Outlier Rejection for Autonomous Acoustic Navigation

Jerome Vaganay, John J. Leonard, and James G. Bellingham
Massachusetts Institute of Technology
Sea Grant College Program
292 Main Street, E38-300
Cambridge, MA 02139
U.S.A.

Abstract - Navigation is a critical requirement for the operation of Autonomous Underwater Vehicles (AUVs). In this paper we present acoustic navigation results for the Odyssey II AUV obtained by using a Kalman filter that integrates dead-reckoning with acoustic range measurements made to an array of acoustic beacons pre-deployed in the operating environment. Because spurious acoustic measurements due to multipath propagation are common, initialization and outlier rejection techniques are addressed. The navigation algorithm has been extensively tested by post-processing of real data acquired by Odyssey II during field operations in a variety of environments. These include the Charles River Basin, the Atlantic Ocean (1.5 km off the Florida coast), and the Pacific Ocean (375 km off the Oregon coast). Our results show improved performance over prior techniques based exclusively on fix computation and dead reckoning.

I. INTRODUCTION

For oceanographic data acquired by an autonomous underwater vehicle (AUV) to be of value to scientists, good navigation information is essential. Reliable navigation is also critical for the safe operation and recovery of the AUV. Because the ocean is impenetrable to electromagnetic energy except at very low frequencies, navigation systems such as LORAN and GPS are unavailable, and instead acoustic systems must be used. Acoustic navigation of AUVs can be very challenging because it requires autonomous processing of travel time measurements affected by noise, drop outs and outliers, as can be seen in Figure 2.

The predominant source of the spurious measurements is the presence of multiple acoustic propagation paths between source and receiver. This phenomenon of multipath is the result of reflections off the ocean surface and/or bottom and refraction of sound waves due to changes in sound speed with depth. The result is that a single ping by an acoustic beacon can be detected by the AUV as a complex sequence of arrivals [5]. The direct path can often be masked by destructive interference, resulting in a spurious travel time value.

The AUV must make decisions regarding the quality of the data without the aid of a human operator. The reliability of the solution must be very high, because a survey AUV cannot "stop and think" when confronted with a difficult situation; the vehicle must be continually in motion to maintain control authority. If the navigation algorithm were to go awry during sea-trials, sending the vehicle hundreds of meters off course, loss of the vehicle would be quite likely.

The Odyssey II is a low-cost, deep-ocean capable AUV developed at MIT Sea Grant for scientific exploration of the oceans [4]. As part of its sensor suite, the Odyssey II is equipped with a custom long base line (LBL) navigation system [1]. During extensive field operations, the Odyssey II has collected a large amount of LBL data in varying acoustic conditions. Analysis of this data has shown that navigation algorithms based on fix computation were not able to handle the problem of outlier rejection in a satisfactory manner. We therefore have pursued an approach that uses a Kalman filter to integrate dead-reckoning with acoustic range measurements to an array of transponders pre-deployed in the operating environment. Initialization and outlier rejection strategies, which are essential for the Kalman filtering to be successful in the presence of spurious measurements, are an important part of the algorithm.

The algorithm is applied to post-process three data sets obtained during different missions of Odyssey II. The first data set, which was obtained off the coast of Florida, exhibits travel time measurements with low noise and only a few outliers. The more challenging second data set, obtained in the Charles river in Boston, contains more outliers and noisier travel times. The third data set obtained in the Pacific over the Juan de Fuca ridge is the most challenging of the three, because an unexpected transponder left on the seafloor during a previous year's experiment interfered with the AUV's navigation system.

The structure of this paper is as follows. In Section II, the problem of LBL navigation of AUVs is described. Section III summarizes the navigation calculations performed during initialization and filtering. Section IV discusses the central problem of outlier rejection. Section V presents experimental results obtained with the algorithm and compares its performance against traditional fix calculation methods with and without outlier rejection. Finally, Section VI discusses our results and future research directions.

Figure 1. Acoustic long baseline navigation.

II. LONG BASELINE NAVIGATION

LBL navigation refers to an underwater position referencing system employing an array of acoustic beacons. Beacon separations for most vehicle applications are typically from hundreds of meters to a few kilometers [8, 3] (Figure 1). 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 the LBL system is as follows. In autonomous mode, the vehicle pings at a regular rate at the "master" frequency. Each acoustic beacon replies at its own frequency. If the speed of sound is known, then the elapsed time between the initial ping and detection of the reply yields the range between the vehicle and the beacon. The navigation problem is then to determine 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.

