can be observed after motion for about 300 m corresponds to the time that an initial fix was obtained and used to reset the vehicle position. As can be seen in figures 7 and 8, which show the position variation between two successive sampling times for each method, the position jitter of the filtered track is several orders of magnitude smaller than that of the fix-based track. The means and standard deviations of the position change when acoustic data are taken into account are shown in table 1.

<table>
<thead>
<tr>
<th></th>
<th>Fix computation</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Standard</td>
</tr>
<tr>
<td>Florida</td>
<td>5.4</td>
<td>6.8</td>
</tr>
<tr>
<td>Buzzard’s Bay</td>
<td>5.4</td>
<td>9.0</td>
</tr>
</tbody>
</table>

Improving the dead-reckoning accuracy by use of accurate sensors would improve the overall performance of the filter. With a greater confidence in the prediction the track would be even smoother as the noise on the travel time measurements would be better filtered. On the other hand, although it would improve the selectivity of the outlier rejection procedure for fix computation, the smoothness of the track would not be improved since it would still jump from a noisy fix to another.

4.2. Travel time processing

In the filtering approach, a single travel time can be processed at the sampling time immediately following its reception. In the fix computation approach, it is necessary to wait for all the beacons to reply, and a fix can be calculated only if at least two returns have been received. Filtering then allows one to take advantage of more acoustic data than fix computation. The travel
equation are negligible at low speeds but can be significant at higher speeds. They also depend on the array configuration as well as on the vehicle direction of motion in the array. The correction mentioned above is applied to the results presented in this paper.

When computing fixes, the least-squares equations also assume that the vehicle pinged and received the replies at the same position although this is not true. Corrections can be applied by means of the 'running fix' method (Cestone et al. 1977). This method consists of modifying the beacons' coordinates used in the computations, based on the dead-reckoned motion between the ping time and the times of reception of each return. This allows one to compute the position at the ping time. Since a new fix is generally computed after a preset time out following the ping time or at the beginning of the following LBL cycle, it is necessary to add the dead-reckoned motion between the ping time and the current time to obtain the current position. This method has been used in the fix computation algorithm applied to the data presented in this paper.

4. Comparison of the two approaches on real data

The two approaches presented above have been applied to a variety of data sets acquired by the Odyssey II. We report results for two experiments.

- Florida. The experiment took place 1.5 km offshore near Boca Raton, in water depths of 10–20 m. The raw travel times measured during the vehicle motion are represented in figure 1.

- Buzzard's Bay. The experiments took place south of Massachusetts in an average water depth of about 12 m. The raw travel times measured during the vehicle motion are represented in figure 2.

Because a speed sensor was unavailable for these missions, a constant longitudinal speed $u$ of 1.4 m s$^{-1}$ is assumed (based on a previous thrust-speed calibration of the Odyssey II).

4.1. Track smoothness

Track smoothness when filtering is a direct consequence of the way that the filter works. As opposed to fix computation, the effect of a travel time measurement on the predicted position is weighted by the measurement variance and by the confidence in the prediction, rather than taken as an exact value. The vehicle track is then less subject to jitter in response to the noise on the acoustic data. The counterpart is that the good behavior of a filter highly depends on model accuracy and on good knowledge of the various noise variances. The state equation is sensitive to all the unknown parameters which induce dead-reckoning errors (underwater currents, heading sensor misalignment, magnetic compass fluctuations due to uncompensated magnetic fields generated by the vehicle, etc.). In general, the idea is to account for all the non-modeled errors in the state noise. In our case, using large noise levels for $n_a$ and

![Figure 1. Raw round-trip travel times measured during the Florida experiment.](image-url)
can proceed. Using an inaccurate position requires setting the initial position error covariance to a large value to account for the uncertainty in this position. This can lead to the validation of erroneous fixes or travel times and may prevent the algorithm from converging to the right track by rejection of the following valid fixes or travel times. For both approaches, it is then necessary to run an initialization procedure during which the vehicle dead-reckons from a rough initial position, before being reset by the output of the procedure.

The track initialization problem is directly related to the difficulty of validating a fix in the presence of outliers in the travel times; the vehicle initial position cannot be simply obtained by use of the first set of returns since these measurements may be affected by outliers. The initialization procedure would benefit from being based on acoustic data only. In these conditions, it is difficult to be able to validate or discard a fix by considering each cycle individually. A method has been proposed which consists of checking the consistency of fixes computed from subsets of the travel times available during a cycle (Atwood et al. 1995). This method is more attractive when at least four beacons are available owing to the sensitivity of two-beacon localization to calibration errors (Cestone et al. 1977). The approach used in this paper is inspired by the way that a human operator piloting a manned subsensible would perform the task—by looking for consistency in the most recently calculated fixes as they would appear on a display. Fixes resulting from valid travel times tend to gather in the vicinity of the real vehicle track, whereas those including outliers tend to locate the vehicle at remote and inconsistent positions. During the initialization, least-squares fixes are computed from at least three returns. This allows one to avoid the ambiguity associated with the use of two-beacon fixes. The procedure assumes a rather linear motion and goes as follows.

