Nonlinear Least Squares 3-D Geolocation Solutions using Time Differences of Arrival

Michael V. Bredemann

University of New Mexico

Follow this and additional works at: https://digitalrepository.unm.edu/math_etds

Part of the Aeronautical Vehicles Commons, Applied Mathematics Commons, Computational Engineering Commons, Computer and Systems Architecture Commons, Digital Circuits Commons, Digital Communications and Networking Commons, Electrical and Electronics Commons, Environmental Monitoring Commons, Geophysics and Seismology Commons, Hardware Systems Commons, Mathematics Commons, Military Vehicles Commons, Numerical Analysis and Scientific Computing Commons, Space Vehicles Commons, Statistics and Probability Commons, Systems and Communications Commons, Systems Architecture Commons, Systems Engineering and Multidisciplinary Design Optimization Commons, and the Theory and Algorithms Commons

Recommended Citation

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at UNM Digital Repository. It has been accepted for inclusion in Mathematics & Statistics ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact amywinter@unm.edu, Isloane@salud.unm.edu, sarahrk@unm.edu.
Michael Van Bredemann

Candidate

Mathematics and Statistics

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Professor Jens Lorenz, Chairman

Professor Mohammad Motamed

Professor Jehanzeb Chaudhry
Nonlinear Least Squares 3-D Geolocation Solutions using Time Differences of Arrival

by

Michael Van Bredemann

B.S. E.E., University of Missouri-Rolla, Dec. 1980
B.S. Applied Mathematics, University of Missouri-Rolla, May 1981
M.S. E.E., University of New Mexico-Albuquerque, May 1986
Ph.D. E.E., University of New Mexico-Albuquerque, May 1995

THESIS

Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science
in
Mathematics

The University of New Mexico
Albuquerque, New Mexico

May 2020
Acknowledgements

To Professors Pedro Embid, Jens Lorenz and Mohammad Motamed, thanks for your excellent technical instruction.
Nonlinear Least Squares 3-D Geolocation
Solutions using Time Differences of Arrival

by
Michael Van Bredemann

B.S. E.E., University of Missouri-Rolla, Dec. 1980
B.S. Applied Mathematics, University of Missouri-Rolla, May 1981
M.S. E.E., University of New Mexico, May 1986
Ph.D. E.E., University of New Mexico, May 1995
M.S. Mathematics, University of New Mexico, May 2020

ABSTRACT

This thesis uses a geometric approach to derive and solve nonlinear least squares minimization problems to geolocate a signal source in three dimensions using time differences of arrival at multiple sensor locations. There is no restriction on the maximum number of sensors used. Residual errors reach the numerical limits of machine precision. Symmetric sensor orientations are found that prevent closed form solutions of source locations lying within the null space. Maximum uncertainties in relative sensor positions and time difference of arrivals, required to locate a source within a maximum specified error, are found from these results. Examples illustrate potential requirements specification applications. The maximum machine epsilon and the maximum number of iterations to reach the least squares solution without loss of source location accuracy are estimated. Improvements in accuracy of least squares solutions over closed form solutions are measured.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xiii</td>
</tr>
<tr>
<td>Notation</td>
<td>xiv</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Applications</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Background</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Contributions</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Organization</td>
<td>7</td>
</tr>
<tr>
<td>2 Preliminaries</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Dot Product</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Directional Cosines</td>
<td>11</td>
</tr>
<tr>
<td>2.3 Matrix Rotations</td>
<td>13</td>
</tr>
<tr>
<td>2.4 Hyperbolic Nature of TDOAs</td>
<td>14</td>
</tr>
</tbody>
</table>
4.2 General Sensor Perturbations ................................... 79
4.3 TDOA Perturbations ............................................. 80
4.4 Requirements Specification Examples ........................... 81
  4.4.1 Example 1 ..................................................... 82
  4.4.2 Example 2 ..................................................... 87

5 Summary ............................................................. 93
  5.1 Results .......................................................... 93
    5.1.1 Geometric Approach ....................................... 94
    5.1.2 Sensor Position Anomaly .................................. 94
    5.1.3 Dilution of Precision Metric ............................... 94
    5.1.4 Modifications to LM Algorithm ........................... 95
    5.1.5 TDOA Computation Improvements ......................... 95
    5.1.6 Optimal Convergence Times ............................... 96
    5.1.7 Maximum Machine Epsilon .............................. 96
    5.1.8 Frames of Reference Models ............................. 96
    5.1.9 Fixed Reference Frame Results ........................ 97
    5.1.10 Floating Reference Frame Results ..................... 97
    5.1.11 TDOA Perturbation Results ............................ 98
    5.1.12 Sensor Position & TDOA Perturbation Results .......... 98
  5.2 Future Research ............................................... 99
## List of Figures