The AUV navigation problem is similar to the localization problem for indoor mobile robots that employ environmental features, such as walls and corners, as "geometric" beacons [10]. In comparison with these land robot localization methods, transponder-based navigation of AUVs may appear to be straightforward; each acoustic beacon can be uniquely identified from the frequency of its reply and the positions of the beacons are known from a calibration procedure. Because of multipath propagation and other sources of error, however, the underwater LBL navigation problem can be extremely difficult. Land robot research employing radar or ultrasonic beacons has encountered similar difficulties [9, 6].

III. NAVIGATION CALCULATIONS

Two different approaches to LBL position computation are possible: fix computation and filtering. In both cases, the vehicle dead reckons its position until LBL data are available. For three beacons, fix computation consists of computing the depth constrained analytical solution corresponding to the intersection of three spheres centered at the beacon locations with radii equal to the measured distances. The resulting fix is used to reset the vehicle position. In the second approach, the position is computed by means of a Kalman filter. The travel times and depth measurements are used as they arrive to correct the predicted position. This method does not explicitly compute a fix, but provides a position estimate which is a weighted combination of dead reckoning measurements and absolute measurements. This leads to a smoother track since the effect of a given set of travel times is only used to a certain degree to correct the predicted position instead of completely resetting it.

Our preference is a filtering approach. Fix computation is necessary for initialization of the filter, however, so we will describe both solution methods. The local earth frame origin is at the 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 the coordinates of beacon i are (x_i, y_i, z_i). The measured round-trip travel times between the vehicle and the beacons are t_i and the associated distances are 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[1] and τ_i is the turn around time of beacon i (typically 15 ms).

A. *Fix computation*

The analytical solution based on measurements from three beacons consists of solving the following non-linear system of equations for the vehicle coordinates:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = d_i^2, \quad i = 1, 2, 3. \quad (1)$$

In order to make the computations easier, we first compute the solution in an intermediate frame obtained by shifting the local earth frame at the location of beacon 1. The beacon coordinates in this frame are (x'_i, y'_i, z'_i) and the vehicle position is (x', y', z'). The intermediate solution is then transformed back to the local earth frame. Advantage is also taken of the accurate knowledge of the vehicle depth, leading to a linear over-constrained problem (3 equations and two unknowns). The vehicle position is then:

$$x = \frac{b_1 c_2 - b_2 c_1}{a_1 b_2 - a_2 b_1} + x_1$$

$$y = \frac{a_2 c_1 - c_2 a_1}{a_1 b_2 - a_2 b_1} + y_1 \quad (2)$$

[1] Spatial variations in the speed of sound are significant, especially variations with depth. Methods for computing an average speed of sound have been described elsewhere [7].

with:

$$a_1 = -2x'_2$$
$$b_1 = -2y'_2$$
$$c_1 = {x'_2}^2 + {y'_2}^2 + {z'_2}^2 - 2z'z'_2 - d_2^2 + d_1^2$$
$$a_2 = -2x'_3$$
$$b_2 = -2y'_3$$
$$c_2 = {x'_3}^2 + {y'_3}^2 + {z'_3}^2 - 2z'z'_3 - d_3^2 + d_1^2.$$

When measurements from four beacons are available, one could do a least squares minimization; however, in our work we have obtained good results for initialization by computing the four fixes from each set of three beacons and selecting the fix with the least residual error (Equation 5).

B. *Kalman filter*

We limit our presentation to the state and observation models used in the filter, as the Kalman equations can be found in many sources [2]. Preliminary simulations of the filter can be found in [12]. The state equation corresponds to dead reckoning, while the travel times and depth measurements are the measurements for the observation model. The position prediction stage of the filter is based on classical dead reckoning:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_k + {}^0A_v(\psi_k, \theta_k, \phi_k)\mathbf{V}_k\Delta t + \mathbf{v}_k \quad (3)$$

where ${}^0A_v$ is the rotation matrix from the local earth frame R_0 to the vehicle frame R_v. It is a function of the yaw, pitch and roll angles (ψ, θ, ϕ) provided by the vehicle attitude and heading system. The number of components of the speed vector $\mathbf{V}_k$ depends on the type of speed sensor available on board.