A line is fitted to the \( M \) most recent fixes by minimizing the absolute deviation of the fixes with respect to the line. When a new fix is obtained, two thresholds are calculated: \( T_1 \) is an upper bound of the distance traveled by the vehicle since the line parameters were last computed, based on \( k \) times the vehicle speed; \( T_2 \) is the maximum authorized lateral deviation from the line for the new fix. For a fix to be considered the initial position, it is necessary that two successive fixes lie closer to the line than \( T_2 \) and be separated by less than \( T_1 \). \( T_2 \) then checks the alignment consistency perpendicular to the direction of motion, whereas \( T_1 \) checks the consistency of successive fixes along the line. In the result section, we used \( M = 5 \), \( k = 1.5 \) and \( T_2 = 15 \) m. Once the initial fix is obtained, it is used to reset the vehicle position and the fix error covariance matrix is used to initialize the associated position error covariance matrix.

3.4. Outlier rejection

The outlier rejection procedures used in each approach are very similar in that they are both based on the predicted vehicle position \( X \) (see (1) for fix computation and (13) for filtering) and the associated error covariance matrix \( P \). In both cases, \( P \) is propagated according to

$$ P_{k+1/k} = FP_{k/k}F^T + J_kCJ_k^T + Q. \quad (15) $$

In this equation, \( F \) is the Jacobian of \( X \) (unit matrix for fix computation), \( J \) is the Jacobian of \( X \) with respect to the speed and orientation measurements, \( C \) refers to the covariance matrix of these measurements, and \( Q \) accounts for modeling errors.

Under normal distribution assumption for the measurements, a validation region can be defined in the measurement space where the measurement should be found with a certain degree of probability (Bar-Shalom and Fortmann 1988). This validation region is defined in the fix domain for fix computation. It is defined in the travel time domain for filtering by transforming the uncertainty in \( X \) into an uncertainty in \( t \) by means of the Jacobian \( H \) of \( t \). The measurement validation test consists of applying a threshold to the normalized innovation squared:

$$ v_k^T S_k^{-1} v_k < \gamma. \quad (16) $$

For the fix computation approach, when a new fix \( X'_k \) is obtained, together with its error covariance matrix \( P'_k \), we have \( v_k = X'_k - X_k \) and \( S_k = P'_k + P_k \). If a fix passes the test, then \( X \) is reset by \( X' \) and \( P \) is reset by \( P' \).

For filtering, when a new travel time \( t_k \) is obtained with its variance \( \sigma_t^2 \), we have

$$ v_k = t_k - \hat{t}_k, \quad \{ \begin{array}{l}
S_k = H_k P_{k+1/k} H_k^T + \sigma_t^2,
\end{array} \} \quad (17) $$

with \( H \) representing the Jacobian of \( t \). A travel time passing the test is used to correct \( X \), and \( P_{k+1/k} \) is corrected according to the classical Kalman equations.

3.5. Compensation of the vehicle motion during a cycle

When filtering, each travel time is related to the predicted vehicle position at the time of its reception by means of the observation equation. This equation is then not exact in that the vehicle moved between the ping and the reception times. The travel time should then be expressed as a function of the vehicle positions at the ping time and at the reception time. This can be approximated in the filter at the expense of a little more computation by dead reckoning the vehicle motion along each cycle and by applying appropriate corrections (Vaganay and Rigaud 1995). The position errors resulting from using a single position in the observation
sion:
\[ f(X_{k+1}) = f(X_k) + G[X_{k+1} - X_k], \]  \hspace{1cm} (5)
where
\[ G = \frac{\partial f(X)}{\partial X} \bigg|_{X=X_k}. \]  \hspace{1cm} (6)
Substituting (5) in (4), the residual at \( X_{k+1} \) can be expressed as follows:
\[ Q(X_{k+1}) = [d_1 - GX_{k+1}]^T N^{-1} [d_1 - GX_{k+1}], \]  \hspace{1cm} (7)
with
\[ d_1 = d - f(X_k) + GX_k. \]  \hspace{1cm} (8)
The computation of \( X_{k+1} \) which minimizes \( Q \) is performed by means of the Gauss–Newton algorithm. The search direction \( s_k \) along which the criterion can be minimized starting from \( X_k \) in the case of a linear least-squares approximation is first determined:
\[ s_k = (G^T N^{-1} G)^{-1} G^T N^{-1} [d - f(X_k)]. \]  \hspace{1cm} (9)
The step along this direction is then determined by looking for the smallest integer \( k \), with \( k \geq 0 \) and such that
\[ Q(X_{k+1}) = Q(X_k + 2^{-k} s_k) < Q(X_k). \]  \hspace{1cm} (10)
If the criterion diminution is still greater than a preset threshold, the algorithm is reiterated with
\[ X_{k+1} = X_k + 2^{-k} s_k. \]  \hspace{1cm} (11)
Otherwise, the minimization stops.

