High Speed Localization Using a Multiple Analytical Distribution Filter
(Short Title: Multiple Analytical Distribution Filter)

Dov Kruger[1], Hongyuan Shi[2], Mina Samaan, Donald Chesley[a], Jeffrey V. Nickerson[b], Len Imas[a]

ABSTRACT

ABSTRACT

Probabilistic localization of robots is often done using Kalman filters, and for better results, with particle or grid filters. Particle and Grid filters require significantly more computation, as they must either consist of large number of particles, with schemes to resample as particles drift into regions of low probability, or as grids with an error due to the finite size of the grid resolution. We present a new approach, of computing an analytical distribution for each data input, separately transforming them to account for the physical motion of the vehicle, the equivalent of the forward step in Kalman filtration, and computing a new grid for each required timestep rather than discretizing distributions onto a grid and intersecting the distributions to define a minimum area for consideration. The resulting computation is considerably faster than particle or grid filters because non-intersecting areas can be ignored. A grid of probabilities is evaluated only at the end of the process, in the most restricted, intersection region, without maintaining a grid in memory. Expected position and variance are computed directly by multiplying probabilities from the underlying analytical distributions.

1. INTRODUCTION

Navigation underwater in an estuary uses data of low-reliability. Salinity and temperature gradients can cause sound refraction, echoes, and changes in the speed of sound that affect the accuracy and reliability of acoustic instrumentation. Surfacing makes accurate navigation easy using GPS, but exposes the Unmanned Underwater Vehicle (UUV) to detection, and in the case of our research in the busy Hudson River Estuary, to the likelihood of collision with large vessels.

Underwater, available navigational inputs include acoustic beacons and dead reckoning (DR). DR accuracy can be dramatically enhanced using either inertial navigation or Doppler Velocimetry (DV). Both are accurate in the short term and with increasing error over longer intervals. Their behavior in estuarine conditions will likely be worse than in homogenous waters. For this paper, we are restricting the instruments to those available on our current UUV, namely Long Baseline (LBL) acoustic beacons, GPS, and DR, without the benefit of any inertial or DV measurement, though that will dramatically improve performance.

LBL is therefore the sole source of external navigational data underwater in our experiments. The operational problem is that providing a field of beacons to cover anything but a small area is impractical, and the data from the beacons in estuarine water is made very noisy by the difficult acoustic conditions, a function of noise, sharp salinity gradients which make sound curve along isohalines, and multipath echoes[1].

Our focus is on high efficiency algorithms to estimate location from the available navigational inputs, and to be able to form an estimate of position with only partial navigational data at any one point in time, a probabilistic version of what is classically called a running fix[2]. However, we also present results from acoustic beacon measurements used to determine location. The raw errors from the acoustic beacon were ±20m, with approximately a 10% failure rate where the answers were completely ridiculous. It appears possible to correct distances calculated acoustically and significantly reduce the error for the readings that are correct. By collecting multiple readings over time and having an overdetermined solution, it should generally be possible to identify which readings are wrong as well.

2. ALGORITHMS OF LOCATION ESTIMATION

Localization of a robot has heretofore been typically done in one of three ways. Kalman filtering, particle filters, and grid filters.

The original and most popular estimate is a Kalman filter, which assumes a set of estimates X={x_i, 1£ *i* £ *n*}, are normally distributed with variance V={v_i, 1£ *i* £ *n*}. The Kalman estimate is computed iteratively on every pair of elements with a weighted linear combination of X, where the weights are inversely proportional to the corresponding variances V:

$$\hat{x}=\frac{v_2x_1+v_1x_2}{v_1+v_2}$$

,

$$\hat{v}=\frac{v_1v_2}{v_1+v_2}$$

.

A physical model is applied to transform the estimates at time *t* into inputs at time *t* + 1 and obtain a new estimate[3]. One problem with Kalman filters is that the variance V depends on the environmental condition and is often not known if it varies as a function of time. More important, all filters in the Kalman family assume an analytical distribution. For the original Kalman filter, the normal distribution is assumed. Subsequent refinements of Kalman filters (extended, etc.) may assume different distributions, but none can take advantage of an observed non-Gaussian distribution, because they all parameterize with a single value. For example, if the likely location is distributed in two Gaussians separated by a distance, the only way to represent the situation is to define a single point in the middle and vastly increase the variance to cover the entire area, as shown in Fig. 1. The obvious disadvantage is that a great deal of information about the actual distribution must be discarded to fit the analytical representation.

Particle filters offer a solution, using a Monte Carlo approach, in which a number of weighted particles represent the distribution, thus capturing distributions with no analytical solutions, including multiple peaks[4]. However, particle filters require many particles to cover the entire area to a desired accuracy, and as particles drift out of the area of highest probability, the filter must reseed them in to maintain accurate coverage, or the effective number of particles is reduced.

Intrinsically more pleasing to the authors is a grid approach, which is deterministic, where the cost is known in advance ($O(n^2)$, where *n* is the grid resolution) and the resulting maximum error can also be computed in advance based on the grid resolution. A grid can be computationally expensive however, as its cost increases with the resolution in two dimensions in both computation and memory, but this is the same effective cost as particle filters, since the accuracy of the random particles is *O*($\sqrt{n}$), where *n* is the number of particles[5]. Since most of the grid has near-zero probability, due to the various instrument readings ruling out most of the space from consideration, an efficient technique to ignore the zeros would be clearly advantageous. Based upon our experience with acoustic navigation using LBL beacons, we present a new algorithm, using significantly less computation than grid filters, but retaining nearly all the accuracy as well as the deterministic behavior of a high resolution grid method. Because this algorithm maintains multiple individual distributions, each of which can be described analytically, we call it a Multiple Analytical Distribution Filter (MADF).

3. THE MULTIPLE ANALYTICAL DISTRIBUTION FILTER (MADF)

The concept behind the Multiple Analytical Distribution Filter is to maintain the accuracy of particle filters by computing the estimate of probability of each location by intersecting all the analytical distributions of each navigational input. In order to avoid the large number of particles and the expense of a grid, MADF computes regions of intersection. Each data source has a characteristic distribution. Using the limited set of distribution types and computing only the intersections generates relatively small regions which can then be traversed to compute estimates of location and variance.

For cases with only fractional navigational data, MADF can be augmented to track previous input signals within a window. By propagating past navigation data forward in time and increasing the variance accordingly, an estimate can be produced even when there is not enough data at that instant in time. The limiting factor is the accuracy of dead reckoning.

Last, even if there is insufficient data to precisely localize at time *t*, it is possible to gain additional information afterwards which can be shifted backward in time. Thus for retrospective collection of scientific data, a more accurate estimate can be made after the fact.

Data Sources for Underwater Navigation

There are a number of sources of navigational data, including GPS, Long Base Line (LBL) acoustic beacons, dead reckoning, digital velocimetry, inertial navigation, and others. Here we restrict the discussion to the first three, but the method is completely general.

A GPS signal is available only on the water surface, but provides highly accurate location with a reasonable error estimate. It can be reasonably viewed as a bivariate normally distributed circle of position as shown in Fig. 2.

$$p(x,y)=\frac{1}{2\pi}\exp\left[-\frac{(x-\mu_x)^2}{2\sigma_x^2}-\frac{(y-\mu_y)^2}{2\sigma_y^2}\right]$$

where *x*, *y* is the location whose probability is to be computed, m_x, m_y is the GPS location, s_x, s_y are the standard deviations for x and y. Most entry level GPS units report this as a single horizontal error estimate.

The LBL acoustic beacons are based on acoustic modems. The UUV sends a short ping to a set of named beacons and they reply after a 50ms delay. The total time of travel is thus twice the distance between the beacon and the UUV plus the 50ms lag time. The location of the beacons is known based on GPS reading when initially deployed. The estimate from an acoustic beacon can be viewed as an annular ring of probability around the known beacon position, shown in Fig. 3.

$$p(r) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(r - r_0)^2}{2\sigma_r^2} \right]$$

where r_0 is the measured distance from the beacon, r is the distance whose probability is to be computed, and σ_r^2 is the variance representing the thickness of the annulus.

The distance to the beacon, r_0, is computed from the delay and the speed of sound in water, calculated as a function of temperature, salinity and pressure. Any empirical formula with sufficient accuracy will do, so we use Coppens' empirical equation[6]:

$$c_{0,s,t} = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + (c_4 + c_5 t + c_6 t^2)(s - 35)$$

$$c_{d,s,t} = c_{0,s,t} + (c_7 + c_8 t) d + (c_9 + c_{10} t) d^2 + \left[c_{11} + c_{12}(s - 35)\right] \cdot (s - 35) \cdot t \cdot d$$

where t is the temperature in degrees Celsius, s is the salinity in practical salinity units (psu), d is the depth in meters, $c_{0,s,t}$ is the sound speed at the surface, $c_{d,s,t}$ is the sound speed at depth d, and c_i, $0 \le i \le 12$, are the coefficients of the empirical fit[6].

Dead reckoning traditionally consists of logging the speed and direction of the vessel and summing it to estimate position. With computers, it is now possible to continuously integrate accelerations, but the principle remains the same. There are three errors associated with dead reckoning, the steering angle, the speed travelled, and a random drift due to environmental currents. The first two are intrinsic to the accuracy of steering and motor servo control loop, while random drift is reduced in effect as the ratio of vehicle speed to environmental current speed increases. An inertial navigation unit or digital velocimetry can reduce the errors by measuring the accelerations in the case of the inertial unit, and velocities for the DV. This results in the following distribution, shown for a 10m DR track with an exaggerated steering error for clarity (Fig. 4).

We approximate the worst-case DR error as a bivariate normal so it is analytically tractable.

$$\sigma_{steer} = \sqrt{\sigma_d^2 + (d \cdot \sigma_{head})^2}$$

To account for random environmental fluctuations, another time-dependent term is added:

$$\sigma_{DR} = \sigma_{steer} + \sigma_{env}(t) = \sqrt{\sigma_d^2 + (d \cdot \sigma_{head})^2} + (\Delta T)\delta_{env}$$

The variance, σ_{DR}^2, may then be used to transform any analytical distribution in the time domain to then intersect it with another analytical distribution collected at a different time. Each DR step can therefore be treated as normally distributed in each dimension, and can be added to the analytical distributions for any navigational instrument such as GPS or LBL beacons.