<table>
<thead>
<tr>
<th>Chapter 2</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1 Dot Product in ( \mathbb{R}^2 )</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2.2 Dot Product of Unit Vectors in ( \mathbb{R}^2 )</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2.3 Directional Cosines</td>
<td>12</td>
</tr>
<tr>
<td>Figure 2.4 Points in ( \mathbb{R}^2 ) for ( \Delta d_{12} = -6 )</td>
<td>16</td>
</tr>
<tr>
<td>Figure 2.5 Points in ( \mathbb{R}^2 ) for ( \Delta d_{12} = +6 )</td>
<td>17</td>
</tr>
<tr>
<td>Figure 2.6 Points in ( \mathbb{R}^3 ) for ( \Delta d_{12} = -6 )</td>
<td>18</td>
</tr>
<tr>
<td>Figure 2.7 Points in ( \mathbb{R}^3 ) for ( \Delta d_{12} = +6 )</td>
<td>19</td>
</tr>
<tr>
<td>Figure 2.8 Constant TDOA Hyperbola in ( \mathbb{R}^2 )</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.9 Constant TDOA Hyperboloid in ( \mathbb{R}^3 )</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.10 IEEE 754 Double Precision Floating Point Format</td>
<td>29</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 3</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3.1 Source Directional Cosine Vectors</td>
<td>44</td>
</tr>
<tr>
<td>Figure 3.2 Unperturbed Sensor Positions</td>
<td>47</td>
</tr>
</tbody>
</table>
Figure 3.3  Non-Reference Frame Relative Sensor Perturbation . . . . . . . 49
Figure 3.4  General Sensor Perturbation Amplitude . . . . . . . . . . . . . 52
Figure 3.5  Sensor Perturbation Directional Cosine Vectors for $P_1$ . . . . . . 54
Figure 3.6  Unperturbed TDOA Computation for $P_{t1}$ and $P_{tk}$ . . . . . . . 57
Figure 3.7  Least Squares Formulation for $P_1$ and $P_k$ . . . . . . . . . . . 65
Figure 3.8  Sensor Position Symmetry Nulls . . . . . . . . . . . . . . . . . 71

Appendix A  

Figure A.1  Relative Sensor Perturbation LSSPE, B=1e0 . . . . . . . . . . . 103
Figure A.2  Relative Sensor Perturbation LSSPE, B=1e1 . . . . . . . . . . . 104
Figure A.3  Relative Sensor Perturbation LSSPE, B=1e2 . . . . . . . . . . . 105
Figure A.4  Relative Sensor Perturbation LSSPE, B=1e3 . . . . . . . . . . . 106
Figure A.5  Relative Sensor Perturbation LSSPE, B=1e4 . . . . . . . . . . . 107
Figure A.6  Relative Sensor Perturbation LSSPE, B=1e5 . . . . . . . . . . . 108
Figure A.7  Relative Sensor Perturbation LSSPE, B=1e6 . . . . . . . . . . . 109
Figure A.8  Relative Sensor Perturbation LSSPE, B=1e7 . . . . . . . . . . . 110
Figure A.9  Relative Sensor Perturbation LSSPE, B=1e8 . . . . . . . . . . . 111

Appendix B  

Figure B.1  General Sensor Perturbation LSSPE, B=1e0 . . . . . . . . . . . 113
Figure B.2  General Sensor Perturbation LSSPE, B=1e1 . . . . . . . . . . . 114
Figure B.3  General Sensor Perturbation LSSPE, B=1e2 . . . . . . . . . . . 115
Figure B.4  General Sensor Perturbation LSSPE, B=1e3 . . . . . . . . . . . 116

x
Figure B.5 General Sensor Perturbation LSSPE, B=1e4 .......... 117
Figure B.6 General Sensor Perturbation LSSPE, B=1e5 .......... 118
Figure B.7 General Sensor Perturbation LSSPE, B=1e6 .......... 119
Figure B.8 General Sensor Perturbation LSSPE, B=1e7 .......... 120
Figure B.9 General Sensor Perturbation LSSPE, B=1e8 .......... 121