Depth measurements are readily obtained from a pressure sensor mounted on the vehicle, and hence the measurement equation for depth is simply $D = z + w_D$. The returns from each transponder arrive asynchronously, depending on their position relative to the vehicle. The travel times are processed one by one as soon as they arrive, whereas a depth measurement is available during each control cycle. There is no need to wait for a set of 3 or more travel times as it is the case for fix computation.

Whereas fix computation assumes the vehicle has not moved between the ping and the receptions, the observation equation for the travel times is expressed as a function of the position at the ping and at the reception of the return:

$$t_i = \frac{1}{c}\sqrt{(x_{ping} - x_i)^2 + (y_{ping} - y_i)^2 + (z_{ping} - z_i)^2} + \frac{1}{c}\sqrt{(x_{recept} - x_i)^2 + (y_{recept} - y_i)^2 + (z_{recept} - z_i)^2} \quad (4)$$

Essentially, this is a minor correction to reflect the fact that the geometric constraint is not really a sphere, but an ellipsoid with foci at the ping and reception locations. The vehicle position at the ping and at the reception can be expressed as a function of the current position by dead reckoning the vehicle displacement between the ping and the current position and the reception time and the current position.

IV. OUTLIER REJECTION

The two techniques presented in the previous section both assume that measurement errors consist only of zero-mean additive noise. As we have seen from Figure 2, however, the acoustic travel time data can possess a large amount of outliers. For LBL navigation of manned submersibles and remotely operated vehicles, bad fixes can be simply ignored by the human pilot of the vehicle. An AUV, on the other hand, cannot rely on a human in-the-loop. The crux of autonomous acoustic navigation, then, is to perform outlier rejection so that the two methods presented in Section III can be properly applied.

There are two different ways in which outlier rejection could be performed: in the time domain (travel time rejection) or in the spatial domain (fix rejection). Since it takes only one bad travel time value to result in an erroneous position, fix rejection has the potential to discard too much information. Therefore, temporal outlier rejection is desirable because erroneous travel times can be rejected individually. However, to reject data in the time domain, there must be an accurate *a priori* initial position estimate from which predicted arrival times can be generated for each of the beacons. Until the filter is initiated, such a dead-reckoned position estimate is not available, and so for initialization outlier rejection must occur in the spatial domain.

A. *Spatial domain (during initialization)*

The presence of outliers in the travel time measurements makes initialization of the algorithm difficult. The first set of three measurements cannot be blindly used to compute the initial fix since an outlier may induce a large position error which would in turn provoke the rejection of all the following fixes. Although the vehicle initial position is generally roughly known before the first LBL returns are detected, it is necessary to obtain a valid and accurate initial fix on which rejection of following LBL data will be based. When three or more beacons are used, the validity of measurements could be checked by thresholding the residual error at the estimated position:

$$\frac{1}{2}\sum_{i=1}^{n}(d_i - \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2})^2 < \tau \quad (5)$$

However, the choice of the threshold value τ is difficult, because it depends on the accuracy with which the bea-

con coordinates are known and on the noise on the measured distances.

A second possibility would be to compute fixes with subsets of the available travel times during an LBL cycle and look for consistency among them [1]. Only a valid set of travel times should lead to consistent fixes. This method is more attractive when at least four beacons are available. In that case, the four three-beacon fixes which can be produced are less subject to errors due to baseline miscalibration than the three two-beacon fixes associated with three-beacon navigation.

The approach used in our algorithm is inspired by the way a human operator piloting an ROV or manned submersible would perform the task — by looking for consistency in the most recently calculated fixes as they appear on the operator screen. 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 in remote and inconsistent positions.

Initialization can then be obtained by first determining the direction of motion of the vehicle and testing the following fixes for consistency with the estimated direction. In order to resolve the ambiguity due to LBL positioning with two beacons, it is preferable to check consistency among three-beacon fixes only. Assuming a linear initial motion for the vehicle, the direction of motion can be determined by a median fit rather than a least-square fit. This option gives less weight to outliers than the least square fit does.

