COMPARISON OF LOCALIZATION ALGORITHMS USING ATTENUATION ESTIMATES

Aishwarya Moni, Scott Rickard

Submitted to EUSIPCO 2008
Complex & Adaptive Systems Laboratory
University College Dublin
Ireland

ABSTRACT

In this paper, algorithms for locating sources using attenuation estimates are discussed. These algorithms assume that the sensor locations are known. Six closed form methods and an iterative method are described. A spatial histogram algorithm is introduced. The computational complexities of all the algorithms are discussed and the accuracies of the algorithms are compared as a function of error in the sensor locations. The iterative method based on GPS [3] is discussed and a spatial histogram method is shown to be the most accurate algorithm at the expense of higher computational load, and is the best technique to use for far field sources.

1. INTRODUCTION

The problem of locating sources has received considerable attention over the past few decades as it is very relevant in many areas such as networking, geology, meteorology and navigation. Consider the case where there are \( N \) sensors \( m_1, m_2, \ldots, m_N \). The signal received at each of the sensors is

\[
x_i(t) = \alpha_i s(t - \delta_i) \tag{1}
\]

where \( x_i(t) \) is the signal mixture produced at the \( i^{th} \) sensor at time \( t \), \( s(t) \) is the source signal at time \( t \), \( \delta_i \) is the propagation delay between the source and the \( i^{th} \) sensor and \( \alpha_i \) is the attenuation in amplitude at the \( i^{th} \) sensor. Our goal is to determine the source location given the sensor signals when slightly inaccurate sensor locations are known. We discuss approaches to this problem which first estimate the intra-sensor attenuation estimates, and then from these estimates, calculate the source location. The focus here is on the many techniques of source location estimation from intra-source attenuation estimates, and not on the process of attenuation estimation itself. Petre Stoica et al describe four closed form techniques for locating sources using time delay estimates in [1]. All the algorithms described in the paper can be modified for locating sources using attenuation estimates. [2] gives an overview of five least square techniques for locating sources using attenuation estimates. In this paper, an additional iterative method based on GPS [3] is discussed and a spatial histogram method is introduced. Also, the accuracy and robustness of all eight localization methods are compared for various source-sensor location scenarios as a function of errors in the sensor locations. Based on simulation results, recommendations are made as to which technique should be used, depending on the whether the source is near or far field.

The attenuation coefficient of the signal between each pair of sensors, \( a_{ij} = \frac{a_i}{a_j} \), can be estimated in many ways. In the time domain, this can be done by calculating the time delay of arrival using cross correlation, shifting one of the signals so the two signals align, and estimating the ratio of the two signals. In the time-frequency domain, the equivalent mixing model is

\[
X_i(\tau, \omega) = a_i e^{-j \omega \delta} S(\tau, \omega) \tag{2}
\]

where \( X_i(\tau, \omega) \) and \( S(\tau, \omega) \) are the time-frequency representations of \( x_i(t) \) and \( s(t) \) respectively. The attenuation coefficient can be estimated by taking the ratio

\[
R_{ij}(\tau, \omega) = \frac{X_i(\tau, \omega)}{X_j(\tau, \omega)} = a_{ij} e^{-j \delta_{ij} \omega}
\]

and the attenuation coefficient estimate at each time-frequency point is then

\[
a_{ij}(\tau, \omega) = |R_{ij}(\tau, \omega)|.
\]

These time-frequency attenuation coefficient estimates can then be averaged to produce a single delay estimate [4].

Assuming that the source is at \( x = (x, y, z) \) and sensor locations are \( m_i = (x_i, y_i, z_i) \), the following equations must be satisfied for each pair of sensors

\[
\frac{d_i}{d_j} = a_{ij} \tag{3}
\]

where \( d_i = \sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2} \). Equation (3) is the equation of a hyper-sphere, on which the source must lie. In the case of \( N \) sensors, \( K = \binom{N}{2} \) such hyper-spheres are found and can be intersected to find the location of \( x \).

\[
a_{ij} = \frac{d_i}{d_j} = \frac{\sqrt{(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2}}{\sqrt{(x-x_j)^2 + (y-y_j)^2 + (z-z_j)^2}}
\]

and the resulting equation is

\[
(x - \frac{x_i - a_{ij}^2 x_j}{1 - a_{ij}^2})^2 + (y - \frac{y_i - a_{ij}^2 y_j}{1 - a_{ij}^2})^2 + (z - \frac{z_i - a_{ij}^2 z_j}{1 - a_{ij}^2})^2
\]

\[
= \frac{a_{ij}^2}{(1 - a_{ij}^2)} ((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2). \tag{4}
\]