Intersecting Propagated Distributions

Having shown that the probability distribution of DR is not exactly symmetric, but close, we consciously neglect the asymmetry. It simply spreads the distributions and makes them less accurate, but the process is the same as for computing the intersection at a single point in time. After transformation, a GPS would have increased variance

$$p(x,y) = \frac{1}{2\pi}\exp\left[-\frac{(x-\mu_x)^2}{2(\sigma_x^2+\sigma_{DR}^2)} - \frac{(y-\mu_y)^2}{2(\sigma_y^2+\sigma_{DR}^2)}\right]$$

Similarly, for the annular case, if at time t_0 the variance is σ_0^2, then after DR, the variance of the distribution can be approximated as

$$p(r) = \frac{1}{\sqrt{2\pi}}\exp\left[-\frac{(r-r_0)^2}{2(\sigma_r^2+\sigma_{DR}^2)}\right]$$

provided

$$\sigma_{DR}^2 << r_0$$

By the time the DR variance is the order of the beacon radius, the distribution is no longer an annulus, but it is also so diffuse as to be almost useless for navigation. It can be approximated as a bivariate normal as is the GPS, except that the radius, intead of being a few meters, is the distance to the beacon, on the order of hundreds of meters.

Intersecting Probability Distributions

The result of intersecting multiple analytical distributions will never be easy to process analytically, which is the reason that particle filters are popular. There are two ways of solving this problem. First, one can intersect just two regions, and define algorithms specifically for this case. We do this for annular sections and bivariate normal distributions, i.e. LBL beacons and GPS. Second, having found a region of intersection, in order to make it analytically manageable, a canonical bounding shape must be defined. The three obvious choices are circles, ellipses, and rectangles. We have chosen rectangles because they have good properties for intersection, namely an easy algorithmic breakdown into a number of cases, each of which uses only two algorithms: testing for containment of a point in a quadrilateral, and determining the point of intersection of two sides.

For the special case of two annular distributions, the geometry is shown in (Fig. 5). The distances r_1 and r_2 are measured acoustically. Since the location of each beacon is known from GPS (x_1, y_1 and x_2, y_2) the distance d_3 can be computed.

$$
\begin{aligned}
d_1 + d_2 &= d_3 \\
r_1^2 &= d_1^2 + h^2 \\
r_2^2 &= d_2^2 + h^2
\end{aligned}
\qquad (1)
$$

To determine the two points of intersection, start at the known beacon point (x_2,y_2), compute the vector to p, then the vectors to S_1 and S_2. Given two annuli, the solution is changed to include an outer and an inner radius chosen to define a region that is likely to contain the location of the vehicle. The bounds of the region are where the outer radius of each circle intersects the inner radius of the other. One approach would be to then cover this area with a grid and compute the intersection of all the probabilities from all the other intersecting distributions (Fig. 6a).

To further restrict the area and compute fewer probabilities, determine a rectangle of coverage spanning the entire curve. Initially the rectangle is larger than the area from the annular sector, but if there are sufficient rectangles to intersect, then the resulting area can be made smaller, with correspondingly less computational effort.

After a final bound is selected, either the intersection of two distributions or the intersection of multiple rectangles, the area is, in the end, scanned in a grid to compute the best estimator of the location, and the best estimator of variance:

$$
\begin{aligned}
\hat{x} &= \sum_i p_i x_i & \sigma_x^2 &= \sum_i p_i (x_i - \hat{x})^2 \\
\hat{y} &= \sum_i p_i y_i & \sigma_y^2 &= \sum_i p_i (y_i - \hat{y})^2
\end{aligned}
$$

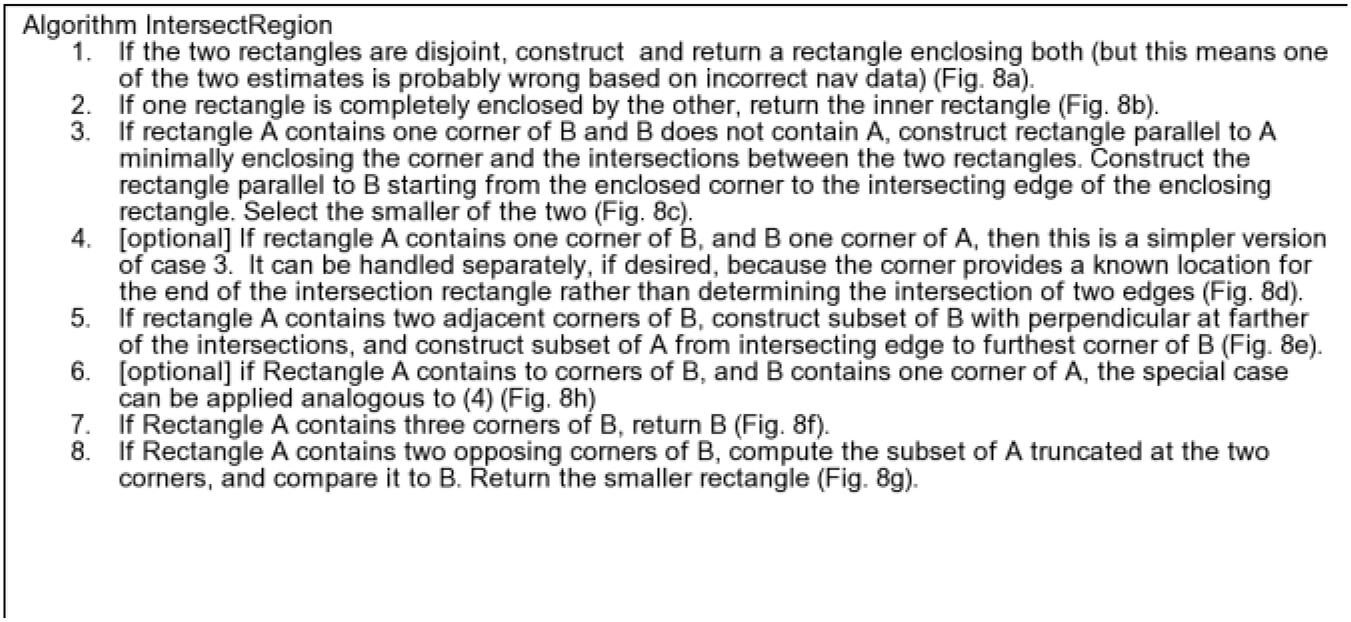
As shown in Fig. 6 above, for a single intersection, the optimal coverage is easily computed by scanning the intersection area. The starting and ending points P and Q are computed twice using Eq. (1) but with d_1 as the specified radius from the circle, and d_2 as the inner radius, and then d_2 as the outer radius.

GPS is a range-based system similar in principal to LBL acoustic beacons. If we had access to raw data including satellite location, and range estimation, we could use that to generate an improved estimate from multiple GPS readings. However, most GPS receivers, including the ones used in our experiments, yield only the estimated location and horizontal error. Thus the raw information which would be used by MADF to improve the estimate from multiple GPS readings has already been stripped out.

Rectangle Intersections

As the number of intersected areas rises, an increasingly irregular shape defines the probability distribution. However, even if a simple shape is used to define a simplified boundary, successive range estimates can shrink the size of the shape. Fig. 7 shows the region of intersection of two circles, and the relative efficiency of using a circle, ellipse, or rectangle as the bounding region to circumscribe the intersection area. The circles shown in the diagram are the outside bound of the area to be considered, such as 3 standard deviations out from the estimated position. To efficiently traverse the region of interest, there are a number of algorithmic choices. One can build a circle enclosing the region, but the size may approach the size of the original navigational circle, a large area. More efficient is to represent the area as an ellipse, but ellipses are not analytically tractable. Third, the region can be represented as a rectangle. The three alternatives are shown by red dashed lines in Fig. 7.

While the rectangle does not yield the tightest bound for a single instance, it is far easier to intersect further, and the subsequent intersections between the rectangles will often yield a tighter bound than the original bounded intersection of only two distributions. Rectangle intersection need only be performed if there are sufficiently accurate (i.e. intersecting) set of rectangles that will net reduce the area to be summed. The following algorithm generates a near-minimal intersection rectangle efficiently, using only simple tests of point inclusion to decide how to intersect each pair of rectangles.



```
Algorithm IntersectRegion
    1. If the two rectangles are disjoint, construct  and return a rectangle enclosing both (but this means one
       of the two estimates is probably wrong based on incorrect nav data) (Fig. 8a).
    2. If one rectangle is completely enclosed by the other, return the inner rectangle (Fig. 8b).
    3. If rectangle A contains one corner of B and B does not contain A, construct rectangle parallel to A
       minimally enclosing the corner and the intersections between the two rectangles. Construct the
       rectangle parallel to B starting from the enclosed corner to the intersecting edge of the enclosing
       rectangle. Select the smaller of the two (Fig. 8c).
    4. [optional] If rectangle A contains one corner of B, and B one corner of A, then this is a simpler version
       of case 3.  It can be handled separately, if desired, because the corner provides a known location for
       the end of the intersection rectangle rather than determining the intersection of two edges (Fig. 8d).
    5. If rectangle A contains two adjacent corners of B, construct subset of B with perpendicular at farther
       of the intersections, and construct subset of A from intersecting edge to furthest corner of B (Fig. 8e).
    6. [optional] if Rectangle A contains to corners of B, and B contains one corner of A, the special case
       can be applied analogous to (4) (Fig. 8h)
    7. If Rectangle A contains three corners of B, return B (Fig. 8f).
    8. If Rectangle A contains two opposing corners of B, compute the subset of A truncated at the two
       corners, and compare it to B. Return the smaller rectangle (Fig. 8g).
```

The resulting rectangle representing the intersection of all navigational data is evaluated in a grid at the desired resolution to determine estimated position and variance (Fig. 9).

Sample Case

A UUV is started at an initial point (0, 0) with an accurate GPS reading. It dives at a heading of 45° for a distance of 100m. At that point, it pings and receives a distance from the beacon at location A. Without additional information, this is insufficient to further refine the propagated GPS estimate at this point if only symmetric circular errors are considered. It then moves on another 100m on the same heading, at which point it pings beacon A and gets another distance check. This time, the intersection of the three allows the algorithm to discard the erroneous intersection possibility at *q* and focus on the intersection at *p*, which is far smaller than the propagated GPS fix as changed by 200 m DR (Fig. 10).