Appendix C

Figure C.1 TDOA Perturbation LSSPE, B=1e0 .......... 123
Figure C.2 TDOA Perturbation LSSPE, B=1e1 .......... 124
Figure C.3 TDOA Perturbation LSSPE, B=1e2 .......... 125
Figure C.4 TDOA Perturbation LSSPE, B=1e3 .......... 126
Figure C.5 TDOA Perturbation LSSPE, B=1e4 .......... 127
Figure C.6 TDOA Perturbation LSSPE, B=1e5 .......... 128
Figure C.7 TDOA Perturbation LSSPE, B=1e6 .......... 129
Figure C.8 TDOA Perturbation LSSPE, B=1e7 .......... 130
Figure C.9 TDOA Perturbation LSSPE, B=1e8 .......... 131

Appendix D

Figure D.1 Example 1 Max LSSPE .......... 133
Figure D.2 Example 1 Max LSSPE Improvement .......... 134
Figure D.3 Example 1 Max Iterations to Convergence .......... 135
Figure D.4 Example 2 Max LSSPE .......... 136
Figure D.5 Example 2 Max LSSPE Improvement .......... 137
Figure D.6 Example 2 Max Iterations to Convergence . . . . . . . . . . . . . 138
List of Tables

Chapter 2

Table 2.1 IEEE 754 Standard Floating Point Binary Formats . . . . . . . . 29

Chapter 4

Table 4.1 LSSPEs ($\epsilon$ vs $R$) for Example 1 Sensor Perturbations . . . . . . . . . . . 84
Table 4.2 LSSPEs ($\epsilon$ vs $R$) for Example 1 TDOA Perturbations . . . . . . 84
Table 4.3 LSSPEs ($\epsilon$ vs $R$) for Example 1 Sensor & TDOA Perturbations . 85
Table 4.4 LSSPEs ($\epsilon$ vs $R$) for Example 2 General Sensor Perturbations . . . 89
Table 4.5 LSSPEs ($\epsilon$ vs $R$) for Example 2 TDOA Perturbations . . . . . . . . 90
Table 4.6 LSSPEs ($\epsilon$ vs $R$) for Example 2 Sensor & TDOA Perturbations . 91
Notation

\[ \Re(f) = \frac{(f + f^*)}{2} = \text{real part of complex valued function } f \]

\[ f^* = \text{complex conjugate of complex valued function } f \]

\[ \vec{f}^* = \text{complex conjugate of complex valued vector } \vec{f} \]

\[ \vec{f}^T = \text{transpose of vector } \vec{f} \]

\[ \left[ \vec{f}^* \right]^T = \left[ \vec{f}^T \right]^* = \text{complex conjugate transpose of complex valued vector } \vec{f} \]

\[ |\vec{f}| = \sqrt{\left( \vec{f}^T \right)^* \vec{f}} = \text{the Euclidean norm of complex valued vector } \vec{f} \]

rand[a : b] = a random number from a uniform distribution from a to b

\[ \alpha = \text{angle of arrival at the origin with respect to the x-axis} \]

\[ \beta = \text{angle of arrival at the origin with respect to the y-axis} \]

\[ \gamma = \text{angle of arrival at the origin with respect to the z-axis} \]

\[ \theta = \text{angle of arrival at the midpoint of two sensors with respect to the line between them} \]

\[ \phi = \text{the angle of rotation around the x-axis} \]

\[ \psi = \text{the angle of rotation around the y-axis} \]

\[ \zeta = \text{the angle of rotation around the z-axis} \]
\[ R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \]

= the rotation matrix around the x-axis

\[ R_y(\psi) = \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{bmatrix} \]

= the rotation matrix around the y-axis

\[ R_z(\zeta) = \begin{bmatrix} \cos(\zeta) & -\sin(\zeta) & 0 \\ \sin(\zeta) & \cos(\zeta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

= the rotation matrix around the z-axis

\[ A^* = \text{complex conjugate of complex valued matrix } A \]

\[ A^T = \text{transpose of matrix } A \]

\[ [A^*]^T = [A^T]^* = \text{complex conjugate transpose of complex valued matrix } A \]

\[ \kappa(A) = \frac{\sigma_{\text{max}}}{\sigma_{\text{min}}} = \text{condition number of matrix } A \]

\[ \sigma_{\text{max}} = \max_{\vec{v} \neq \vec{0}} \frac{|A\vec{v}|}{|\vec{v}|} = \text{maximum singular value of matrix } A \]

\[ \sigma_{\text{min}} = \min_{\vec{v} \neq \vec{0}} \frac{|A\vec{v}|}{|\vec{v}|} = \text{minimum singular value of matrix } A \]