A few closed form form solutions have been proposed over the...
years for localization using time delay estimates. 2 describes the various algorithms which have been modified to localize sources using attenuation estimates instead. [5] solves Equation (4) (described in 2.1). [6] find the least squares solution to Equation (4) (described in 2.2). [2] uses the fact that two spheres intersect to form a circle, and intersects the planes that the circles lie on to get a location (described in 2.4). [7] uses the result from [6] and adds a correction step using Lagrangian multipliers (described in 2.5). [1] provides an overview of some of the techniques discussed here and introduces a new approximate least squares technique (described in 2.3).

The consideration of the iterative technique based on GPS [3] and the introduction of the spatial histogram technique are contributions of the paper. Also, via simulation results, we provide guidance to which technique should be used depending on the relative position of source and sensors.

2. LOCALIZATION METHODS

The localization algorithms are briefly discussed.

2.1 Spherical Intersection

This method is described in [5]. The equations of the spheres resulting from Equation (4) are

\[(x-x_{ci})^2 + (y-y_{ci})^2 + (z-z_{ci})^2 = R_i^2\]  

(5)

for \(i = 1,\ldots,K\). Expanding Equation (5), the following equation results

\[x_i^2 + y_i^2 + z_i^2 - R_i^2 + D_s^2 = 2x_i x + 2y_i y + 2z_i z\]  

(6)

where \(D_s^2 = x^2 + y^2 + z^2\). In matrix form, for all \(i\), Equation (6) can be written as

\[
\begin{align*}
\Delta + D_s^2 1_K &= 2Mx \\
\begin{bmatrix}
D_1^2 - R_1^2 \\
D_2^2 - R_2^2 \\
\vdots \\
D_K^2 - R_K^2
\end{bmatrix}
&=
\begin{bmatrix}
x_{c1} & y_{c1} & z_{c1} \\
x_{c2} & y_{c2} & z_{c2} \\
\vdots & \vdots & \vdots \\
x_{cK} & y_{cK} & z_{cK}
\end{bmatrix}
\end{align*}
\]

where \(\Delta = (I - M)^{-1}M^T\), \(M = (I - M)^{-1}M^T\), and \(\Delta + D_s^2 1_K\) is a vector of ones and \(x = \begin{bmatrix} x \\ y \\ z \end{bmatrix}\).

From Equation (7), \(x\) in terms of \(D_s\) is

\[
x = \frac{1}{2}M^T(\Delta + D_s^2 1_K) \]

(8)

where \(M^+ = (M^T M)^{-1}M^T\), is the Moore-Penrose pseudo inverse of \(M\) when \(K \neq 3\). So if the distance of the source to the origin \(D_s\) was known, \(x\) can be calculated. It is known that

\[D_s = (x^T x)^{1/2}\]  

(9)

Substituting Equation (8) into Equation (9) and squaring both sides yields

\[D_s^2 = \frac{1}{4}((\Delta^T + D_s^2 1_K^T)(M^+)(\Delta + D_s^2 1_K))\]  

(10)

The result is a quadratic equation in \(D_s^2\)

\[aD_s^4 + bD_s^2 + c = 0\]  

(11)

where

\[
a = (1^T (M^+)^T M^+ 1_K) \\
b = (1^T (M^+)^T M^+ \Delta + \Delta^T (M^+)^T M^+ 1_K - 4) \\
c = \Delta^T (M^+)^T M^+ \Delta
\]

The solution for \(D_s^2\) is the following

\[D_s^2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}; D_s \geq 0\]  

(12)

This solution to \(D_s^2\) can be substituted into Equation (8) to get the intersection point \(x\). This method of solving for the intersection point results in two possible answers for \(x\), which are then substituted back into Equation (5) to choose the right \(x\).