The covariance matrix of the estimation error is given by
\[ C_X = (G^T N^{-1} G)^{-1}. \]  \hspace{1cm} (12)
When four beacons are deployed, as is the case in one of the two experiments described in the results section, the LCL cycles will most often give rise to a set of four returns. The least-squares algorithm can then be used with \( n = 4 \). A single outlying distance, however, may prevent the algorithm from converging (the residual does not decrease), although a solution would have been obtained by running the least-squares algorithm on the three valid returns. The least-squares algorithm is then run with the four returns first and checked for convergence. If it did not converge, the least-squares solution is run with the four possible combinations of three returns. The three-beacon fix with the least residual is kept. In case there is more than one spurious distance, the algorithm does not provide any position update.

3.2. Filtering approach

The filtering approach consists of dead reckoning from an estimate of the vehicle position and incorporating travel time information to correct for drift. Valid travel times are used one by one as they arrive and their influence on the position estimate by their noise variance and the confidence in the predicted position. We limit our presentation to the state and observation models used in the filter, as the Kalman equations can be found in many sources (Bar-Shalom and Fortmann 1988). Preliminary simulations of the filter have been reported by Vaganay and Rigaud (1995).

The state vector is made up of the \( x \) and \( y \) components of the vehicle position, and of the velocity errors \( u_d \) and \( v_d \) expressed in the vehicle frame (alongship and athwartship respectively). These drifts are assumed to capture the influence of underwater currents, the effect of side slip during turns, and uncertainties on the measured velocity. Considering a vehicle able to measure its longitudinal velocity \( u \) only, the state equation can be written
\[
X_{k+1/k} = \begin{bmatrix} x \\ y \\ u_d \\ \end{bmatrix} + 
\begin{bmatrix} x \\ y \\ u_d \\ \end{bmatrix} 
= \begin{bmatrix} x \\ y \\ u_d \\ \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ n_u \\ \end{bmatrix}
\]  \hspace{1cm} (13)

The returns from each beacon arrive asynchronously, depending on their position relative to the vehicle. The travel times are processed one by one at the sampling time immediately following their reception. The observation equation expresses a travel time as a function of the vehicle position, of the beacons’ coordinates, of the speed \( c \) of sound and of the vehicle depth \( z \) directly measured by the on-board pressure sensor:
\[ t_i = \frac{2}{c} \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}. \]  \hspace{1cm} (14)
The travel time variance \( \sigma^2_t \) is taken to be large enough to encompass uncertainties in the speed of sound, in the measured depth and in the beacons’ coordinates.

Since the travel times are used one by one, the same algorithm can be used with any number of beacons, without considering different equations depending on the number of travel times for a given cycle. It is clear, however, that the accuracy of the estimated position depends on the number of travel times available at each cycle. Using a single observation from time to time allows one to correct the vehicle position partially, but it is preferable to have at least two travel times to constrain the position estimate further.

3.3. Track initialization

Both approaches require an initial position estimate and an initial error covariance matrix from which they
oceans (Bellingham et al. 1994). As part of its sensor suite, the Odyssey II is equipped with a custom LBL navigation system (Atwood et al. 1995). During extensive field operations, the Odyssey II has collected a large amount of LBL data at various locations. Two of these data sets have been post-processed in order to illustrate the advantages and drawbacks of both approaches. The first experiment took place in Boca Raton, Florida, and the second in Buzzard’s Bay, Massachusetts.

The structure of the paper is as follows. Section 2 briefly describes the principle of LBL navigation. Section 3 presents the two algorithms with some comments on track initialization, outlier rejection and vehicle motion compensation. Section 4 discusses the advantages and drawbacks of each approach and presents some results obtained with the two above-cited data sets.

2. Long-baseline navigation principle

LBL navigation refers to an underwater positioning system employing an array of acoustic beacons separated by the order of hundreds of meters to a few kilometers. The array is usually calibrated through use of an additional acoustic transponder that is hung from a surface ship and interrogates the array from various locations.

The operation of our LBL system is as follows. In autonomous mode, the vehicle pings at a regular rate (every 5 s) at the 'master' frequency (9 kHz). Each acoustic beacon replies at its own frequency (10, 11, 12 and 13 kHz). If the average speed of sound is known, then the elapsed time between the initial ping and detection of the replies yields the range between the vehicle and the beacons. The navigation problem then consists of determining the vehicle position, given round-trip travel times measured from the vehicle to each beacon and the location of each beacon as determined in calibration.