The initialization procedure then goes as follows: fixes are computed if three travel times are available during an LBL cycle. If four travel times are available, the three beacon fix with the least residual is used. A line is fitted to the N 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: threshold T_1 is the distance traveled by the vehicle since the line parameters were last computed, based on the vehicle speed. Threshold T_2 is the maximum authorized lateral deviation from the line for the new fix, and is defined by $T_2 = \beta + T_1 tan(\alpha)$. ($\alpha = 1$ degree and $\beta = 5$ meters are used for the results in Section V.) For a fix to be considered the initial position, it is necessary that two fixes lie closer to the line than T_2 and be separated by less than T_1. T_2 then checks alignment consistency perpendicularly to the direction of motion, whereas T_1 checks consistency of successive fixes along the line. The validated fix is used to initialize the filter state vector. An estimate of the variance of the initial fix is derived by differentiating Equation (2) with respect to the measured distances, based on an assumed variance σ_t^2 for valid travel times. This initial variance is used to initialize the position error covariance matrix P_k.

B. Time domain (during filtering)

In the case of initialization, waiting several LBL cycles before being able to declare a fix as a valid initial fix was not too restrictive. It is, however, no longer the case for the rest of the motion where the vehicle has to be able to accept or discard LBL data as quickly as possible in order to maintain an accurate position. Furthermore, once an initial accurate fix is obtained, the short term accuracy of dead reckoning measurements allow one to predict more or less where the vehicle should be. Rejection of outliers after initialization should then definitely take advantage of this ability.

Some methods propagate the position uncertainty due to dead reckoning from the initial position, and therefore define a validation gate in space in which the next fix should be. A fix out of the validation region is then discarded, while a fix in the validation region is used to reset dead reckoning and reduce the position uncertainty. Other methods transform the above position uncertainty to the travel time domain, defining validation regions for each of the times of flight. A Kalman filter such as the one described above directly performs this transformation.

When the vehicle dead reckons, its position is predicted using the state equation (3). The associated error covariance is also predicted by:

$$P_{k+1/k} = P_{k/k} + J_k C J_k^t + Q_k \tag{6}$$

where C is a diagonal matrix containing the covariances of the speed components and orientation angles, J is the Jacobian matrix of the state vector with respect to the speed and orientation, and Q is the covariance matrix of the state noise (Equation (3)).

After propagation of the position error, the uncertainty in position is transformed into uncertainty in the predicted travel time using the observation equation (Equation (4)). Outliers are then gated using the well-known Mahalanobis distance [2]. When a new travel time arrives, its normalized innovation squared is compared to a threshold γ:

$$\nu_k^t S^{-1} \nu_k < \gamma \tag{7}$$

with

$$\nu_k = t_k - \hat{t}_k$$

$$S_k = H_k P_{k+1/k} H_k^t + \sigma_t^2$$

where H is the travel time Jacobian. A validated measurement is then used in the estimation step of the Kalman filter to correct the current predicted position.

V. EXPERIMENTAL RESULTS

The algorithm has been applied to a variety of data sets acquired by the Odyssey II in varying acoustic conditions. We report results for three different missions: 1) Florida, 1.5 km offshore near Boca Raton (January 18,

	beacon depth	water depth	duration
Florida	5 m	10-20 m	70 min
Charles River	1 m	3-10 m	60 min
Juan de Fuca	2000 m	2000 m	190 min

Table I
Beacon depth, water depth, and mission duration.

	cycles	ping rate (s)	σ_t (ms)	γ
Florida	8	5	3.3	20
Charles river	11	5	3.3	20
Juan de Fuca	29	10	10.0	20

Table II
Number of cycles required for initialization, ping rate, travel time standard deviation σ_t, and validation gate size γ.

	B_1	B_2	B_3	B_4
Florida	97.8	97.0	75.3	98.4
Charles river	75.6	78.8	69.2	60.6
Juan de Fuca	59.7	54.4	48.6	-

Table III
Percentage of returns used for each beacon.

	0	1	2	3	4
Florida	3.8	1.7	10.5	35.0	48.9
Charles river	3.9	3.6	12.6	34.4	45.5
Juan de Fuca	30.8	16.3	24.4	28.5	-

Table IV
Percentage of cycles using n travel times.

1995), 2) Charles River Basin (December 6, 1994) 3) Juan de Fuca ridge, 375 km off the Oregon coast (June 28, 1995). For the first two missions, four beacons were deployed, while for the Juan de Fuca mission only three beacons were used. Table I provides more information for the experiments. For each of the three missions, we show the vehicle track produced by the Kalman filter with outlier rejection, superimposed on fixes computed with outlier rejection, and the fixes obtained by using all the input data, without outlier rejection.