\[ \vec{S} = \text{a potential source location vector} \]

\[ f(\vec{S}) = \text{a nonlinear complex valued function of vector } \vec{S} \]
\[ \hat{f}(\vec{S}) = \left[ f_1(\vec{S}), f_2(\vec{S}), \cdots, f_N(\vec{S}) \right]^T \] is a vector of \( N \) nonlinear complex valued functions of vector \( \vec{S} \)

\[ F_c(\vec{S}) = \frac{1}{2} \sum_{i=1}^{N} \left[ f_i(\vec{S}) f_i^*(\vec{S}) \right] \] is a real valued cost function of \( N \) nonlinear complex valued functions of \( \vec{S} \)

\[ \hat{\nabla}_S f(\vec{S}) = 1\text{st derivative of function } f \text{ with respect to vector } \vec{S} \]

\[ \hat{\nabla}_S^2 f(\vec{S}) = 2\text{nd derivative of function } f \text{ with respect to vector } \vec{S} \]

\[
J = \begin{bmatrix}
\frac{\partial f_1(\vec{S})}{\partial x} & \frac{\partial f_1(\vec{S})}{\partial y} & \frac{\partial f_1(\vec{S})}{\partial z} \\
\frac{\partial f_2(\vec{S})}{\partial x} & \frac{\partial f_2(\vec{S})}{\partial y} & \frac{\partial f_2(\vec{S})}{\partial z} \\
\vdots & \vdots & \vdots \\
\frac{\partial f_N(\vec{S})}{\partial x} & \frac{\partial f_N(\vec{S})}{\partial y} & \frac{\partial f_N(\vec{S})}{\partial z}
\end{bmatrix}
\] = the Jacobian matrix of \( \hat{f}(\vec{S}) \)

\[
H_i = \begin{bmatrix}
\frac{\partial^2 f_i(\vec{S})}{\partial x^2} & \frac{\partial^2 f_i(\vec{S})}{\partial x \partial y} & \frac{\partial^2 f_i(\vec{S})}{\partial x \partial z} \\
\frac{\partial^2 f_i(\vec{S})}{\partial y \partial x} & \frac{\partial^2 f_i(\vec{S})}{\partial y^2} & \frac{\partial^2 f_i(\vec{S})}{\partial y \partial z} \\
\frac{\partial^2 f_i(\vec{S})}{\partial z \partial x} & \frac{\partial^2 f_i(\vec{S})}{\partial z \partial y} & \frac{\partial^2 f_i(\vec{S})}{\partial z^2}
\end{bmatrix}
\] = the Hessian matrix of \( f_i(\vec{S}) \)

\( N_S = \) the number of sensors

\( N_P = \) the number of sensor pairs

\( \epsilon = \) the convergence threshold

\( \text{eps} = \) the machine epsilon

\( \mu \lambda = \) the gradient gain in the Levenberg-Marquardt algorithm
\( \mu \) = the damping parameter and limited gradient gain factor

\( \lambda \) = the unlimited gradient gain factor

\( \hat{\vec{P}}_1 \) = the vector pointing to the unperturbed center sensor

\( \hat{\vec{P}}_1 = [0, 0, 0]^T \)

\( \hat{\vec{P}}_k \) = the vector pointing to the unperturbed \( k \)-th sensor

\( \hat{\vec{P}}_{tk} \) = the vector pointing to the perturbed \( k \)-th sensor

\( B \) = the unperturbed distance between the center sensor and other sensors

\[ B = \left| \hat{\vec{P}}_1 - \hat{\vec{P}}_k \right| = \sqrt{(\hat{\vec{P}}_1 - \hat{\vec{P}}_k)^T (\hat{\vec{P}}_1 - \hat{\vec{P}}_k)}, k \neq 1 \]

\( B \in \{1e0, \ldots, 1e8\} \) meters

\( \xi \) = the relative sensor perturbation angle \( \in \{0, 1e-15, \ldots, 1e-1\} \) radians

\( \xi \in \{0, 1e-15, \ldots, 1e-1\} \) radians

\( \eta \) = the random relative sensor rotation angle

\( \eta \in [0 : 2\pi] \) radians

\( \varrho \) = the magnitude of the general sensor position perturbation

\[ \varrho = B \times \xi / 2 \] meters

\( \hat{\varphi}_k \) = the \( k \)-th sensor position directional cosine general perturbation vector