2.2 Spherical Interpolation

This technique is described in [6]. Since the measurement of amplitude may not be very accurate, Equation (5) may not be correct. The error function is defined as below

\[
e_i = (x-x_{ci})^2 + (y-y_{ci})^2 + (z-z_{ci})^2 - R_i^2 \\
= x_i^2 + y_i^2 + z_i^2 - R_i^2 + D_s^2 - 2x_i x - 2y_i y - 2z_i z \\
\epsilon = \Delta + D_s^2 1_K - 2Mx
\]

(13)

where \(\Delta\) and \(M\), and \(x\) are the same as before. Substituting Equation (8) into Equation (13) yields the following.

\[
\epsilon = \Delta + D_s^2 1_K - M(M^T M)^{-1}M^T(\Delta + D_s^2 1_K)
\]

(14)

where \(P_m = M(M^T M)^{-1}M^T\) and \(P_m^+ = I - P_m\). This method will only work when the number of spheres \(K\) is greater than 3. In the \(K = 3\) case, \(P_m = I\) and the equation error \(\epsilon\) is zero for all \(D_s\). In the more general case of \(K\) sensors, Equation (14) holds so that

\[
\epsilon^T \epsilon = (\Delta + D_s^2 1_K)^T P_m^+ P_m^+ (\Delta + D_s^2 1_K)
\]

(15)

and differentiating with respect to \(D_s\) yields

\[
\frac{\delta \epsilon^T \epsilon}{\delta D_s} = 4D_s 1_K^T P_m^+ \Delta + 4D_s^3 1_K^T P_m^+ 1_K^T
\]

(16)

Equating \(\frac{\delta \epsilon^T \epsilon}{\delta D_s} = 0\) gives either \(D_s = 0\) which we discard or the following

\[D_s^2 = -I_K^T P_m^+ \Delta (I_K^T P_m^+ 1_K)^{-1}\]  

(17)

Substituting Equation (17) into Equation (8) gives a unique solution for \(x\).
2.3 Subspace Minimization

This algorithm is introduced in [8]. Equation (13) is hard to solve because of the unknown quantity $D_2$. One way to eliminate $D_2$ is to project the solution into a subspace that is perpendicular to the vector $1_k$. This can be achieved by multiplying every term in Equation (13) by a projection matrix $P_u^\perp := I - \frac{1_k 1_k^T}{1_k^T 1_k}$. Since $P_u^\perp 1_k = 0$, Equation (13) becomes

$$P_u^\perp M x = P_u^\perp \Delta + P_u^\perp \epsilon$$

(18)

and the least squares solution of Equation (18) can be estimated as

$$x = (P_u^\perp M)^+ \Delta.$$  

(19)

2.4 Quadratic Elimination

This method is introduced in [2] and is based on the fact that two spheres intersect to form a circle, and the planes that the circles lie on are intersected to find the solution. Using Equation (5), and subtracting the $j^{th}$ sphere from the $i^{th}$ sphere, the following equation of a plane results

$$2(x_i - x_j)x + 2(y_i - y_j)y + 2(z_i - z_j)z$$

$$= x_i^2 - x_j^2 + y_i^2 - y_j^2 + z_i^2 - z_j^2 - R_i^2 + R_j^2$$

(20)

and in matrix format, for all pairs of spheres, Equation (20) can be rewritten as

$$P x = \hat{\Delta}$$

(21)

where

$$P = \begin{pmatrix}
2(x_1 - x_2) & 2(y_1 - y_2) & 2(z_1 - z_2) \\
2(x_2 - x_3) & 2(y_2 - y_3) & 2(z_2 - z_3) \\
\vdots & \vdots & \vdots \\
2(x_{i-1} - x_i) & 2(y_{i-1} - y_i) & 2(z_{i-1} - z_i) \\
D_1^2 - R_1^2 & D_2^2 - R_2^2 & D_3^2 - R_3^2
\end{pmatrix}$$

and $L = \begin{pmatrix} K \\ 2 \end{pmatrix}$.

The least square solution to Equation (21) is

$$x = P^+ \hat{\Delta}.$$

(22)