The solid lines on Figures 3, 6, and 7 show the output of the Kalman filter algorithm. On these figures the letters a and b, respectively, represent the assumed initial position (from which the vehicle dead reckons until the initial fix is computed) and the initial fix determined by the initialization method. The vehicle dead-reckons from a until b is obtained. Fix b is then used to initialize the filter and its variance is used to initialize matrix P_k. The time domain outlier rejection takes over and the travel times are filtered as the Kalman filter proceeds. Table II shows how the quality of the LBL data affects the number of cycles required for initialization. With a median fit to the last 5 fixes, initialization took only 8 cycles for the best environment (Florida), but 29 cycles for the worst (Juan de Fuca).

The crosses on Figures 3, 6, and 7 correspond to the fixes computed during each run using the following validation procedure. We used the method described in Section III-A with an adapted outlier rejection method very similar to that used with the Kalman filter. The only difference is that it works in the space domain instead of the time domain. The vehicle position is first initialized as described in Section IV-A. As when using the Kalman filter, the vehicle position is predicted by dead-reckoning, and the position uncertainty is propagated by Equation (6). When a new fix is computed (regardless of the validity of the travel times), its variance is also computed. The fix is validated or discarded by computing its normalized innovation squared and comparing it with a threshold, as in Equation (7). Since there can be four three-beacon fixes when four beacons are used, the fix which passes the test with the least normalized innovation is selected to reset the dead-reckoned position. The covariance matrix P_k is also reset with the variance of the selected fix.

It can be seen that fix computation with outlier rejection works quite well for a friendly environment such as Florida (Figure 3). Performance rapidly degrades as the acoustic environment becomes harsher (Figures 6 and 7). Although these fixes contain jitter, they give a rough idea of the actual vehicle track.

Figure 4 shows the travel times which passed the validation test (Equation 7) for the Charles river experiment and were actually used to correct the vehicle position. The efficiency of the outlier rejection can be observed by comparison with the corresponding raw travel times in Figure 2. Table III shows how the quality of the acoustic environment affects the amount of valuable LBL data. For the good acoustic environment of Florida, the travel times were used at more than 95 %, except for beacon 3. For the Charles river experiment, the number of outliers increases, and the percentage of travel times used drops to between about 60 and 80 %. With the unexpected transponder in the Juan de Fuca experiment, it decreases to about 55 %.

Since travel times can possibly be discarded during each cycle, the position correction can be performed on a subset of the measurements. It is desired to have as many valid returns as possible during a cycle to completely constrain the vehicle position. As a matter of fact, although the filter is able to process a single travel time for a given cycle, the use of this data only partially corrects the vehicle position. Table IV shows the percentage of the total number of LBL cycles during which none, 1, 2, 3 or 4 travel times were used. Almost 50 % of the corrections involved 4 travel times for the Florida and Charles river experiments. Only three beacons were used in Juan de Fuca, and only 28.5 % of the corrections actually used all three beacons. Note that 30.8 % of the

LBL cycles were useless since no correction occurred. In spite of the very bad quality of the measurements, the algorithm was able to provide a nicer track than that obtained by fix computation (Figure 7).

Figures 5, 8, and 9 show dramatically the importance of outlier rejection. These plots show the result of the fix computation algorithm and the associated outlier rejection method described above by setting the threshold to a very large value. The result of doing this is that all fixes are accepted, so that the algorithm behaves as if no outlier rejection were performed. Even for the Florida data, the tracks are unacceptable because the vehicle position jumps several hundreds of meters from one fix to the next. The worst case is shown by the Juan de Fuca run, where completely spurious data from the interfering transponder provokes the vehicle position to jump by several kilometers.

VI. DISCUSSION

Kalman filtering and time domain outlier rejection results in a much smoother track than with fix computation and space domain outlier rejection. One source of error in fix computation is that the vehicle motion between the ping and the reception of the returns is not taken into account.

Because a speed sensor was unavailable for these missions, a constant speed of 1.4 m/s is assumed (based on a previous thrust/speed calibration of the Odyssey II). This assumption introduces errors into the Kalman filter. These errors are greatest during turns, when side-slip occurs, and during depth changes, when the speed during ascent is greater than during descent because the vehicle is trimmed to have slight positive buoyancy.