\( \hat{\vec{p}}_k \) = the \( k \)-th sensor position general perturbation vector

\[ \hat{\vec{p}}_k = \varrho \times \hat{\varphi}_k \]

\( D_{ik} \) = the distance between sensors \( i \) and \( k \)
\[ D_{ik} = |\vec{P}_i - \vec{P}_k| = \sqrt{\left(\vec{P}_i - \vec{P}_k\right)^T \left(\vec{P}_i - \vec{P}_k\right)}, \] or
\[ D_{ik} = |\vec{P}_{ti} - \vec{P}_{tk}| = \sqrt{\left(\vec{P}_{ti} - \vec{P}_{tk}\right)^T \left(\vec{P}_{ti} - \vec{P}_{tk}\right)}, \] depending on context

\[ \vec{M}_{ik} = \text{the vector pointing to the midpoint of sensors } i \text{ and } k \]
\[ \vec{M}_{ik} = \left(\vec{P}_i + \vec{P}_k\right)/2, \] or
\[ \vec{M}_{ik} = \left(\vec{P}_{ti} + \vec{P}_{tk}\right)/2, \] depending on context

\[ R_{ik} = \text{the distance between the source and the midpoint of sensors } i \text{ and } k \]

\[ R_{ik} = |\vec{R}_s - \vec{M}_{ik}| = \sqrt{\left(\vec{R}_s - \vec{M}_{ik}\right)^T \left(\vec{R}_s - \vec{M}_{ik}\right)}, \] or
\[ R_{ik} = |\vec{S} - \vec{M}_{ik}| = \sqrt{\left(\vec{S} - \vec{M}_{ik}\right)^T \left(\vec{S} - \vec{M}_{ik}\right)}, \] depending on context

\[ R = \text{the true source distance from the reference frame origin} \]

\[ R \in \{1\text{e-}2, 1\text{e-}1, \ldots, 1\text{e}21\} \text{ meters} \]

\[ \vec{r}_s = \text{the true source directional cosine vector from the reference frame origin} \]

\[ \vec{R}_s = \text{the true source position vector from the reference frame origin} \]

\[ d_k = \text{the distance between the source position and } k\text{-th sensor position} \]

\[ d_k = |\vec{R}_s - \vec{P}_{tk}| = \sqrt{\left(\vec{R}_s - \vec{P}_{tk}\right)^T \left(\vec{R}_s - \vec{P}_{tk}\right)}, \] or
\[ d_k = |\vec{S} - \vec{P}_k| = \sqrt{\left(\vec{S} - \vec{P}_k\right)^T \left(\vec{S} - \vec{P}_k\right)}, \] depending on context

\[ d_{ik} = d_i - d_k \]

\[ \Delta t_{ik} = \text{the time difference of arrival between sensors } i \text{ and } k \]

\[ \Delta t_{ik} = d_{ik}/v, \text{ where } v \text{ is the signal speed} \]

\[ v = c = 299,792,458 \text{ m/s} = \text{the speed of light used in these simulations} \]
Chapter 1

Introduction

The three dimensional (3D) geolocation problem addressed in this thesis is defined as follows:

Given a maximum uncertainty in $k$ known stationary sensor positions, $P_1, P_2, \ldots, P_k$, and in the corresponding time difference of arrivals (TDOAs) at each sensor pair of a signal emitted from an unknown stationary source location, how accurately can the signal source position be determined? Or conversely, what maximum uncertainties in known sensor positions and TDOA measurements are required to geolocate a source within a prescribed maximum error?

Numerical results characterize the effects of sensor position and TDOA uncertainties on 3D geolocation accuracy. The results apply to stationary sources and sensors with direct line of sight signal paths and are independent of phenomenology. Adjustments for the effects of source and sensor motion, variations in the signal transmission medium, reflections in multi-path environments, and the feasibility of the maximum
uncertainties are beyond the scope of this work.

1.1 Applications

There are many applications for TDOA geolocations, including the location of a sniper or explosion in the acoustic environment, an underground tremor or detonation in the seismic environment, an underwater mammal or submersible in the sonar environment, and a lightning strike or 911 cell phone call in the electromagnetic environment, among others. The numerical approach presented here can be applied to each using the appropriate signal speed. The application targeted is the geolocation of an optical or radio frequency (RF) source using multiple sensors in a satellite system.

The assumptions of a stationary source and sensors and constant velocity direct line of sight signal paths are impractical for this application. They are included to help characterize maximum location errors for various parameters effecting the solution. A similar approach may characterize other parameters that alter the assumptions.