Retrospective Transformation to Correct a Scientific Dataset

Navigational accuracy is not always available in realtime. Sometimes, accuracy is low, but the next moment could bring a fresh input that precisely locates the UUV. In such cases, the location can be recomputed after the fact to take advantage of the new information that has arrived. This can be done as a batch process after a UUV brings back scientific data, in order to more precisely identify the location at which data was taken, or it may be done continuously as the UUV is in motion, updating the track in hindsight as new information becomes available. The procedure is exactly identical to the forward time procedure, except that DR is performed in reverse and future measurements as well as past are used to calculate the location. In the example shown above, retrospectively we can fix our location at *p*. This approach can also be used to improve navigation with a window. The algorithm follows:

```
Algorithm MADF-window
1.   Compute MADF estimate of location and store in window for the useful lifetime of
     the data. That is to say, until the DR errors are great enough that the oldest data is
     no longer useful, at which point it may be discarded.
2.   If the accuracy of the current fix is greater than one in the past, perform MADF
     including the current estimate to refine the past estimate. This method is not
     iterative; repeated use can skew the estimates toward one point.
```

4. EXPERIMENTAL PLAN

The localizations experiments were conducted during the course of testing the UUV path planner, described in our companion paper[7]. The path planner would often request trajectories that are deep to avoid unfavorable currents. The beacons were to be positioned along the course of the UUV so that at least two should be in range at all times, and on occasion, more might be available simultaneously to provide corroboration. The UUV used in the experiment is a Nekton Ranger (mass about 15kg and length around 60cm) equipped with a GPS receiver, a conductivity-temperature-depth (CTD) sensor and an acoustic modem. It would first go on an upstream mission, then a downstream, and finally a cross channel mission, attempting to gather a steady stream of beacon data. The UUV would porpoise for a GPS fix every 30 seconds for the first test, then every 150 seconds and 300 seconds for tests where the planner wanted the UUV to stay down deep. The GPS estimated position was therefore to be interpolated when under water, and the accuracy is increasingly unknown as time increased from the nearest GPS fix.

To characterize LBL signal accuracy and noise, we also planned a surface experiment with a boat to simultaneously gather ground truth with a GPS, and continuously ping two beacons with a deckbox. Two beacons, one in shallow water (3 m) and one in deeper water (9 m) were place approximately 1m off the bottom. A CTD at each beacon was to record the salinity, temperature and depth for computing the speed of sound at the beacon end. A CTD on the boat was periodically lowered to the bottom, capturing the vertical profile of salinity and temperature. By circling the beacons, it was hoped we would be able to detect any error in beacon placement, and then by making passes up and downstream, identify the error in acoustic distance measurement as a function of distance from the beacon, and possibly salinity and temperature profile. The theoretical accuracy of Micromodem beacons as stated in their manual is 9.6cm. In homogenous conditions, the distance agrees with GPS readings to within 1m.

5. EXPERIMENTAL RESULTS

In the first trial, acoustic conditions were not ideal, and only 200 navigational pings were received. Maximum distance measured was 692 m. The UUV continuously measured temp/salinity with an onboard CTD, while position was estimated from GPS fixes by linearly interpolating; with dive times of 30 seconds, the UUV was at most 16 seconds away from a GPS lock forward or backward in time, and errors of up to 20 meters were observed, with a consistent over-estimate which we attributed to curvature of sound or reflection. Because of the interpolation of GPS readings, it is not completely clear how much of the error was due to the acoustic beacons vs. the interpolation. Ordinarily, errors of up to 18m were measured. Fig. 11a shows the distribution of errors between GPS estimated distance and LBL beacon-derived distance. The distribution is not normal, possibly due to the small number of samples, or due to errors introduced in linearly interpolating GPS location. Occasionally, a spurious reading would give a completely absurd answer. In the first experiment, these false readings were always too short. We now believe them to be random noise spoofing an acknowledge packet, for which the Micromodem returns an arbitrarily short return time. When far enough from an existing position estimate, these spurious readings can easily be rejected. However, doing so requires a steady stream of navigational input for verification.

In an attempt to confirm the results, we used a boat with continuous GPS navigation simultaneous with readings from a transducer. Errors were typically less than 18 m as in the previous experiment, but this time consistently *underestimating* the distance by approximately 8 m. Consecutive errors were not independent, and we observed errors to grow steadily to a maximum of 18 m and then shrink again. Fig. 11b shows that the distribution after removing the outliers (values outside 3 times the deviation) is statistically close to a normal distribution, based on the Shapiro-Wilk normality test (the test statistic $W = 0.993$, p-value = 0.06). For the boat experiment, the beacons returned obviously incorrect values approximately 10% of the time. As before these errors were usually underestimates, but there were also a few overestimates, when presumably the real acknowledge is not heard due to noise, and then a spoofed acknowledge makes it appear that the beacon is further away than is the case.

Combining the navigational values from different times requires an accurate dead reckoning to avoid diffusing the navigational data from different times too much for it to usefully intersect data from other times. At a drift error of .5m/s, two readings from 20 seconds apart are already broadened by ±10 m which is impractical. Our drift errors with UUV have not been quantified yet due to a lack of data to establish ground truth, but with an inertial navigation unit, and digital velocimetry we hope to be able to precisely define the DR accuracy and provide an easy way to determine when an acoustic beacon result is inconsistent with prior readings.

The physical experiments showed that there were far too many unknowns in the experimental setup. Our biggest problem is a lack of accurate ground truth; we do not know how accurate the GPS fixes were. Instead of comparing MADF to particle filters on the experimental data, we built a simulation which has perfect knowledge of the real location of the UUV, added normally distributed noise to reported beacon distances, and compared localization using MADF (sweeping through each intersection with 4´4 grid as in Fig 6a), particle filters using 200, 500, and 1000 particles. The grid filter is also computed as a reference. The particle filter is implemented according to the adopted version for wireless sensor networks[5]. The simulation is setup as in Fig. 12. The beacons form an equilateral triangle and location A is at the center and location B is at one corner. In the simulation, as the number of particles and the grid resolution increases, MADF, particle filters and grid filters converge to the same answer, as expected. For fewer particles, MADF results in lower errors with high probability, as shown in Fig 13. The reason for the poor particle filter performance is a loose bound on the initial position, so the particles are not sampling efficiently. However, if we were to tighten the bound, then sometimes a beacon intersection would be outside the sampled region, and particle coverage of the high probability region would be nonexistent.

We also benchmark the processing time for MADF and particle filters in our simulations. Table 1 lists the total CPU time for 10,000 iterations of each method. MADF is considerably more efficient than a particle filter with just 200 particles, while providing more accuracy. If we constrained the particles to a tighter grouping, accuracy would go up (efficiency would of course, remain the same) but reliability would drastically be reduced in the event of disagreement between navigational signals.

6. FUTURE WORK

The experimental focus until now has been to obtain a tighter bound on the errors in the raw navigation inputs from the LBL beacons in order to determine the accuracy of position estimates. The next phase of testing in the Hudson River is to try to use the UUVs operationally to acquire data over an area on a regular basis. In the process, a large amount of data will be collected on the performance of LBL acoustic beacons under a wide range of environmental conditions. We will try to increase the localization accuracy, both with our current equipment and a new UUV equipped with inertial navigation and DV. A new location estimate will be computed and compared with particle filters to test the relative accuracy of each technique. All the acoustic tools must be validated in estuarine environments, and will almost certainly not perform as well as they have in homogenous settings.

We are currently comparing simplistic particle filters against a basic MADF implementation. The next step is to compare a better particle filter algorithm which samples a distribution, translates by the physical model (model the motion between fixes) and recomputes the weights. We believe that the margin will be tighter, but that MADF is still computationally more efficient and simpler than particle filters with resampling.

There are also extensions to MADF that may increase its utility and scope. First, it may be possible to break down the area of intersection into triangular regions instead of the current rectangles. Triangles are more efficient for covering only the area of interest, but multiple triangles would require multiple intersection tests, so it is unclear if the result is more efficient than the current algorithm. Also, given that a grid of probabilities is being computed, higher order interpolation schemes might be used to reduce grid error. There are also other potentially useful navigational inputs with unique distributions to consider, such as RADAR/SONAR/laser range finding returning information about range and angle simultaneously, and using bottom scanning SONAR to recognize local features as a very precise way to measure relative motion.

In attempting to determine the position of a UUV continuously, it is necessary to identify incorrect navigational inputs so that they can be appropriately ignored. Developing an efficient algorithm that can discard bad ones will significantly reduce error over current techniques.

We believe it may be possible, based on measuring the temperature/salinity profile, to predict when values are wrong, and to reduce RMS error to less than 8 m. Decreasing the errors from acoustic beacons, coupled with improved accuracy of dead reckoning using inertial navigation and/or DV instruments we are aquiring would provide a high-accuracy way of building a feature map, from which navigation could then be done with high repeatability without beacons.

The Nekton Ranger will also be calibrated in the Davidson Laboratory tank to derive drag and maneuvering coefficients to improve dead reckoning accuracy in the absence of any other instrumentation. Last, a series of tests in the river should help to identify the accuracy of model forecasts, and compute a local assimilation of the data to try to reduce local disagreement between model and measured currents.

7. CONCLUSIONS

The MADF algorithm is nearly equivalent to a grid filter, with the slight reduction in accuracy due to treating the DR distribution symmetrically. MADF is computationally far more efficient in the cases where the intersection area is small, both by restricting the number of probabilities computed and by computing on the fly and not requiring the overhead of a grid in memory. MADF is therefore a good candidate to compete with particle and grid filters in any navigational application, such as using GPS in conjunction with an accelerometer or other means of odometry. Urban use of GPS, with signals frequently eclipsed by buildings, is an ideal application. Given access to the underlying data per satellite, GPS is exactly analogous to the LBL beacon system used in the experiments described, except that it is more accurate. Theoretically, other applications where particle filters are now used, such as financial applications would also be acceptable to MADF. However, the biggest cost of particle or grid filters is in maintaining accurate coverage of a two-dimensional (or greater) area; MADF is therefore likely to be competitive only in situations where the dimension d ³ 2, with a bigger advantage on 3-dimensional problems than on 2-d ones.