Transponder-based navigation of AUVs may appear to be straightforward since each acoustic beacon can be uniquely identified from the frequency of its reply. Because of multipath propagation, noise and other error sources, however, the underwater LBL navigation problem can be difficult owing to the challenge of defining efficient autonomous outlier rejection procedures.

3. Presentation of the two approaches

Let us first define the following parameters. The local Earth frame origin is at the water surface. The x axis points north, the y axis points east, and the z axis points down. The vehicle position in this frame is \((x, y, z)\), and its orientation is defined by its yaw, pitch and roll angles \((\psi, \theta, \phi)\). The coordinates of beacon \(i\) are \((x_i, y_i, z_i)\). The measured round-trip travel times between the vehicle and beacon \(i\) is \(t_i\) and the associated distance is \(d_i\). The distances are computed from the travel times by the approximate relation \(d_i = c(t_i - \tau_i) / 2\), where \(c\) is the average speed of sound and \(\tau_i\) is the turn-around time of beacon \(i\) (typically 15 ms).

3.1. Fix computation approach

The fix computation approach consists of dead reckoning from the last acoustic fix and resetting the vehicle position as soon as a new fix is obtained. The dead-reckoned vehicle position \(X = [x, y]^T\) is calculated by means of the classical equation

\[
X_{k+1} = X_k + \mathbf{A}_k(\psi_k, \theta_k, \phi_k) V_k \Delta t,
\]

(1)

where the number and nature (water or bottom referenced) of the components of \(V_k\) depend on the speed sensor type, \(\mathbf{A}_k(\psi_k, \theta_k, \phi_k)\) is the rotation matrix allowing one to transform \(V_k\) from the vehicle frame to the earth reference frame, and \(\Delta t\) is the sampling interval.

Ideally, a new fix is obtained at every LBL cycle, although the number of returns may sometimes not be sufficient to compute a fix (0 or 1 return). Spurious travel times, however, can lead to inaccurate fixes which have to be discarded. A validation procedure is needed to determine whether a calculated fix is accurate enough to be used for a reset. Such a procedure is referred to as a 'fix domain' outlier rejection procedure because it runs on calculated fixes rather than on the travel times themselves. The details of the procedure are given in section 3.4.

We now present the least-squares algorithm allowing one to compute a fix from two or more travel times. It only optimizes the \(x\) and \(y\) position components and makes use of the vehicle depth \(z\) which is directly and accurately measured by the on-board pressure sensor.

The vector \(d = [d_1^T \cdots d_n^T]^T\) of the distances measured between the vehicle and the \(n\) beacons \((n \geq 2)\) during an LBL cycle can be related to the vehicle position by the following equation:

\[
d = f(X) + n,
\]

(2)

with

\[
f(X) = \begin{bmatrix}
[(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2]^{1/2} \\
\vdots \\
[(x - x_n)^2 + (y - y_n)^2 + (z - z_n)^2]^{1/2}
\end{bmatrix},
\]

(3)

where \(n\) is a Gaussian noise vector, with covariance matrix \(\mathbf{N}\), assumed to encompass error sources such as errors on the beacons' locations, on the travel time measurements, on the speed of sound and on the vehicle depth. The residual \(Q(X)\) to minimize under the Gaussian noise assumption is given by

\[
Q(X) = [d - f(X)]^T \mathbf{N}^{-1} [d - f(X)].
\]

(4)

The function \(f(X)\) being nonlinear, it is linearized about a reference position \(X_k\) by a first-order Taylor expan-
Comparison of fix computation and filtering for autonomous acoustic navigation

JÉRÔME VAGANAY†‡, JAMES G. BELLINGHAM† and JOHN J. LEONARD†§

Oceanographic data acquired by an autonomous underwater vehicle (AUV) must be correlated with accurate position information in order to be of value to scientists. Accurate navigation is therefore an essential requirement which can be fulfilled by use of acoustic long-baseline navigation systems and dead-reckoning sensors. Two approaches are presented in this paper. The fix computation approach consists of dead reckoning between fixes computed from a set of acoustic travel times. The filtering approach consists of correcting the vehicle dead-reckoned motion by taking into account the influence of the measured travel times. Fix computation is widely used for the positioning of manned submersibles and there seems to be some reluctance to switch to a filter-based approach, mainly because of the fear for divergence. After a detailed description of these two approaches, their advantages and drawbacks are compared by applying both algorithms to real data sets collected by the Odyssey II AUV developed at Massachusetts Institute of Technology Sea Grant. The filter presented in this paper is not subject to divergence. In the worst case, it would reject all the acoustic travel times measurements and proceed by dead reckoning only, so that re-initialization would be needed. This situation would also happen in the fix computation approach if the algorithm locked on an erroneous fix.