1.2 Background

There is an extensive amount of prior work on the topic of source location. The references provided are not exhaustive, but were used or considered in this application. Methods historically used to locate a signal source are commonly referred to as
time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), and received signal strength (RSS), or combinations thereof. [Li et al., 2016] summarizes a recent survey of the different techniques. When the source or sensors are in relative motion, frequency difference of arrival (FDOA) is sometimes used in conjunction with TDOA for improved geolocation results or for source tracking.

Closed form solutions for source locations offer the least computational complexity for the TDOA problem. It is well established that four sensors are required to unambiguously locate a source using only TDOA information in two dimensions (2D). Similarly, five sensors are required to unambiguously locate a source in 3D. Additional information and special cases allow a source to be located with fewer sensors. See [Holle and Lopez, 1993] for a 2D pictorial example of the 3 sensor ambiguity.

3D closed form solutions have been found using 4, 5 and 6 sensors. [Potluri, 2002] and [Mellen et al., 2003] derived 4-sensor solutions, which require additional information to resolve the square root dual solution ambiguity. [Bakhoum, 2006] appears to be the first publication of an unambiguous 5-sensor TDOA geolocation. Bakhoum converted the problem to the solution of a system of 3 linear equations by first eliminating the initial time of signal transmission. [Spencer, 2010] derived closed form solutions for both 4 and 5 sensors in spherical coordinates, with the 4-sensor solution retaining the square root ambiguity. [Dong et al., 2014] reduced the 5-sensor geolocation to a solution of 4 linear equations, explicitly including the initial time of signal transmission. Using 6-sensors, [Li and Dong, 2014] added an unknown constant ve-
locity variable in the solution. Although simple to compute, closed form solutions do not exploit additional information offered in an over constrained sensor set.

Averages are sometimes used, when an over constrained solution set exists, as in [Abel and Smith, 1987]. It can be argued that the average produces better accuracies over least squares results, but this depends on the probability distribution of the errors. [Harter, 1974a] and [Harter, 1974b] describe historical alternatives to the method of least squares considered in the theory of errors. Outliers can be omitted to improve accuracies in all methods.

The method of least squares is commonly used and the approach taken here. One goal in this work is to extract system requirements for the maximum uncertainty of multiple parameters to satisfy a desired or defined maximum source location error. To reach this goal, the maximum of all least squares minimizations are found over random perturbations of a maximum error in one or more parameters. The expected value of the location error can be found from assumptions on the error probability distribution of random uncertainties and is already well studied and documented. The errors found here represent the absolute worst case.

For linear problems, the method of least squares forms a convex hull. For nonlinear problems, as in this case, the nonlinear squared error is linearized about a point. The hope is to find a starting point close enough to a local minimum that corresponds to the desired solution. Descent methods are typically used to reach the local minimum. For a suitable starting point, the time to and rate of convergence depend on the
descent method.

[Gustafsson and Gunnarsson, 2003] used the stochastic gradient, a version of the conjugate gradient, and a particle filter method in 2D simulations. As reported in [Frandsen et al., 2004], the steepest descent and conjugate gradient methods have a linear convergence rate and can be slow, but are stable, and Newton type methods are quadratically convergent, but susceptible to instability when far from the minimum.

The algorithm chosen here is a damped Newton type of descent combining the stable steepest descent with the faster Newton type method. A version of the Levenberg-Marquardt algorithm, see [Levenberg, 1944] and [Marquardt, 1963], is used with the Gauss-Newton estimate of the second derivative (eliminating the Hessian), with the intent to minimize computation time while maintaining sufficient accuracy to meet the goals. See [Madsen et al., 2004] and their references for Gauss-Newton approximations. One of the above mentioned or other methods may find the solution from any starting point or improve the time to convergence for all parameter variations.

1.3 Contributions

The contributions of this thesis are as follows.

1. A new geometric approach is used to formulate a nonlinear least squares minimization problem to geolocate a signal source in three dimensions using time differences of arrival.

2. A symmetric 5-sensor position anomaly is revealed that prevents a closed form
solution for a signal source position lying within the null space using only known
sensor positions and TDOAs between sensor pairs.

3. The condition number of Bakhoum’s 5-sensor closed form solution 3x3 matrix
is used as a new TDOA dilution of precision metric to extract an optimal,
non-anomalous, 5-sensor set, from which the initial starting point is computed.