Like particle and grid filters, MADF will support estimating position with 3 or more inputs, with only 2 inputs in conjuction with a prior estimate, or even with only a single estimate at each timeWhen there are 3 simultaneous navigation readings for example, MADF should be able to efficiently calculate the best estimate of position and variance. When there are only two simultaneous navigational inputs, the algorithm can generate a best estimate of position and variance provided there is a prior position estimate that can reject one of the two possibile solutions. With only one navigational input, MADF can create a full positional estimate if there is sufficient parallax with prior estimates; if not, the error bound in one dimension is tight, but the other will be poorly bounded. Finally, if at some point, more information becomes available, it is possible to go back through the record and add information to prior estimates of position. In many cases, this will mean that it is possible to reduce the error estimate of position after the fact. Coverage with a particle filter would require twice the number of particles, since half would be advanced through time, and half would be transformed backward through time. With MADF, having more intersections narrows the area of interest, and the resulting area is even smaller. With minimal memory requirements, MADF appears to be a prime candidate for replacing particle filters as a navigation algorithm on lightweight CPUs running UUVs.

The acoustic experiments appear to show that LBL beacons are giving very high quality measurements (less than 3m error up to 1400m ranges, often less than 1m at ranges of 300-500m) but that environmental conditions are adding up to 20 m errors to the measurement; these environmental errors are not independent, rather they are sequentially related. If further experiments show that we can completely or partially predict the errors by measuring salinity and temperature gradients, then it might be possible to fix position to within 3 m underwater. However, since a high density of beacons is impractical, it is essential to be able to use a single beacon and correlate multiple readings over time. For this, some accurate measurement of DR estimation is essential, and either accurate inertial navigation or digital velocimetry will dramatically reduce the errors.

8. ACKNOWLEDGMENT

This research was supported by the Office of Naval Research, Grant #N00014-05-1-00632, Navy Force Protection Technology Assessment Project.

9. REFERENCES

1. H. Shi, D. Kruger, and J. V. Nickerson, “Incorporating environmental information into underwater acoustic sensor coverage estimation in estuaries,” *Proceedings of 2007 AFCEA/IEEE Military Communication Conference*, IEEE, October 2007.
2. N. Bowditch. *American Practical Navigator*, Paradise Cay Publications, 2002
3. G. Welch, G. Bishop, “An Introduction to the Kalman Filter”, *Technical Report*, 95-041, University of North Carolina at Chapel Hill, Department of Computer Science, July, 2006.
4. S. Thrun, W. Burghard, D. Fox, *Probablistic Robotics.* Cambridge, MA: MIT Press, 2005, pp. 96-104.
5. M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, *A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking*. IEEE Transactions on Signal Processing. Vol 50 No. 2, 2002, pp174-188.
6. Technical Guides - Speed of Sound in Sea-Water. National Physical Laboratory.[Online]. Available: http://resource.npl.co.uk/acoustics/techguides/soundseawater/content.html#COP
7. D. Kruger, H. Shi, M. Samaan, J. V. Nickerson, L. Imas, “Dynamic UUV Path Planning in an Estuarine Current Field”, in review with *Journal of Underwater Acoustics*.

LIST OF FIGURE CAPTIONS

Fig. 1. Kalman filters are not good at representing non-Gaussian distributions (in this example, bimodal)
Fig. 2. The probability distribution of a GPS location, with only a horizontal error estimate
Fig. 3. The probability distribution of an LBL beacon
Fig. 4. The conservative analytical distribution representing dead reckoning error
Fig. 5. Determining the intersection point of two beacons.
Fig. 6. (a) Sweeping the intersection region and (b) determining a bounding rectangle.
Fig. 7. Three choices of bounding shape.
Fig. 8. Different cases for intersecting rectangles used in algorithm IntersectRegion
Fig. 9. Once the smallest rectangle is calculated, probabilities must be computed on a grid
Fig. 10. Sample case, navigating with two beacon pings 30 seconds apart. Red line shows DR track

Fig. 11. Histogram showing the error of distance estimation from LBL vs. GPS based on data obtained from (a) the UUV operation experiment, and (b) the boat experiment.
Fig. 12. Simulation setup to compare MADF with particle filter using 200, 500, and 1000 particles. Two locations *A* and *B* are tested with 1000 trials.
Fig. 13. CDF of simulated errors from MADF (4´4) and particle filters with 200, 500, 1000 particles at (a) location A and (b) location B.

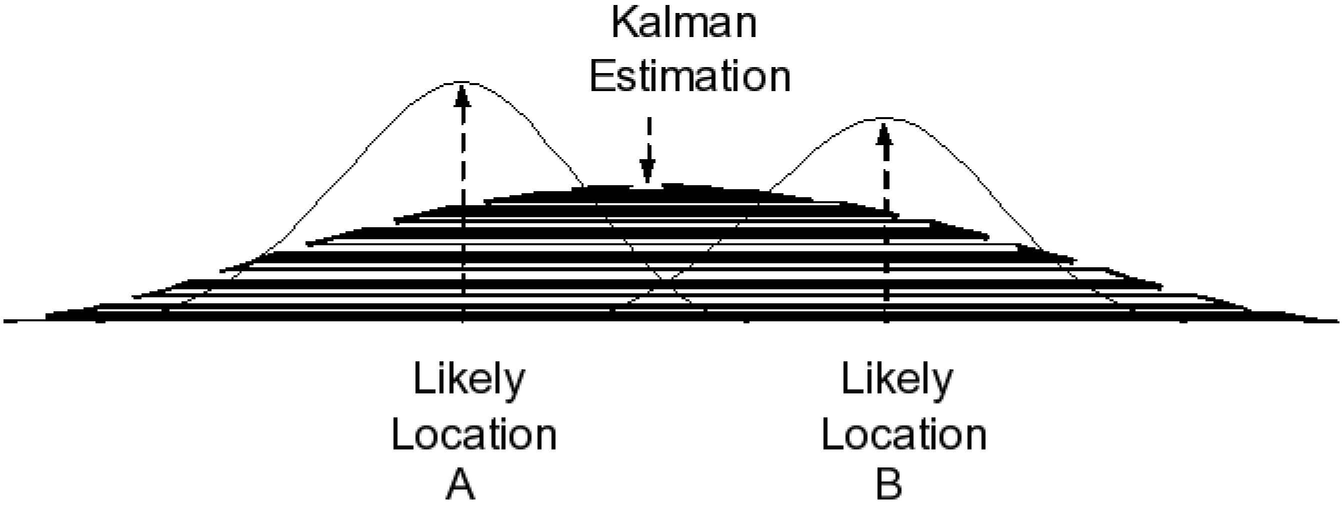


Fig. 1. Kalman filters are not good at representing non-Gaussian distributions (in this example, bimodal)

(Generated using Microsoft Visio)

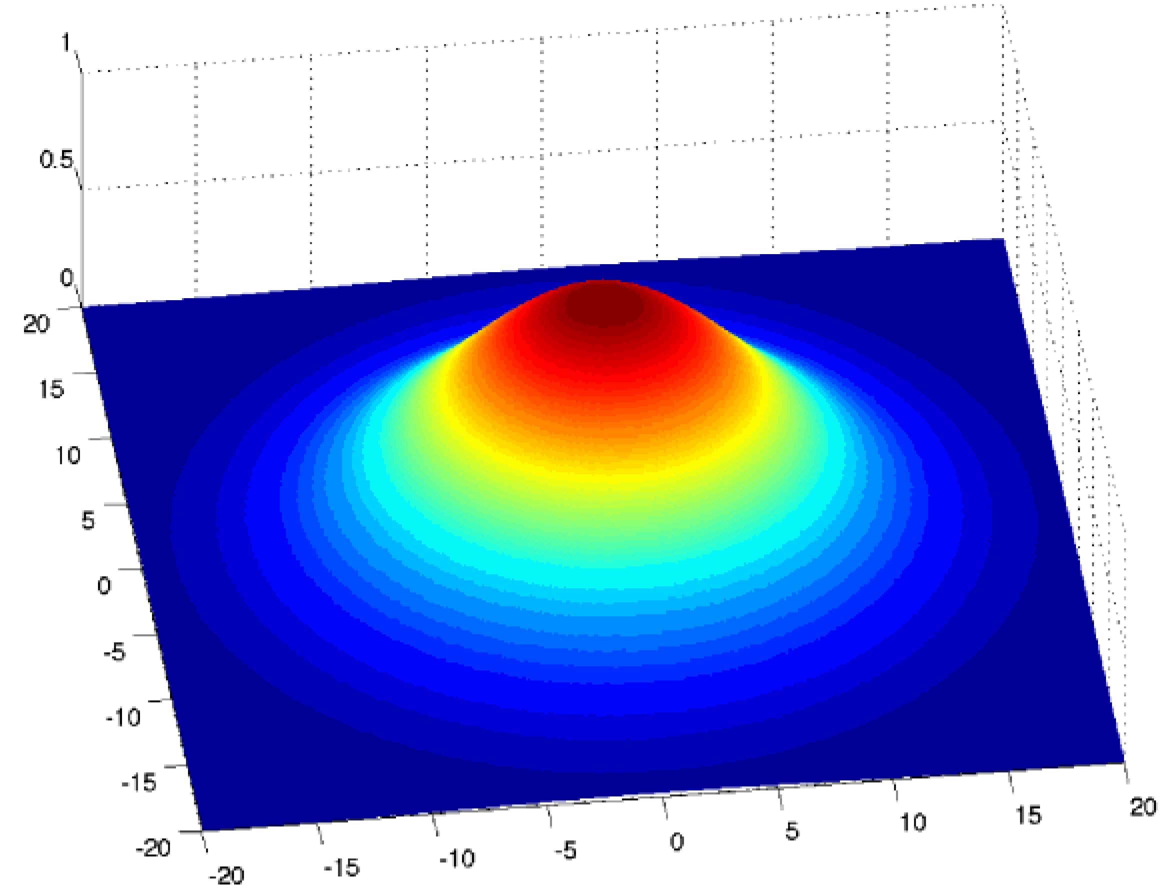


Fig. 2. The probability distribution of a GPS location, with only a horizontal error estimate (Generated through Matlab)