The algorithm is currently being integrated into the main vehicle software suite to run on-board the vehicle. Several at-sea trials are scheduled in the coming months, where we will test the ability of the vehicle to execute waypoints using the algorithm reported here. The vehicle is being fitted with a Doppler Log, which will add a speed measurement capability. This should yield more accurate tracks and provide the ability to estimate water currents [12].

In the future, it would be desirable to implement a more theoretically advanced technique for outlier rejection and initialization. In our view, the ideal approach would be to employ Reid's multiple hypothesis tracking (MHT) algorithm for this purpose [11]. This algorithm is attractive because it can postpone decisions concerning the validity of measurements until more data is acquired by setting up a tree of data interpretation hypotheses. The computational requirements, however, of the MHT are potentially severe; work on efficient pruning and hypotheses management strategies would be necessary to allow real-time execution on an AUV. The MHT, however, assumes a track can be initialized from a single measurement, so some form of fix computation technique as described above would be necessary at start-up.

ACKNOWLEDGEMENTS

Support for this work was provided by MIT Sea Grant under grant number NA46-RG-0434, by the Office of Naval Research under grant number N00014-92-J-1287, and by the National Science Foundation under grant number IRI-9224640. The authors would like to thank Dr. D. K. Atwood, Dr. B. A. Moran and Dr. J. W. Bales for their assistance.

REFERENCES

[1] D. K. Atwood, J. J. Leonard, J. G. Bellingham, and B. A. Moran. An acoustic navigation system for multiple vehicles. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, September 1995.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.

[3] J. G. Bellingham, T. R. Consi, U. Tedrow, and D. Di Massa. Hyperbolic acoustic navigation for underwater vehicles: Implementation and demonstration. In *Proceedings AUV '92*, pages 304–309, 1992.

[4] J. G. Bellingham, C. A. Goudey, T. R. Consi, J. W. Bales, D. K. Atwood, J. J. Leonard, and C. Chryssostomidis. A second generation survey AUV. In *IEEE Conference on Autonomous Underwater Vehicles*, Cambridge, MA, 1994.

[5] M. Deffenbaugh. A matched field processing approach to long range acoustic navigation. Master's thesis, Massachusetts Institute of Technology, 1994.

[6] H. F. Durrant-Whyte, E. Bell, and P. Avery. The design of a radar-based navigation system for large outdoor vehicles. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 764–769, May 1995.

[7] D. B. Heckman and R. C. Abbott. An acoustic navigation technique. In *IEEE Oceans '73*, pages 591–595, 1973.

[8] M. Hunt, W. Marquet, D. Moller, K. Peal, W. Smith, and R. Spindel. An acoustic navigation system. Technical Report WHOI-74-6, Woods Hole Oceanographic Institution, 1974.

[9] L. Kleeman. Optimal position and heading for mobile robots using ultrasonic beacons and dead-reckoning. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2582–2587, Nice, France, May 1992.

[10] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. Robotics and Automation*, 7(3), June 1991.

[11] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, AC-24(6), December 1979.

[12] J. Vaganay and V. Rigaud. Autonomous navigation: an example of localization algorithm switching based on navigation conditions. In *7th Int. Conf. on Advanced Robotics*, Sant Feliu de Guixols, Spain, September 1995.

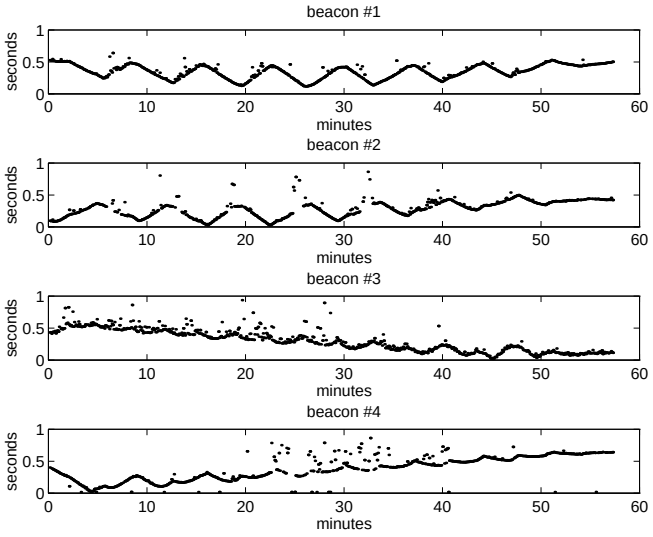

Figure 2. Round-trip acoustic travel time measurements for Charles River experiment.