4. The Levenberg-Marquardt algorithm is modified to solve nonlinear least squares
minimizations with residual errors reaching the limits of machine precision.

5. An alternate TDOA computation for larger source distances improve the TDOA
accuracy by up to 8 orders of magnitude for double precision floating point
numbers allowing residual errors to reach the limits of machine precision.

6. Optimal thresholds are determined to minimize the number of steps to reach
convergence.

7. The maximum machine epsilon is estimated to minimize power and resource
consumption in hardware implementations.

8. Models are developed for sensor position errors using reference frames that are
either fixed or floating with respect to a sensor.

9. Graphs of simulation results provide an easy estimate of the maximum sen-
sor and TDOA perturbations required to geolocate a signal source within a
prescribed maximum error for the given sensor positions.
10. Examples illustrate procedures for fixed reference frame and floating reference frame system engineering requirements specifications or feasibility assessments.

1.4 Organization

This thesis is organized as follows:

- Chapter 2 identifies existing mathematical tools used with some minor alterations;

- Chapter 3 presents the sensor and TDOA error models, TDOA computation improvements, least squares formulation, disclosure of the symmetric sensor anomaly, use of the new dilution of precision metric to avoid the anomaly, scaling parameters, convergence thresholds and estimated machine epsilons;

- Chapter 4 applies the models of Chapter 3 to simulations and describes a procedure illustrated with examples to extract system engineering requirements or feasibility assessments for a desired geolocation accuracy;

- Chapter 5 summarizes the results achieved in this thesis and offers suggestions for future research;

- Appendix A contains graphs of maximum geolocation errors for various levels of maximum uncertainty in relative sensor positions with unperturbed TDOAs, in which the reference frame is attached to the middle sensor;
• Appendix B contains graphs of maximum geolocation errors for various levels of maximum uncertainty in general sensor positions with unperturbed TDOAs, in which the reference frame is not attached to any sensor;

• Appendix C contains graphs of maximum geolocation errors for various levels of maximum uncertainty in TDOA measurements with unperturbed sensors; and,

• Appendix D contains graphs of maximum geolocation errors for combined maximum uncertainties in sensor positions and TDOA measurements for the examples in Chapter 4.
Chapter 2

Preliminaries

This chapter summarizes existing tools used in this thesis with minor variations.

2.1 Dot Product

The dot product of two real column vectors, \( \vec{v} \) and \( \vec{w} \), is given as

\[
\vec{v} \cdot \vec{w} = \vec{w} \cdot \vec{v} = \vec{v}^T \vec{w} = \vec{w}^T \vec{v} = |\vec{v}| |\vec{w}| \cos(\theta)
\]

(2.1)

where

\( \vec{v}, \vec{w} \) = real column vectors of dimension 2 x 1 or 3 x 1

\( |\vec{v}| \) = the magnitude of real vector \( \vec{v} = \sqrt{\vec{v}^T \vec{v}} \)

\( |\vec{w}| \) = the magnitude of real vector \( \vec{w} = \sqrt{\vec{w}^T \vec{w}} \)

\( \theta \) = the angle between \( \vec{v} \) and \( \vec{w} \) as illustrated in Figure 2.1.
Figure 2.1: Dot Product in $\mathbb{R}^2$

The dot product of unit vectors is simply the cosine of the angle between them. If $\vec{u}_1 = \vec{v}/|\vec{v}|$ and $\vec{u}_2 = \vec{w}/|\vec{w}|$, then $\vec{u}_1 \cdot \vec{u}_2 = \cos(\theta)$ as illustrated in Figure 2.2.

Figure 2.2: Dot Product of Unit Vectors in $\mathbb{R}^2$
\[ \Rightarrow \cos(\theta) = \vec{u}_1 \cdot \vec{u}_2 = \frac{\vec{v}^T \vec{w}}{\mid \vec{v} \mid \mid \vec{w} \mid} \quad (2.2) \]

### 2.2 Directional Cosines

Directional cosines are used to indicate an angle of arrival from or direction to a point of interest. Given a vector, \( \vec{S}_0 \), pointing from the origin to a point, \( S_0 \) not located at the origin, in a standard Euclidean frame, the directional cosine vector, \( \vec{u}_0 \), pointing from the origin toward the point, \( S_0 \), is simply the vector divided by its magnitude.