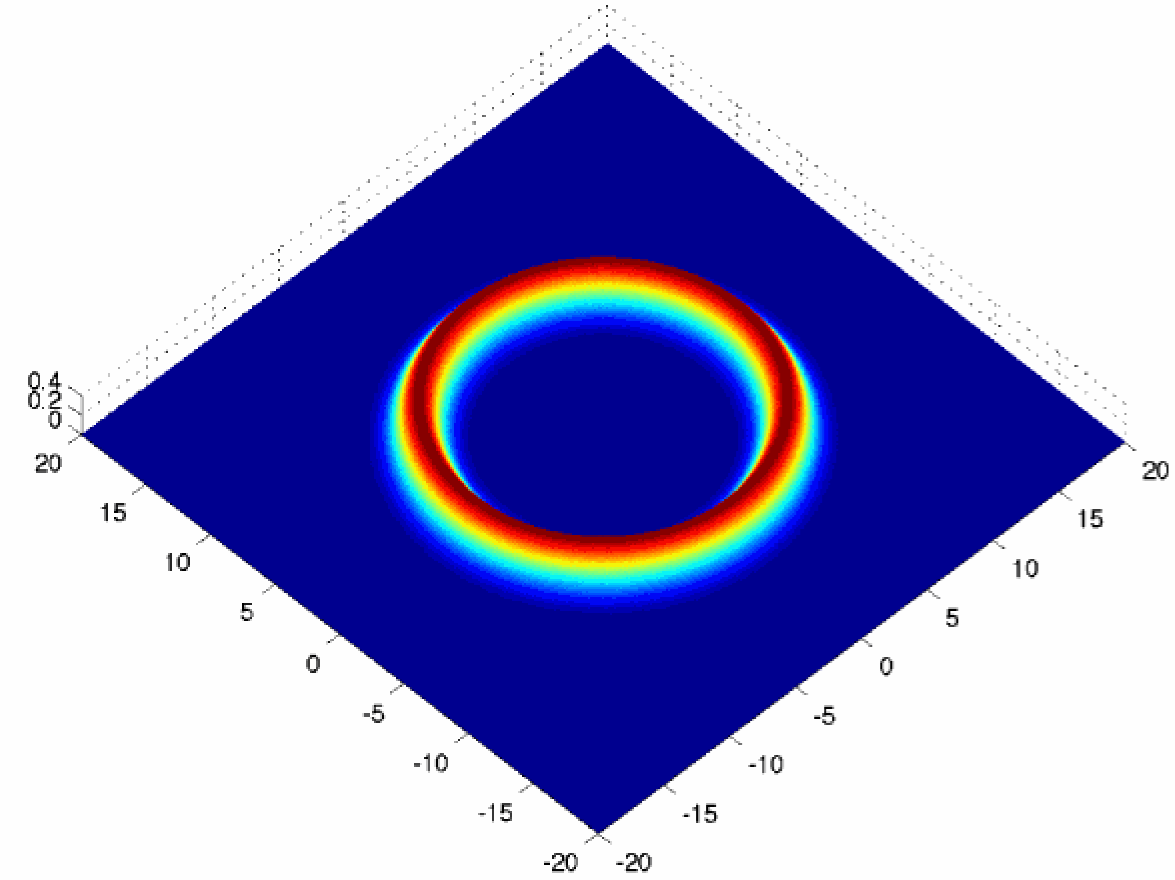


Fig. 3. The probability distribution of an LBL beacon
(Generated through Matlab)

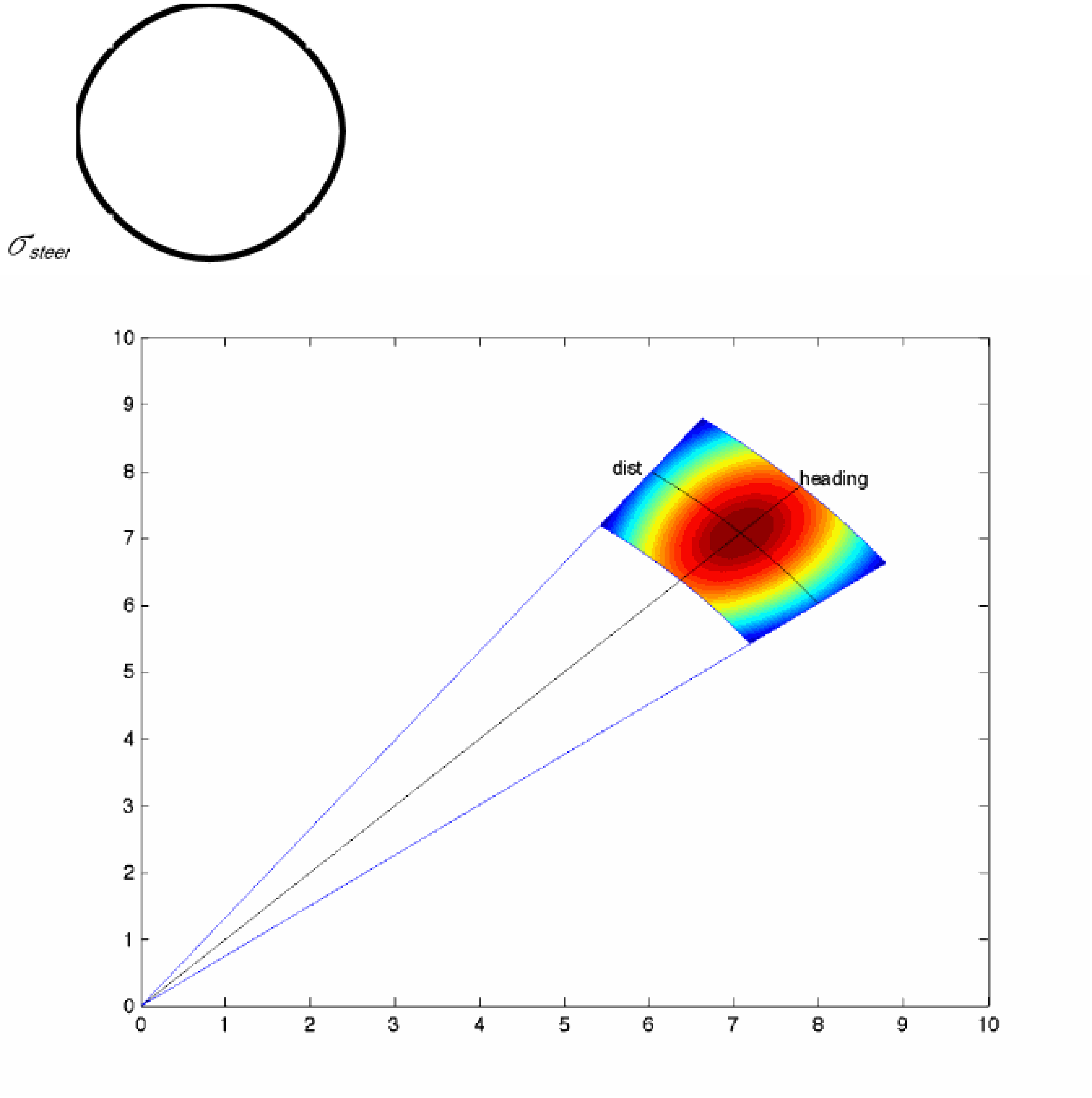


Fig. 4. The conservative analytical distribution representing dead reckoning error
Generated through Matlab and Microsoft Word (the actual circle)

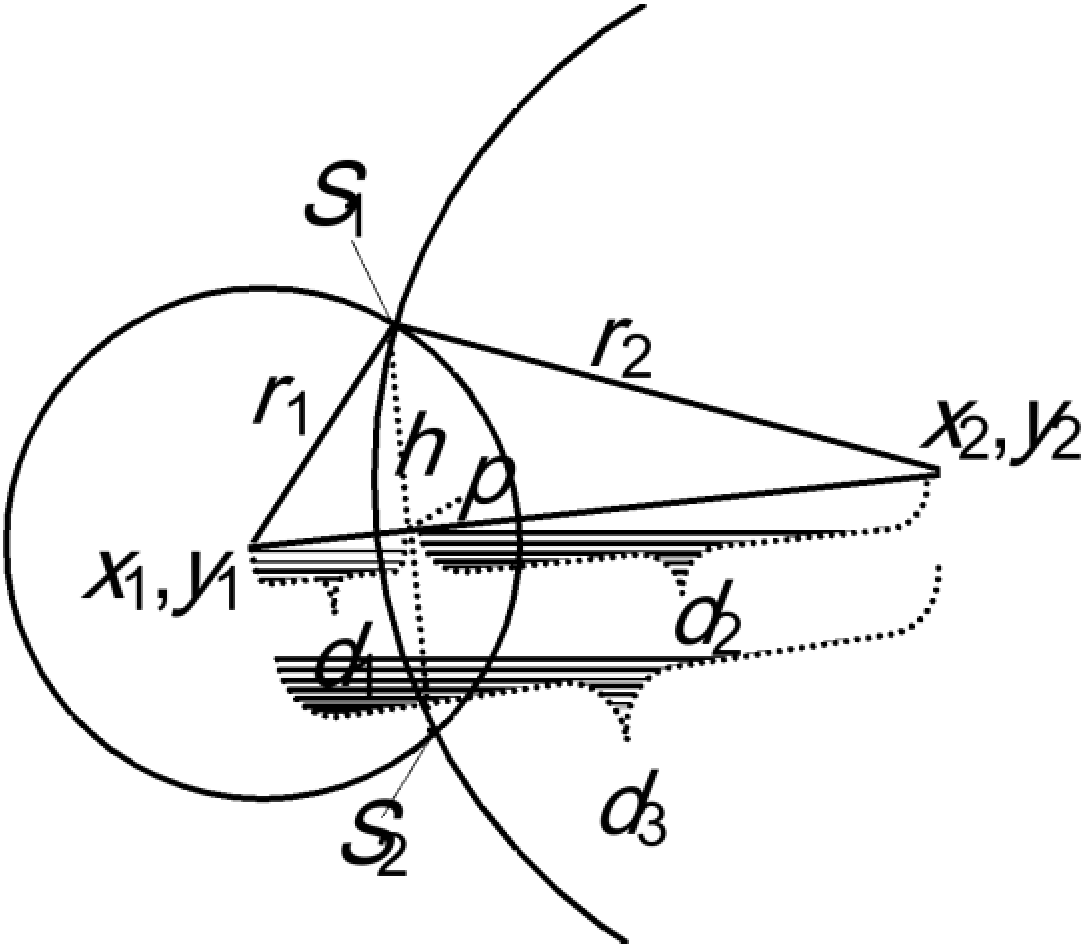


Fig. 5. Determining the intersection point of two beacons.