1. Introduction

One of the applications of survey-type autonomous underwater vehicles (AUVs) is the acquisition of oceanographic data for scientists. AUVs have indeed many advantages compared with existing systems. For instance, they are cheaper, easier to use, and faster and safer than towed systems. The critical problem to solve is that of navigation, since data have no significant value unless they can be associated with good estimates of the positions where they were acquired. Acoustic long-baseline (LBL) systems have the potential for accuracy. Owing to the low position update rate, however, these systems have to be associated with dead-reckoning sensors. LBL data are also affected by outliers due to noise and multiple propagation paths between emitter and receiver (Deffenbaugh 1994), so that outlier rejection procedures are required.

We consider two different approaches to vehicle positioning: fix computation and filtering. Fix computation consists of computing the least-squares solution corresponding to the intersection of spheres centered at the beacon locations with radii equal to the measured distances. The resulting fix is used to reset the dead-reckoned position. Outlier rejection is performed at the fix level. It consists of propagating the position uncertainty due to dead reckoning and defining validation regions for the validation of a new fix. When filtering, the vehicle position is predicted using dead-reckoning measurements and corrected using acoustic measurements by means of a Kalman filter, so that the estimated position is a weighted combination of acoustic and dead-reckoning data. Outlier rejection is performed by transforming the position uncertainty due to dead reckoning in the travel time domain.

Odyssey II is a low-cost deep-ocean-capable AUV developed at Massachusetts Institute of Technology (MIT) Sea Grant for scientific exploration of the
\[
    f(X_{k+1}) = f(X_k) + G[X_{k+1} - X_k],
\]

where
\[
    G = \left. \frac{\partial f(X)}{\partial X} \right|_{X = X_k}. \tag{6}
\]

Substituting (5) in (4), the residual at \(X_{k+1}\) can be expressed as follows:
\[
    Q(X_{k+1}) = \left[ d_t - GX_{k+1} \right]^T N^{-1} \left[ d_t - GX_{k+1} \right], \tag{7}
\]

with
\[
    d_t = d - f(X_k) + GX_k. \tag{8}
\]

The computation of \(X_{k+1}\) which minimizes \(Q\) is performed by means of the Gauss–Newton algorithm. The search direction \(s_k\) along which the criterion can be minimized starting from \(X_k\) in the case of a linear least-squares approximation is first determined:
\[
    s_k = (G^T N^{-1} G)^{-1} G^T N^{-1} \left[ d - f(X_k) \right]. \tag{9}
\]

The step along this direction is then determined by looking for the smallest integer \(k\), with \(k \geq 0\) and such that
\[
    Q(X_{k+1}) = Q(X_k + 2^{-k} s_k) < Q(X_k). \tag{10}
\]

If the criterion diminution is still greater than a preset threshold, the algorithm is reiterated with
\[
    X_{k+1} = X_k + 2^{-k} s_k. \tag{11}
\]

Otherwise, the minimization stops.

The covariance matrix of the estimation error is given by
\[
    C_X = (G^T N^{-1} G)^{-1}. \tag{12}
\]

When four beacons are deployed, as is the case in one of the two experiments described in the results section, the LCL cycles will most often give rise to a set of four returns. The least-squares algorithm can then be used with \(n = 4\). A single outlying distance, however, may prevent the algorithm from converging (the residual does not decrease), although a solution would have been obtained by running the least-squares algorithm on the three valid returns. The least-squares algorithm is then run with the four returns first and checked for convergence. If it did not converge, the least-squares solution is run with the four possible combinations of three returns. The three-beacon fix with the least residual is kept. In case there is more than one spurious distance, the algorithm does not provide any position update.

3.2. Filtering approach

The filtering approach consists of dead reckoning from an estimate of the vehicle position and incorporating travel time information to correct for drift. Valid travel times are used one by one as they arrive and their influence on the position estimate by their noise variance and the confidence in the predicted position. We limit our presentation to the state and observation models used in the filter, as the Kalman equations can be found in many sources (Bar-Shalom and Fortmann 1988). Preliminary simulations of the filter have been reported by Vaganay and Rigaud (1995).

The state vector is made up of the \(x\) and \(y\) components of the vehicle position, and of the velocity errors \(u_d\) and \(v_d\) expressed in the vehicle frame (alongship and athwartship respectively). These drifts are assumed to capture the influence of underwater currents, the effect of side slip during turns, and uncertainties on the measured velocity. Considering a vehicle able to measure its longitudinal velocity \(u\) only, the state equation can be written
\[
    X_{k+1} = \begin{bmatrix}
        x \\
        y \\
        u_d \\
        v_d
    \end{bmatrix}_{k+1} = \begin{bmatrix}
        x \\
        y \\
        u_d \\
        v_d
    \end{bmatrix}_k + \Delta t \begin{bmatrix}
        0 \\
        0 \\
        u + u_d \\
        v_d
    \end{bmatrix} + \begin{bmatrix}
        0 \\
        0 \\
        n_u \\
        n_v
    \end{bmatrix}. \tag{13}
\]

The returns from each beacon arrive asynchronously, depending on their position relative to the vehicle. The travel times are processed one by one at the sampling time immediately following their reception. The observation equation expresses a travel time as a function of the vehicle position, of the beacons’ coordinates, of the speed \(c\) of sound and of the vehicle depth \(z\) directly measured by the on-board pressure sensor:
\[
    t_i = \frac{2}{c} \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}. \tag{14}
\]

The travel time variance \(\sigma_t^2\) is taken to be large enough to encompass uncertainties in the speed of sound, in the measured depth and in the beacons’ coordinates.