2.5 Linear Correction using Lagrangian multipliers

This algorithm is described in [7]. Equation (13) can be rewritten as

$$\epsilon = \hat{M} \hat{x} - \Delta$$

(23)

where

$$\hat{M} = \begin{pmatrix}
2x_1 & 2y_1 & 2z_1 & -1 \\
\vdots & \vdots & \vdots & \vdots \\
2x_i & 2y_i & 2z_i & -1 \\
2x_K & 2y_K & 2z_K & -1
\end{pmatrix}$$

and $\hat{x} = \begin{pmatrix} x \\ y \\ z \\ D_1^2 \end{pmatrix}$

$\epsilon^T \epsilon$ has to be minimized constrained to:

$$\hat{x}^T \Sigma \hat{x} = 2\hat{x}^T \pi$$

(24)

where $\Sigma = \begin{pmatrix} 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}$ and $\pi = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} 1/2)$. To solve the constrained optimization problem, the Lagrange multiplier method can be used:

$$L(\hat{x}, \lambda) = (\hat{M} \hat{x} - \Delta)^T (\hat{M} \hat{x} - \Delta) - \lambda (\hat{x}^T \Sigma \hat{x} - 2\hat{x}^T \pi)$$

$$= \hat{x}^T (\hat{M}^T \hat{M} + \lambda \Sigma) \hat{x} - \hat{x}^T (2\hat{M}^T \Delta + 2\lambda \pi) + \Delta^T \Delta$$

(25)

and differentiating Equation (25) yields

$$\nabla_\hat{x} L(\hat{x}, \lambda) = 2(\hat{M}^T \hat{M} + \lambda \Sigma) \hat{x} - 2(2\hat{M}^T \Delta + 2\lambda \pi)$$

$$\implies (\hat{M}^T \hat{M} + \lambda \Sigma) \hat{x} = 2\hat{M}^T \Delta + 2\lambda \pi$$

(26)

when $\nabla_\hat{x} L(\hat{x}, \lambda) = 0$. Equation (26) can be substituted into Equation (24) to solve for $\lambda$ explicitly, and then $\hat{x}$ can be solved. Let the initial estimate of $\hat{x}$ be denoted by $\hat{x}_1$ such that

$$\hat{x}_1 = \hat{x} + \delta \hat{x}.$$  

(27)

Substituting for $\hat{x}_1$ into Equation (26) yields

$$\lambda(\Sigma \hat{x}_1 - \pi) = \hat{M} \Delta - \hat{M}^T \hat{M} \hat{x}_1$$

(28)

Let $\hat{x}_1 = \hat{M}^+ \Delta$ which is the least squares estimate to $\hat{M} \hat{x}_1 = \Delta$. Then the linear correction equation is

$$\delta \hat{x} = (I - \lambda_0 (\hat{M}^T \hat{M})^{-1}) \Sigma \hat{x}_1 + \lambda_0 (\hat{M}^T \hat{M})^{-1} \pi$$

(29)

where $\lambda_0 = (\hat{M}^T \hat{M})^{-1} \Sigma \hat{x}_1$.  

2.6 Approximate Least Squares Estimates

This technique is introduced in [1]. From Equation (23) error to minimize is now

$$\epsilon = \Delta - \hat{M} \hat{x}$$

(30)

and the least squares criterion can be rewritten as (to within an additive constant)

$$\epsilon^T \epsilon = (\hat{x} - \hat{x}_1)^T \hat{M}^T \hat{M} (\hat{x} - \hat{x}_1)$$

(31)

where $\hat{x}_1 = \hat{M}^+ \Delta = [x_1, y_1, z_1, D_1^2]^T$. $\hat{x} - \hat{x}_1$ is linearized near $\hat{x}_1$ by means of a Taylor series expansion

$$\hat{x} - \hat{x}_1 \approx \hat{x} - \hat{x}_1 + \frac{\delta \hat{x}}{\delta \hat{x}^T} |_{\hat{x} = \hat{x}_1} (\hat{x} - \hat{x}_1)$$

$$= \delta + G(\hat{x} - \hat{x}_1)$$

(32)

where

$$\delta = \begin{pmatrix}
D_1^2 - x_1 \\
D_2^2 - y_1 \\
D_3^2 - z_1 \\
D_4^2 - D_5^2
\end{pmatrix}$$

$$G = \begin{pmatrix}
2x_1 & 2y_1 & 2z_1 & 0 \\
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{pmatrix}$$

and

$$\hat{x}_1 = \begin{pmatrix}
x_1 \\
y_1 \\
z_1 \\
D_1^2
\end{pmatrix}.$$  

Substituting Equation (32) in Equation (31) yields

$$\epsilon^T \epsilon = (\delta + G(\hat{x} - \hat{x}_1))^T (\hat{M}^T \hat{M}) (\delta + G(\hat{x} - \hat{x}_1))$$

(33)