(Generated through Microsoft Word)

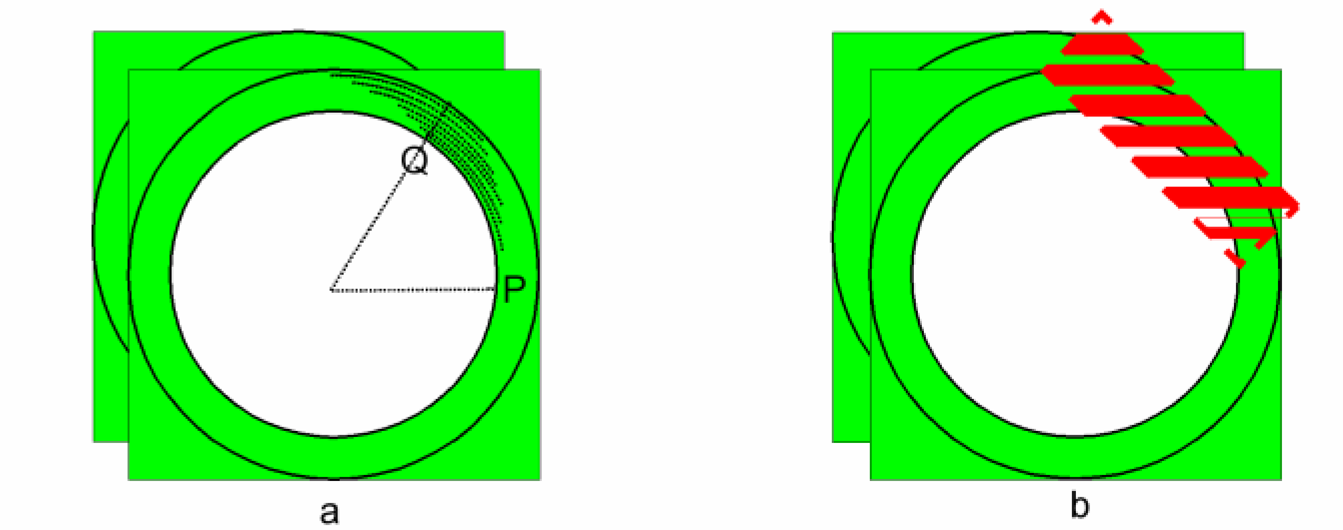


Fig. 6. (a) Sweeping the intersection region and (b) determining a bounding rectangle. (Generated through Microsoft Word)

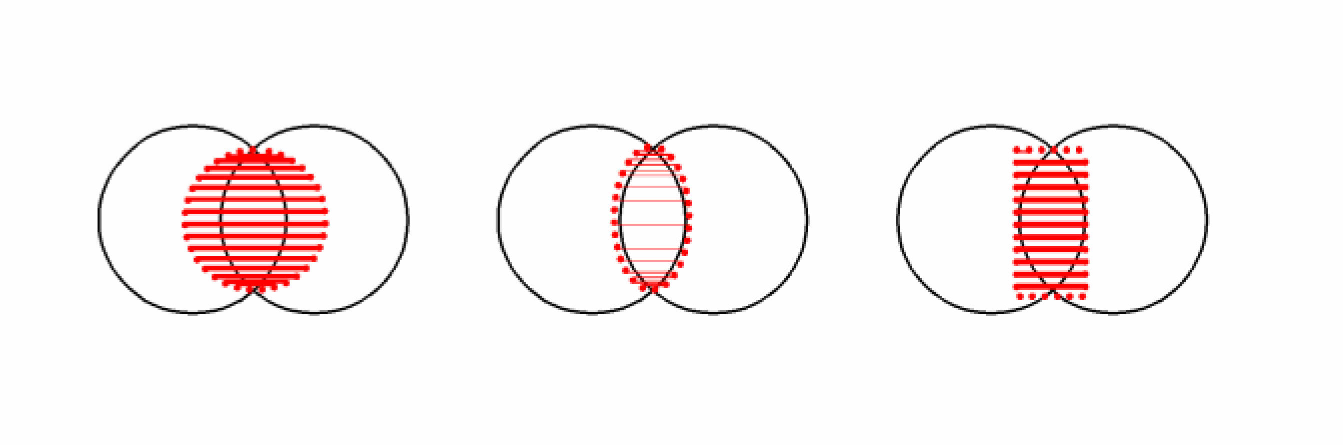


Fig. 7. Three choices of bounding shape.

(Generated through Microsoft Word)

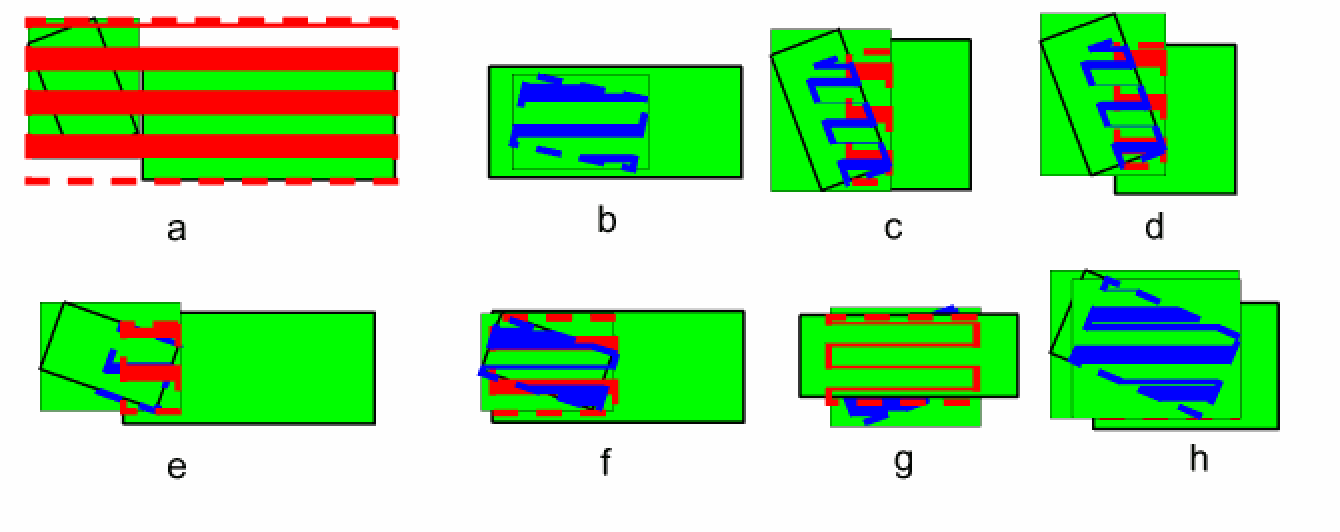


Fig. 8. Different cases for intersecting rectangles used in algorithm IntersectRegion
(Generated through Microsoft Word)

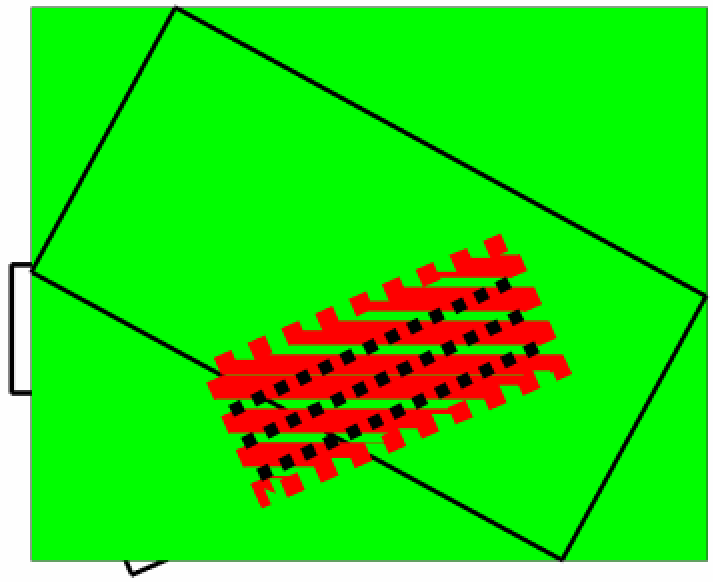


Fig. 9. Once the smallest rectangle is calculated, probabilities must be computed on a grid (Generated through Microsoft Word)