Since the travel times are used one by one, the same algorithm can be used with any number of beacons, without considering different equations depending on the number of travel times for a given cycle. It is clear, however, that the accuracy of the estimated position depends on the number of travel times available at each cycle. Using a single observation from time to time allows one to correct the vehicle position partially, but it is preferable to have at least two travel times to constrain the position estimate further.

3.3. Track initialization

Both approaches require an initial position estimate and an initial error covariance matrix from which they
can proceed. Using an inaccurate position requires setting the initial position error covariance to a large value to account for the uncertainty in this position. This can lead to the validation of erroneous fixes or travel times and may prevent the algorithm from converging to the right track by rejection of the following valid fixes or travel times. For both approaches, it is then necessary to run an initialization procedure during which the vehicle dead-reckons from a rough initial position, before being reset by the output of the procedure.

The track initialization problem is directly related to the difficulty of validating a fix in the presence of outliers in the travel times; the vehicle initial position cannot be simply obtained by use of the first set of returns since these measurements may be affected by outliers. The initialization procedure would benefit from being based on acoustic data only. In these conditions, it is difficult to be able to validate or discard a fix by considering each cycle individually. A method has been proposed which consists of checking the consistency of fixes computed from subsets of the travel times available during a cycle (Atwood et al. 1995). This method is more attractive when at least four beacons are available owing to the sensitivity of two-beacon localization to calibration errors (Cestone et al. 1977). The approach used in this paper is inspired by the way that a human operator piloting a manned submersible would perform the task—by looking for consistency in the most recently calculated fixes as they would appear on a display. Fixes resulting from valid travel times tend to gather in the vicinity of the real vehicle track, whereas those including outliers tend to locate the vehicle at remote and inconsistent positions. During the initialization, least-squares fixes are computed from at least three returns. This allows one to avoid the ambiguity associated with the use of two-beacon fixes. The procedure assumes a rather linear motion and goes as follows.

A line is fitted to the \( M \) most recent fixes by minimizing the absolute deviation of the fixes with respect to the line. When a new fix is obtained, two thresholds are calculated: \( T_1 \) is an upper bound of the distance traveled by the vehicle since the line parameters were last computed, based on \( k \) times the vehicle speed; \( T_2 \) is the maximum authorized lateral deviation from the line for the new fix. For a fix to be considered the initial position, it is necessary that two successive fixes lie closer to the line than \( T_1 \) and be separated by less than \( T_2 \). \( T_2 \) then checks the alignment consistency perpendicular to the direction of motion, whereas \( T_1 \) checks the consistency of successive fixes along the line. In the result section, we used \( M = 5 \), \( k = 1.5 \) and \( T_2 = 15 \) m. Once the initial fix is obtained, it is used to reset the vehicle position and the fix error covariance matrix is used to initialize the associated position error covariance matrix.

### 3.4. Outlier rejection

The outlier rejection procedures used in each approach are very similar in that they are both based on the predicted vehicle position \( X \) (see (1) for fix computation and (13) for filtering) and the associated error covariance matrix \( P \). In both cases, \( P \) is propagated according to

\[
P_{k+1/k} = FP_{k/k}F^T + J_k C J_k^T + Q.
\]

In this equation, \( F \) is the Jacobian of \( X \) (unit matrix for fix computation), \( J \) is the Jacobian of \( X \) with respect to the speed and orientation measurements, \( C \) refers to the covariance matrix of these measurements, and \( Q \) accounts for modeling errors.

Under normal distribution assumption for the measurements, a validation region can be defined in the measurement space where the measurement should be found with a certain degree of probability (Bar-Shalom and Fortmann 1988). This validation region is defined in the fix domain for fix computation. It is defined in the travel time domain for filtering by transforming the uncertainty in \( X \) into an uncertainty in \( t_i \) by means of the Jacobian \( H \) of \( t_i \). The measurement validation test consists of applying a threshold to the normalized innovation squared:

\[
v_k^T S^{-1} v_k < \gamma.
\]

(16)

For the fix computation approach, when a new fix \( X'_k \) is obtained, together with its error covariance matrix \( P'_k \), we have \( v_k = X'_k - X_k \) and \( S_k = P'_k + P_k \). If a fix passes the test, then \( X \) is reset by \( X' \) and \( P \) is reset by \( P' \).