Minimizing the above equation results in the following least mean squares solution for $\hat{x}$

$$\hat{x} = \hat{x}_1 - (\hat{G}^T \hat{M}^T \hat{M})^{-1} (\hat{G}^T \hat{M}^T \hat{M} \delta).$$

(34)
2.7 Iterative method GPS

This method is based on the algorithm described in [3]. For each pair of sensors, the equation of the resulting hypersphere is

\[ R_i = \sqrt{(x - x_{ci})^2 + (y - y_{ci})^2 + (z - z_{ci})^2}. \]  

(35)

Allowing for noise, Equation (35) can be modified to

\[ R_i = \sqrt{(x - x_{ci})^2 + (y - y_{ci})^2 + (z - z_{ci})^2} + b \]  

(36)

where \( b \) is the noise. In Equation (36), there are four variables, \( x, y, z \) and \( b \). Differentiating Equation (36) yields the following equation:

\[ \delta R_i = \frac{(x - x_{ci})\delta x + (y - y_{ci})\delta y + (z - z_{ci})\delta z}{\sqrt{(x - x_{ci})^2 + (y - y_{ci})^2 + (z - z_{ci})^2}} + \delta b. \]  

(37)

\( K \) such equations exist. In matrix form, Equation (37) can be written as the following.

\[ \hat{d} = \alpha \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta b \end{pmatrix} \]  

(38)

where

\[ \alpha = \begin{pmatrix} (x_{ci} - y_{ci}) & (y_{ci} - z_{ci}) & (z_{ci} - x_{ci}) & 1 \\ (x_{ci} - y_{ci}) & (y_{ci} - z_{ci}) & (z_{ci} - x_{ci}) & 1 \\ (x_{ci} - y_{ci}) & (y_{ci} - z_{ci}) & (z_{ci} - x_{ci}) & 1 \\ (x_{ci} - y_{ci}) & (y_{ci} - z_{ci}) & (z_{ci} - x_{ci}) & 1 \end{pmatrix} \]

and \( \hat{d} = \begin{pmatrix} \delta R_1 \\ \delta R_2 \\ \delta R_3 \\ \delta R_K \end{pmatrix} \).

Equation (38) can be solved iteratively. The first step is to guess initial values for \( x = x_{g0}, y = y_{g0}, z = z_{g0} \) and \( b = b_{g0} \). Using these values, \( R_i \) can be calculated using Equation (36). The difference between these \( R_i \) and the \( R_i \) calculated from attenuation estimates is \( \delta R_i \). Now the only unknowns in the equation are \( \delta x, \delta y, \delta z \) and \( \delta b \). Using the least squares method to minimize the difference between the two sides in Equation (38), the following update results:

\[ \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta b \end{pmatrix} = \alpha^+ \hat{d}. \]  

(39)

Values for \( x, y, z \) and \( b \) can be updated using the following equation.

\[ \begin{pmatrix} x_{g1} \\ y_{g1} \\ z_{g1} \\ b_{g1} \end{pmatrix} = \begin{pmatrix} x_{g0} \\ y_{g0} \\ z_{g0} \\ b_{g0} \end{pmatrix} + \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta b \end{pmatrix}. \]  

(40)

Using these new values, new \( R_i \) and thus new \( \delta R_i \) can be found and the steps repeated. The iterations are continued till the error, \( \delta y = \sqrt{\delta x^2 + \delta y^2 + \delta z^2 + \delta b^2} \) reaches a specified threshold.

2.8 Spatial Histogram

The new method that is introduced in this paper is the spatial histogram algorithm. For each point in a given lattice, the likelihood of the point being the source is calculated as below for each pair of sensors.

\[ P_i(x) = \exp(-\frac{1}{W}(||x - x_{ci}|| - R_i^2)) \]  

(41)

where \( ||x - x_{ci}|| = \sqrt{(x - x_{ci})^2 + (y - y_{ci})^2 + (z - z_{ci})^2} \), \( i = 1, \ldots, K \) and \( W \) is an arbitrary weight. For each lattice point, the values in Equation (41) are multiplied for all pairs.

\[ P(x) = \prod_{i=1}^{K} P_i(x). \]  

(42)

The lattice point that gives maximum \( P(x) \) is the maximum likelihood estimate of the source location.