q *p*

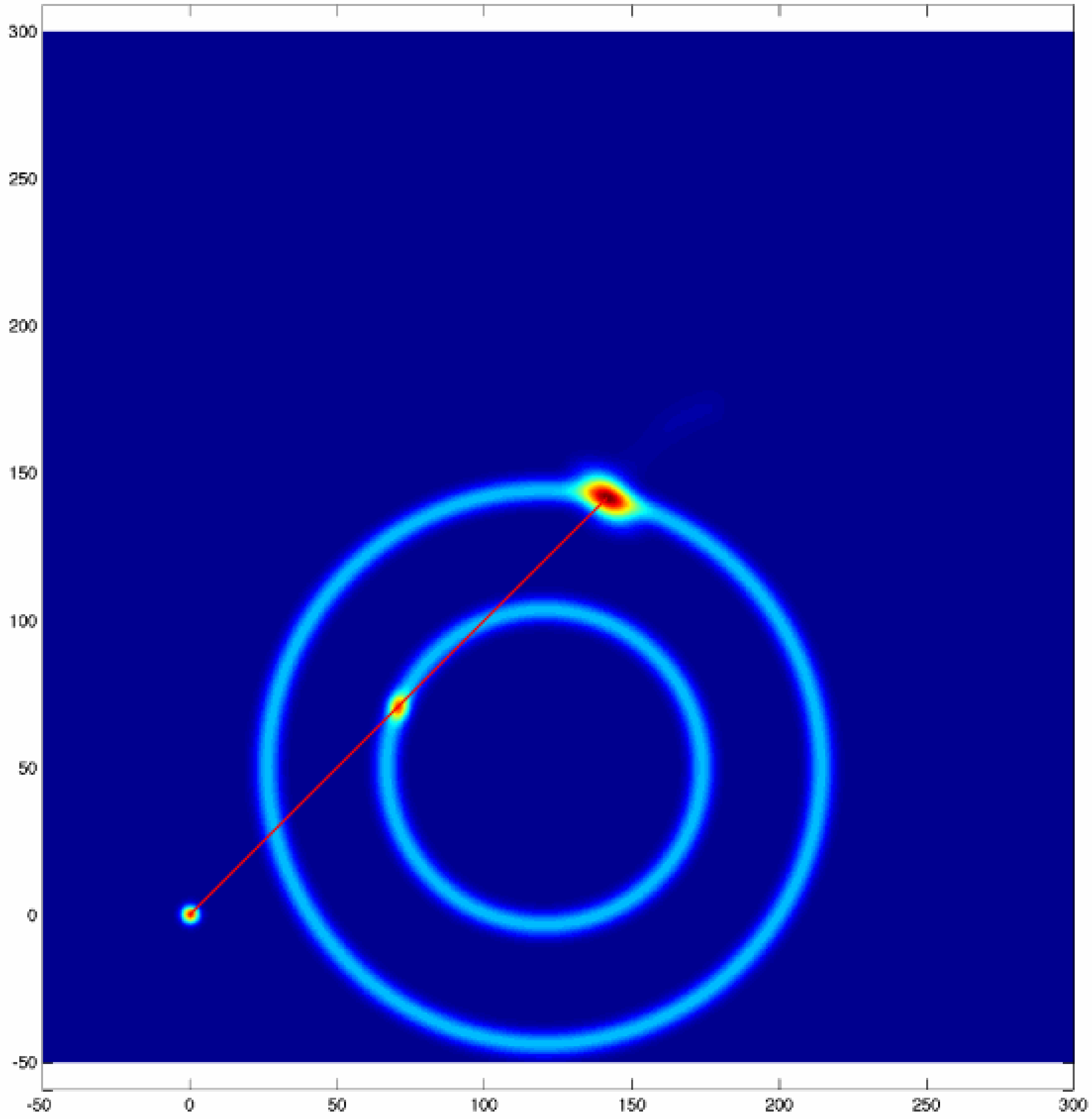


Fig. 10. Sample case, navigating with two beacon pings 30 seconds apart. Red line shows DR track

(Generated through Matlab and denoted through Microsoft Word)

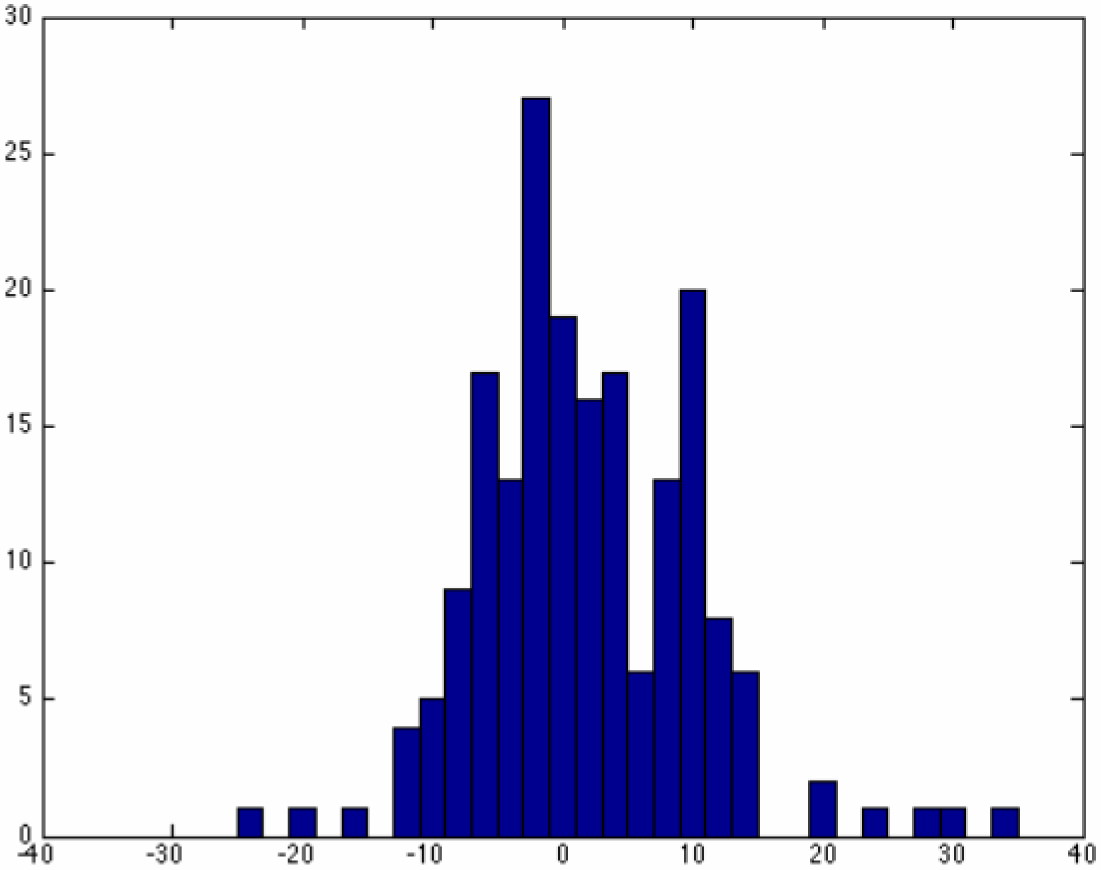


Fig. 11 (a)

(Generated through Matlab)

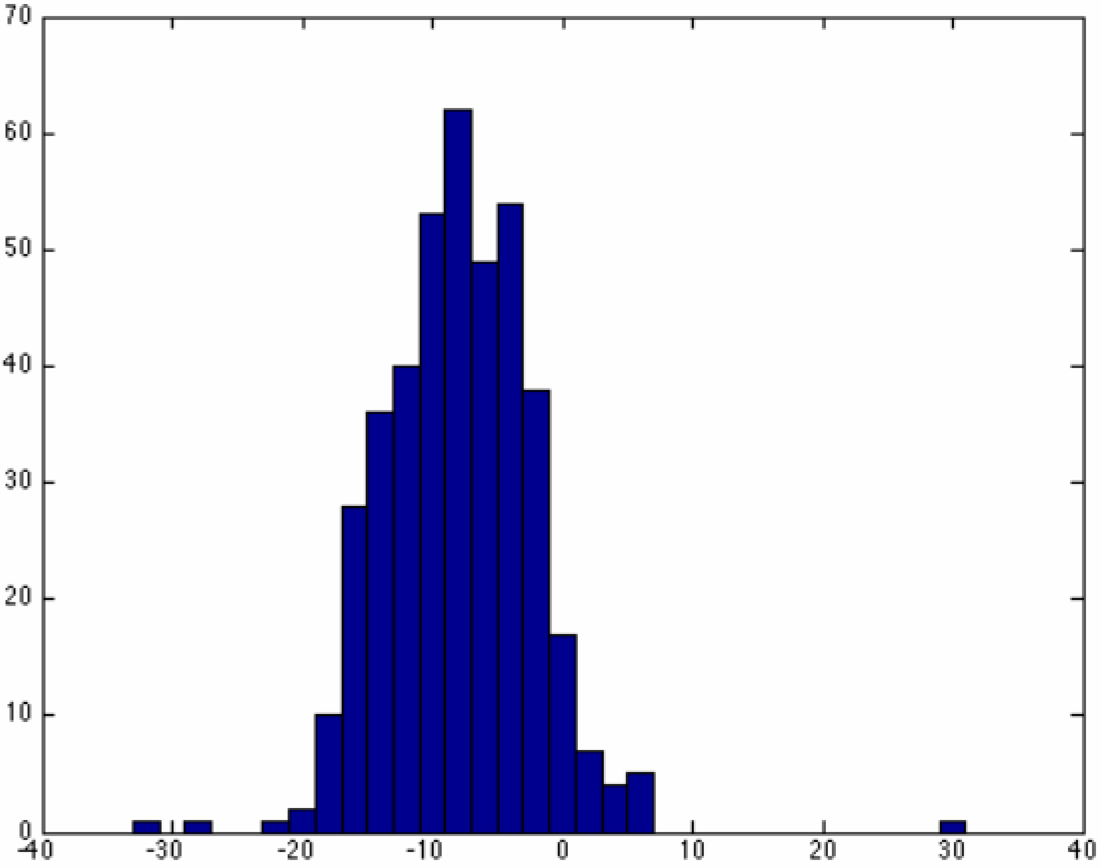


*Fig. 11 (*b)

*Fig. 11. Histogram showing the error of distance estimation from L*BL vs. GPS based on data obtained from (a) the UUV operation experiment, and (b) the boat experiment.

(Generated through Matlab)