For filtering, when a new travel time \( t_k \) is obtained with its variance \( \sigma^2_t \), we have

\[
\begin{align*}
v_k &= t_k - \hat{t}_k, \\
S_k &= H_k P_{k+1/k} H_k^T + \sigma^2_t,
\end{align*}
\]

(17)

with \( H \) representing the Jacobian of \( t_k \). A travel time passing the test is used to correct \( X \), and \( P_{k+1/k} \) is corrected according to the classical Kalman equations.

### 3.5. Compensation of the vehicle motion during a cycle

When filtering, each travel time is related to the predicted vehicle position at the time of its reception by means of the observation equation. This equation is then not exact in that the vehicle moved between the ping and the reception times. The travel time should then be expressed as a function of the vehicle positions at the ping time and at the reception time. This can be approximated in the filter at the expense of a little more computation by dead reckoning the vehicle motion along each cycle and by applying appropriate corrections (Vaganay and Rigaud 1995). The position errors resulting from using a single position in the observation
equation are negligible at low speeds but can be significant at higher speeds. They also depend on the array configuration as well as on the vehicle direction of motion in the array. The correction mentioned above is applied to the results presented in this paper.

When computing fixes, the least-squares equations also assume that the vehicle pinged and received the replies at the same position although this is not true. Corrections can be applied by means of the 'running fix' method (Cestone et al. 1977). This method consists of modifying the beacons' coordinates used in the computations, based on the dead-reckoned motion between the ping time and the times of reception of each return. This allows one to compute the position at the ping time. Since a new fix is generally computed after a pre-set time out following the ping time or at the beginning of the following LBL cycle, it is necessary to add the dead-reckoned motion between the ping time and the current time to obtain the current position. This method has been used in the fix computation algorithm applied to the data presented in this paper.

4. Comparison of the two approaches on real data

The two approaches presented above have been applied to a variety of data sets acquired by the Odyssey II. We report results for two experiments.

- **Florida.** The experiment took place 1.5 km offshore near Boca Raton, in water depths of 10–20 m. The raw travel times measured during the vehicle motion are represented in figure 1.

- **Buzzard's Bay.** The experiments took place south of Massachusetts in an average water depth of about 12 m. The raw travel times measured during the vehicle motion are represented in figure 2.

Because a speed sensor was unavailable for these missions, a constant longitudinal speed \( u \) of 1.4 m\( \text{s}^{-1} \) is assumed (based on a previous thrust-speed calibration of the Odyssey II).

4.1. Track smoothness

Track smoothness when filtering is a direct consequence of the way that the filter works. As opposed to fix computation, the effect of a travel time measurement on the predicted position is weighted by the measurement variance and by the confidence in the prediction, rather than taken as an exact value. The vehicle track is then less subject to jitter in response to the noise on the acoustic data. The counterpart is that the good behavior of a filter highly depends on model accuracy and on good knowledge of the various noise variances. The state equation is sensitive to all the unknown parameters which induce dead-reckoning errors (underwater currents, heading sensor misalignment, magnetic compass fluctuations due to uncompensated magnetic fields generated by the vehicle, etc.). In general, the idea is to account for all the non-modeled errors in the state noise. In our case, using large noise levels for \( n_u \) and

![Figure 1. Raw round-trip travel times measured during the Florida experiment.](image-url)
\( n \) (13) helps the filter to adapt to sudden changes in the drifts due to turns for instance. Unfortunately, it also decreases the confidence in the position prediction, which leads to a less smooth track by validation and use of ‘not-so-good’ travel times. In the worst case, decreasing the confidence in the prediction could facilitate the validation of erroneous travel times and induce the rejection of all the following acoustic data. The vehicle would then end up dead reckoning and the filter would have to be re-initialized. In addition, the noise variances are generally set to fixed values, although they may vary in reality. With more sensor information it is possible to try to estimate the various errors source separately by use of algorithms adapted to the navigation conditions. This idea has been addressed by Vaganay and Rigaud (1995).

The smoothness of the filtered track compared with the track obtained by fix computation is clearly shown in figures 3 and 4 and in figures 5 and 6. In figure 5, the motion started in the upper left corner. The sudden change that can be observed after motion for about 300 m corresponds to the time that an initial fix was obtained and used to reset the vehicle position. As can be seen in figures 7 and 8, which show the position variation between two successive sampling times for each method, the position jitter of the filtered track is several orders of magnitude smaller than that of the fix-based track. The means and standard deviations of the position change when acoustic data are taken into account are shown in table 1.

Table 1. Mean and standard deviations of the position change

<table>
<thead>
<tr>
<th></th>
<th>Fix computation</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Standard</td>
</tr>
<tr>
<td>Florida</td>
<td>5.4</td>
<td>6.8</td>
</tr>
<tr>
<td>Buzzard’s Bay</td>
<td>5.4</td>
<td>9.0</td>
</tr>
</tbody>
</table>

Improving the dead-reckoning accuracy by use of accurate sensors would improve the overall performance of the filter. With a greater confidence in the prediction the track would be even smoother as the noise on the travel time measurements would be better filtered. On the other hand, although it would improve the selectivity of the outlier rejection procedure for fix computation, the smoothness of the track would not be improved since it would still jump from a noisy fix to another.