3. COMPUTATIONAL COMPLEXITY

The computational complexity of each algorithm will be discussed. The time taken for each algorithm is taken into account as well as the computational effort needed for each algorithm. Figure 1 shows how the time taken for each algorithm increases as the number of sensors increase. The implementation of the algorithm is not optimized, the graph is there to give a guide to the practical complexity of the algorithms as a function of the number of sensors. The size of the matrix to be inverted in each algorithm is given in Table 1, where \( niter \) is the number of iterations, \( K = \left(\frac{N}{2}\right) \) and \( N \) is the number of sensors. Essentially there are four groups of the complexities of the algorithms:

- The spherical intersection, spherical interpolation, subspace minimization linear correction and approximate least square techniques are all very quick as the operations involve finding the pseudo inverse of a \( K \)-by-3 or \( K \)-by-4 matrix. Thus the complexity is \( O(N^2) \).
- The quadratic elimination technique involves finding the pseudo inverse of a \( \left(\frac{K}{2}\right) \)-by-3 matrix and thus the complexity is \( O(N^4) \).
- The iterative method based on GPS is slow, and this is
<table>
<thead>
<tr>
<th>Localization Algorithm</th>
<th>Size of Matrix to be Inverted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spherical Intersection</td>
<td>$K$-by-$3$</td>
</tr>
<tr>
<td>Spherical Interpolation</td>
<td>$K$-by-$3$</td>
</tr>
<tr>
<td>Subspace Minimization</td>
<td>$K$-by-$3$</td>
</tr>
<tr>
<td>Quadratic Elimination</td>
<td>$(K-2)$-by-$3$</td>
</tr>
<tr>
<td>Linear Correction</td>
<td>$(K-3)$-by-$3$ and $2 \times (4-3)$-by-$4$</td>
</tr>
<tr>
<td>Approximate Least Squares</td>
<td>$(K-3)$ and $(3-3)$-by-$3$</td>
</tr>
<tr>
<td>GPS</td>
<td>$niter \times (K-4)$</td>
</tr>
<tr>
<td>Spatial Histogram</td>
<td>none</td>
</tr>
</tbody>
</table>

Table 1: Computational Complexities of Algorithms.

Figure 2: Error in output vs. error in input - source outside 'cube' of sensors (far-field).

Figure 3: Error in output vs. error in input - source inside 'cube' of sensors (near-field).

due to the size of the matrix to be inverted increasing and the number of iterations that are being run. At each iteration the pseudo inverse of a $K$-by-$4$ matrix is found. Thus the complexity is $O(niter N^2)$.

- The spatial histogram method is the slowest from a practical standpoint because the distance to $K$ spheres are calculated for each lattice point and the number of lattice points is very large. There is a clear tradeoff between accuracy and speed for this method. The complexity is $O(\Gamma N^2)$ where $\Gamma$ depends on the lattice resolution and size.

4. COMPARISON OF METHODS

In the simulations carried out, the sensors were placed at the vertices of a cube. The source was moved along a diagonal of the 'cube', extending to outside the 'cube', to test which method works best for each location of the source. Two source locations have been chosen, one inside the 'cube' which has typical results for a near field source and one outside the 'cube' which has typical results for a far field source. For each of these source locations, different levels of noise were added to the known locations of sensors and the correct attenuation values were input to the different algorithms to test how robust each algorithm was to error in the measured position of the sensors. The algorithms were run to localize the source. The error in source position was calculated (the distance from estimated source position to the correct source position). Figure 2 shows the percentage output error vs. percentage input error for a typical far field source. Figure 3 shows the percentage output error vs. percentage input error for a typical near field source. The spatial histogram method proved to be the most robust algorithm for both near field and far field sources and was the best technique for far field sources. The iterative method based on GPS achieved the best for near field sources but wasn’t robust to sensor location error for far field sources. The closed form solutions gave similar results, again achieving better results for near field sources. The approximate least squares method, however gave poor results for both near field and far field sources with about 2000% error in estimated source location for very low sensor location errors and is not visible in Figure 2 and Figure 3.

5. CONCLUSION

Six closed form solutions and an iterative technique were described and a spatial histogram technique was introduced for the localization of a source using relative attenuation and sensor locations. The computational complexities of the techniques were also discussed. The results of the simulations show that the errors are lower in the near field case for all the techniques discussed compared to the far field case. The iterative technique provides the best results for near field sources but is more computationally intensive than the closed form algorithms. The spatial histogram method, which has the highest computational load, is robust to sensor location error for both near field and far field sources and is the best technique for far field sources.

REFERENCES