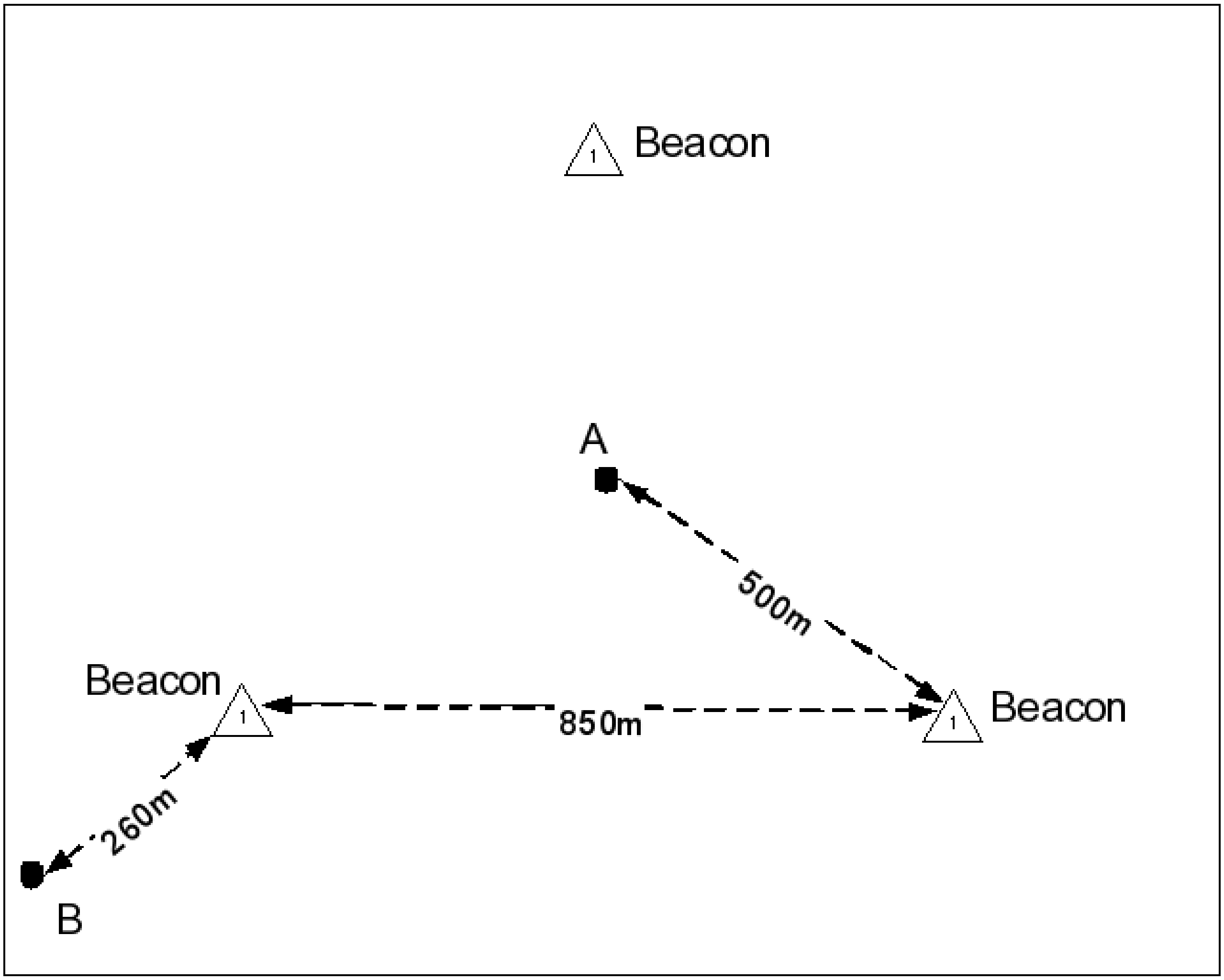


Fig. 12 Simulation setup to compare MADF with particle filter using 200, 500, and 1000 particles. Two locations *A* and *B* are tested with 1000 trials.

(Generated through Microsoft Visio)

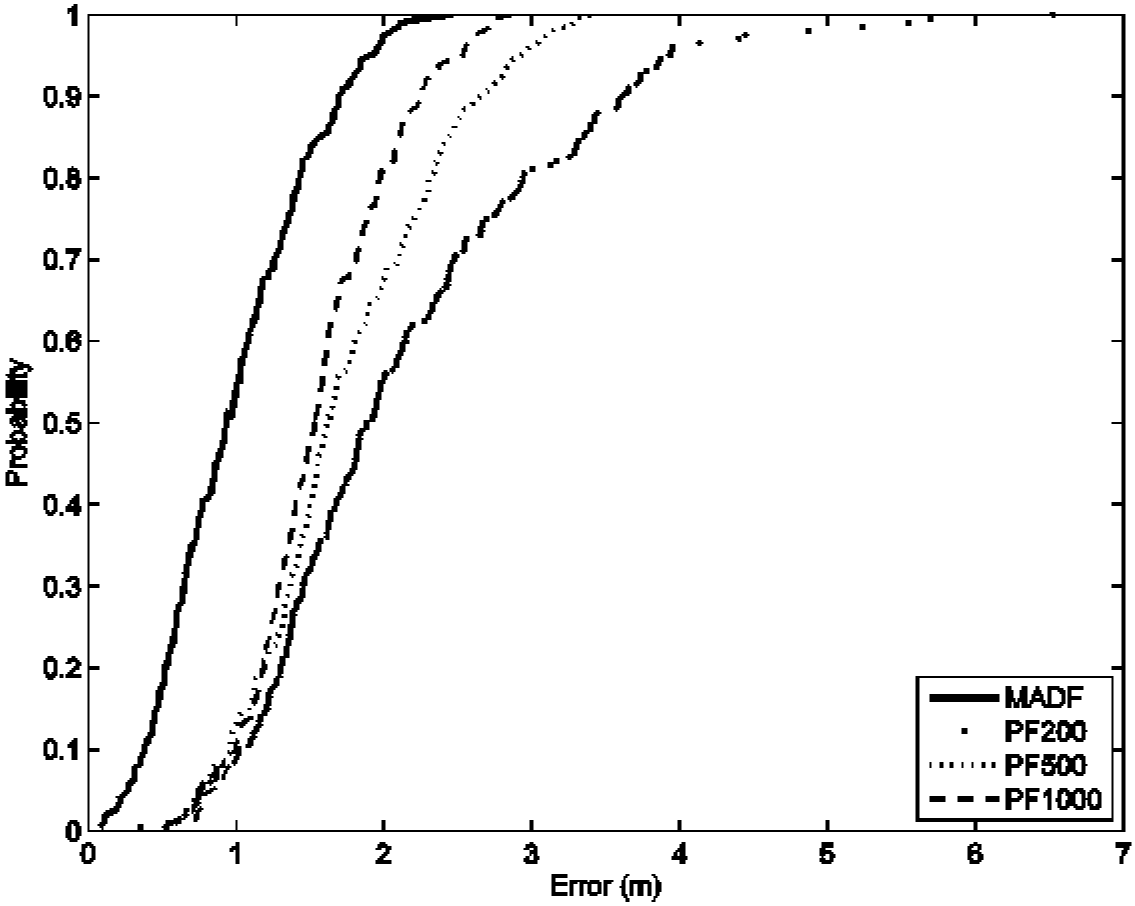


Fig. 13(a)

(Generated through Matlab)

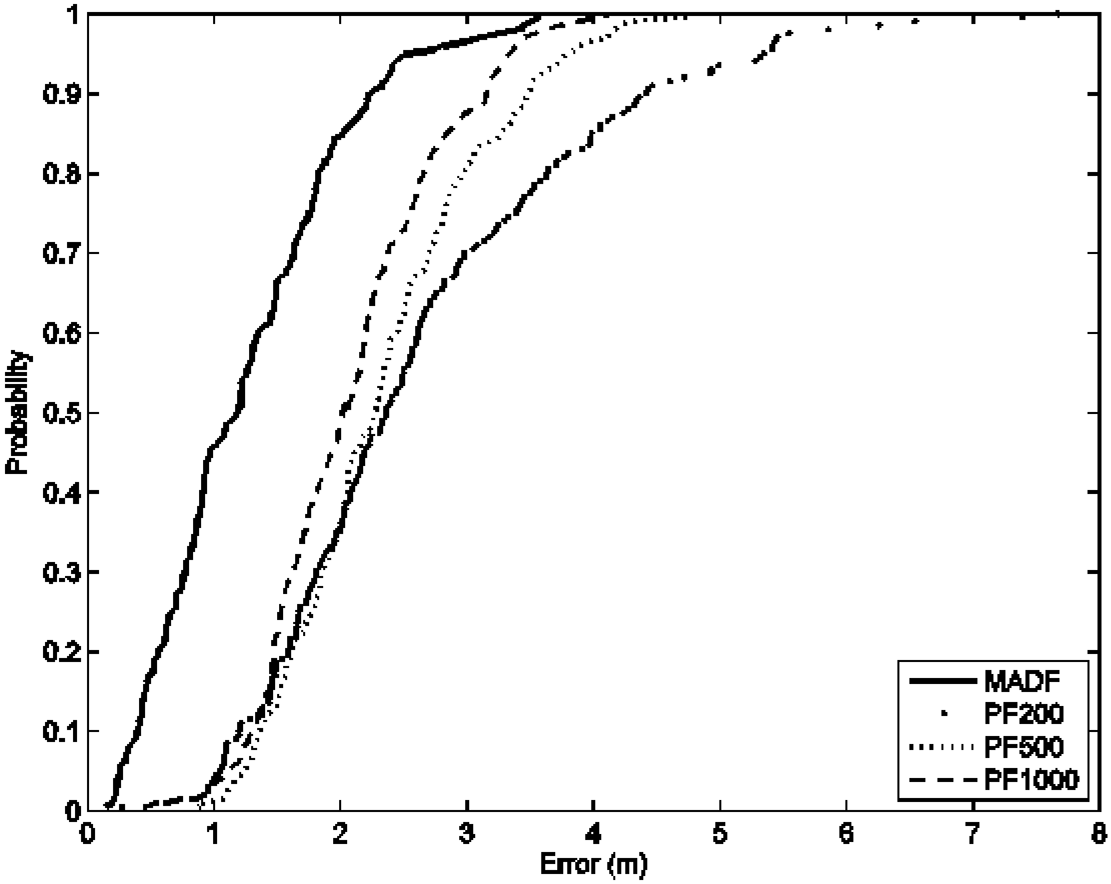


Fig. 13(b)
Fig. 13 CDF of simulated errors from MADF (4´4) and particle filters with 200, 500, 1000 particles at (a) location A and (b) location B.

(Generated through Matlab)

LIST OF TABLE LEGENDS

Table 1. Comparison of Processing Time for 10,000 Iterations of Different Algorithms

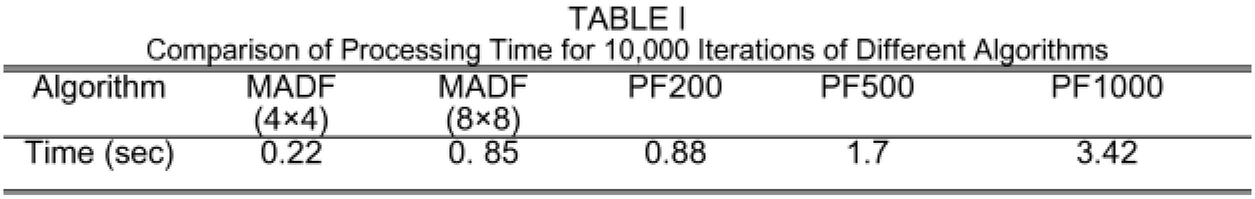


TABLE I
Comparison of Processing Time for 10,000 Iterations of Different Algorithms

Algorithm	MADF (4×4)	MADF (8×8)	PF200	PF500	PF1000
Time (sec)	0.22	0. 85	0.88	1.7	3.42

[1] Center for Maritime Systems, Stevens Institute of Technology, Hoboken, NJ 07030 USA. (Corresponding e-mail: dkruger@stevens.edu)

[2] Center for Decision Technologies, Stevens Institute of Technology, Hoboken, NJ 07030 USA