4.2. Travel time processing

In the filtering approach, a single travel time can be processed at the sampling time immediately following its reception. In the fix computation approach, it is necessary to wait for all the beacons to reply, and a fix can be calculated only if at least two returns have been received. Filtering then allows one to take advantage of more acoustic data than fix computation. The travel
times validated by the filter during each experiment are represented in figures 9 and 10. These can be compared with the raw travel times in figures 1 and 2. Results concerning the use of the measured travel times for the Florida and Buzzard’s Bay experiments are represented in table 2. The second column contains the total number of LBL cycles after initialization. The third column contains the number of cycles during which the filter used a

<table>
<thead>
<tr>
<th></th>
<th>Total number of cycles</th>
<th>Number of cycles with single travel time (filter)</th>
<th>Use of travel time (fix computation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Florida</td>
<td>861</td>
<td>18</td>
<td>92.6</td>
</tr>
<tr>
<td>Buzzard’s Bay</td>
<td>630</td>
<td>52</td>
<td>87.3</td>
</tr>
</tbody>
</table>
single travel time to correct the state. The fourth and fifth columns contain the percentages of use of the total number of returns for the filter and the fix computation approach.

In the tracks presented in figures 4 and 6, two-beacon fixes were allowed in order not to penalize the fix computation approach regarding the percentage of use of acoustic data. Because of their greater sensitivity to baseline miscalibration, however, these fixes largely participate to the track jitter. In fact the less accurate fixes in figures 4 and 6 result from the use of only two travel times. Figures 11 and 12 show the tracks obtained by precluding the use of two-beacon fixes. It can be seen that these tracks are a little smoother, but the percentages of use of acoustic data are also reduced to 86.7% and 67.4% for Florida and Buzzard's Bay experiments.
respectively. When four beacons are deployed, precluding the use of two-beacon fixes is not too restrictive and provides a smoother track. When navigating in a more common three-beacon array, however, there exists a trade-off between the desired track smoothness and the number of position updates.

4.3. Outlier rejection procedures

Outlier rejection can be performed at almost no extra computational cost when using the filter. Since the propagation of the position uncertainty during dead reckoning is already part of the normal computations of the filter, the only calculation that needs to be added is the
threshold of the normalized innovation squared, which determines the validity of each travel time.

For fix computation, outlier rejection has to be added. One possibility has been presented in this paper. Fix validation can also be viewed as an extension of the initialization problem, since a procedure that works to determine an initial fix should work just as well along

the complete track. Figure 13 shows the fixes that were selected by the initialization procedure (see section 3.3) iterated over the complete vehicle trajectory in Buzzard's Bay. These fixes are interesting in that they depend only very little on the dead-reckoning sensor measurements. The only drawback is that the position update rate is lower because several LBL cycles can be
necessary to obtain a fix. If dead reckoning can be trusted long enough, this is, however, a viable approach.

5. Conclusion
Filtering appears to be a more attractive solution essentially because of the smoother tracks that it provides. Another great advantage of the filtering approach is its ability to process travel times one by one as they arrive, so that cycles involving only one or two valid travel times can be used to update the vehicle position. Furthermore, as opposed to the fix computation approach, using a pair of travel times does not result in a position jump, since the effects of baseline miscalibration are filtered out. On the other hand, the computation of fixes is still required to initialize the vehicle track.
Running a good initialization procedure all along the vehicle motion can also provide valuable reference fixes and serve as a back-up for the filtering approach, or as a reference to check the good behavior of the filter.

Although filtering has several advantages compared with fix computation, it is not an easy solution to implement in an AUV. The filter behavior is highly affected by errors on the models and on the parameters. The risk of systematic rejection of travel times (because the filter used erroneous acoustic data) is always a concern, but it could be monitored to re-initialize the filter when needed. It has to be noted that systematic rejection of fixes can also happen when using the fix computation approach if the algorithm locks on an erroneous fix.

With appropriate sensors, a solution could be to use several filters, corresponding to different navigation conditions (Vaganay and Rigaud 1995), in cooperation with a fix computation approach. A bank of filters could allow estimation of model parameters. Fixes could be calculated in parallel and serve as a reference to check the good behavior of the filters every time that a fix is considered valid. Even if several consecutive LBL cycles are necessary to validate a fix, this information could still prevent the filter from malfunctioning. This highlights the need for a navigation supervisor which would manage the bank of filters, determine when a fix is valid, check the good behavior of the filters and determine whether a fix must be used to re-initialize the filters.

References