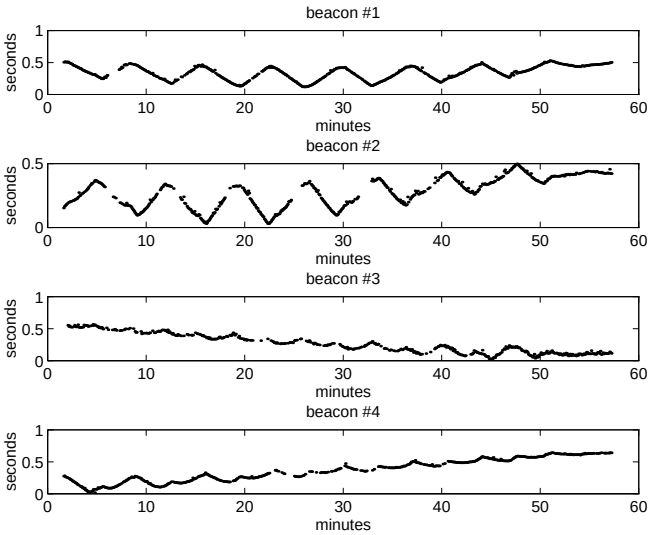

Figure 4. Acoustic travel time measurements after outlier rejection by Kalman filtering for Charles River experiment.

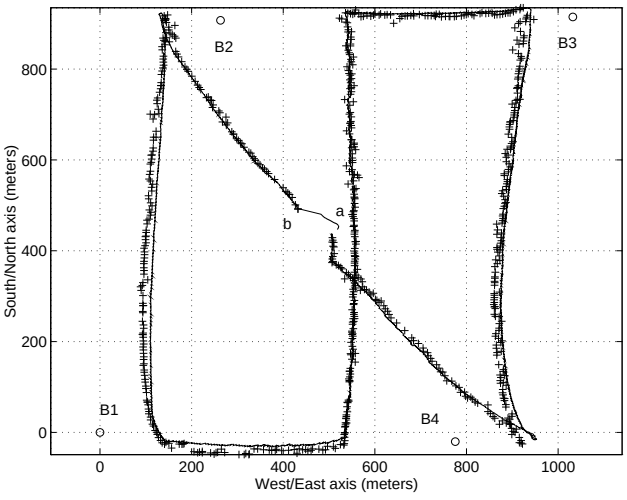

Figure 3. Vehicle track for Florida Experiment. The solid line shows the Kalman filtered trajectory, using time domain outlier rejection after initialization. The crosses indicate fixes computed with spatial domain outlier rejection.

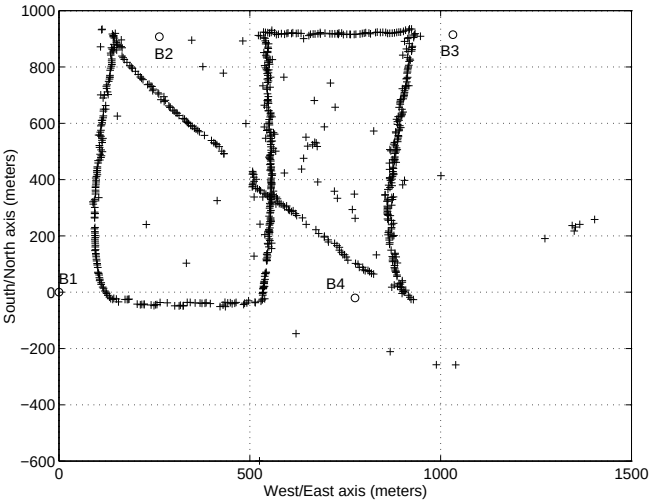

Figure 5. Fixes without outlier rejection for Florida experiment.

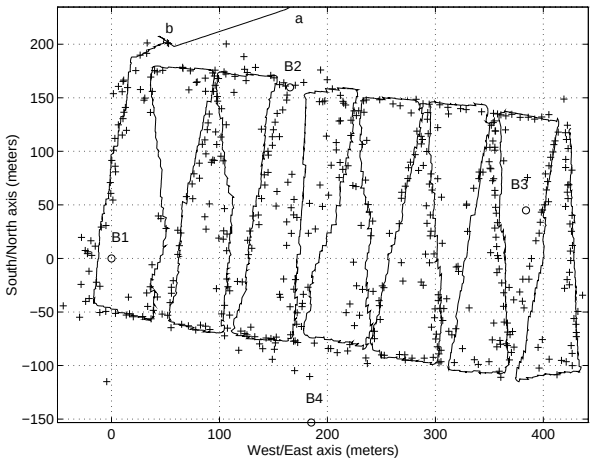

Figure 6. Kalman filtered vehicle track and fixes for Charles River Experiment.

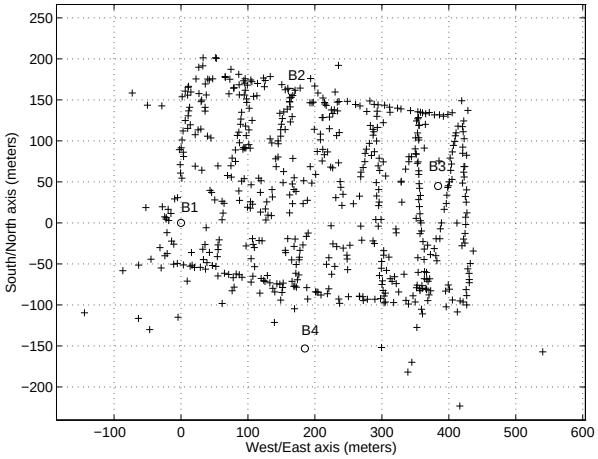

Figure 8. Fixes without outlier rejection for Charles River experiment.

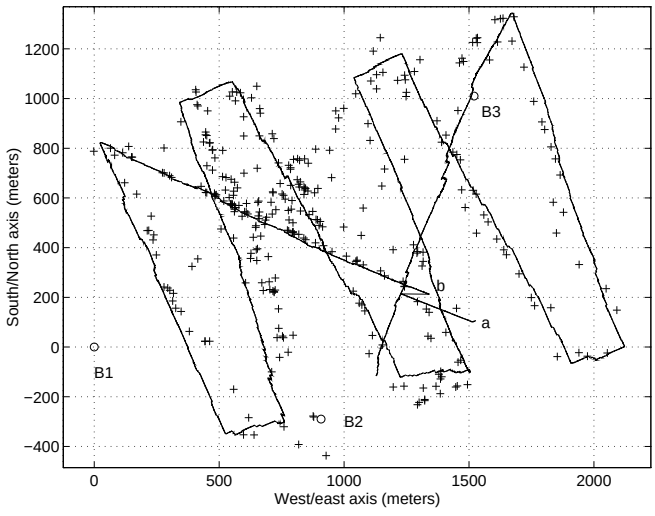

Figure 7. Kalman filtered vehicle track and fixes for Juan de Fuca Experiment.

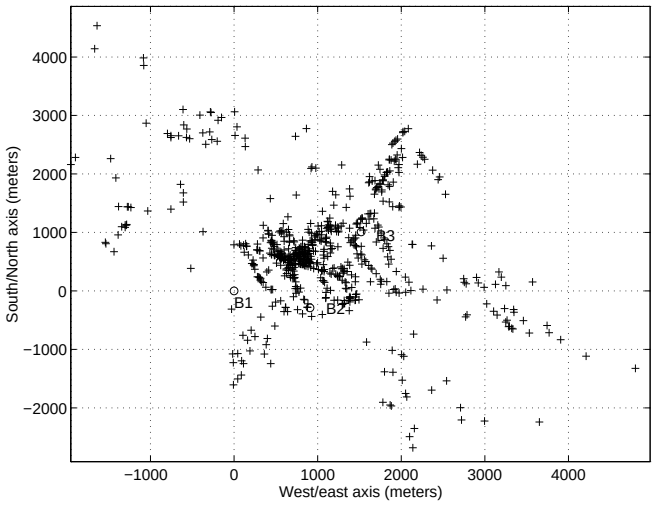

Figure 9. Fixes without outlier rejection for Juan de Fuca experiment.