\[
\begin{align*}
R_0 &= |\vec{S}_0| = \sqrt{\vec{S}_0^T \vec{S}_0} \\
\vec{u}_0 &= \frac{\vec{S}_0}{R_0} = \frac{\vec{S}_0}{|\vec{S}_0|} = \frac{S_0}{R_0} = [u_{0,x}, u_{0,y}, u_{0,z}]^T = [\cos(\alpha), \cos(\beta), \cos(\gamma)]^T \quad (2.3)
\end{align*}
\]

Each element of \( \vec{u}_0 \) corresponds to the directional cosine with respect to the x, y, and z direction, \( \vec{e}_x = [1, 0, 0]^T \), \( \vec{e}_y = [0, 1, 0]^T \), and \( \vec{e}_z = [0, 0, 1]^T \), respectively.

\[
\begin{align*}
\cos(\alpha) &= \vec{u}_0 \cdot \vec{e}_x = u_{0,x} \\
\cos(\beta) &= \vec{u}_0 \cdot \vec{e}_y = u_{0,y} \\
\cos(\gamma) &= \vec{u}_0 \cdot \vec{e}_z = u_{0,z}
\end{align*}
\]

Any point, \( S_0 \), referenced to the origin in a reference frame, other than the origin,
can be represented as a vector $\vec{S}_0 = R_0 \vec{u}_0$, where $\vec{u}_0$ is the directional cosine vector formed by dividing $\vec{S}_0$ by its magnitude as illustrated in Figure 2.3. This form is used to generate random signal source positions at a fixed distance from a reference frame in subsequent simulations. It is also used to form random sensor perturbations from unperturbed sensor positions.

Figure 2.3: Directional Cosines
For any point on the unit sphere centered at the origin, \( |\vec{u}_0| = 1 \), the directional cosines have only two degrees of freedom. Any two of the directional angles may be chosen and the third is constrained to remain on the unit sphere per Equation 2.4.

\[
\vec{u}_0^T \vec{u}_0 = u_{0,x}^2 + u_{0,y}^2 + u_{0,z}^2 = 1
\]

\[\Rightarrow \cos^2(\alpha) + \cos^2(\beta) + \cos^2(\gamma) = 1 \quad (2.4)\]

### 2.3 Matrix Rotations

A Euclidean coordinate system is generally assumed in \( \mathbb{R}^3 \). Rotation in \( \mathbb{R}^3 \) by a positive angle, \( \phi \), around the x-axis is a rotation of the y- and z-axes in a counter clockwise direction when looking at the origin from a point on the positive x-axis, and is implemented with the pre-multiplication matrix, \( R_x(\phi) \), defined as:

\[
R_x(\phi) = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(\phi) & -\sin(\phi) \\
0 & \sin(\phi) & \cos(\phi)
\end{bmatrix}, \text{ Rotation around x-axis by an angle of } \phi \quad (2.5)
\]

Similarly, rotation in \( \mathbb{R}^3 \) by a positive angle, \( \psi \), around the y-axis, and by a positive angle, \( \theta \), around the z-axis are implemented with the pre-multiplication matrices \( R_y(\psi) \) and \( R_z(\theta) \) respectively.
\[ R_y(\psi) = \begin{bmatrix} \cos(\psi) & 0 & \sin(\psi) \\ 0 & 1 & 0 \\ -\sin(\psi) & 0 & \cos(\psi) \end{bmatrix} \], Rotation around y-axis by an angle of \( \psi \) (2.6)

\[ R_z(\zeta) = \begin{bmatrix} \cos(\zeta) & -\sin(\zeta) & 0 \\ \sin(\zeta) & \cos(\zeta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \], Rotation around z-axis by an angle of \( \zeta \) (2.7)

### 2.4 Hyperbolic Nature of TDOAs

Assuming constant signal speeds and direct signal paths, the time difference of arrival of a signal from a source to a pair of sensors is the difference of the distances from the source to each sensor divided by the speed of the signal. The signal speed, \( v \), could be that of light, sound, earth tremors, water vibrations or other pressure or wave transmissions depending on the application. Under these assumptions, constant time differences of arrival are equivalent to constant signal path differences.

For the two dimensional case, assume the sensors are a distance \( D_{12} \) apart and located on the x-axis at \( P_1 = (-D_{12}/2, 0) \) and \( P_2 = (+D_{12}/2, 0) \) as shown in Figures 2.4 and 2.5 for \( D_{12} = 10 \). Let \( S_0 = (x, y) \) be the signal point source, and let

\[ d_1 = |S_0 - P_1| = \sqrt{(x + D_{12}/2)^2 + (y - 0)^2} \]

\[ d_2 = |S_0 - P_2| = \sqrt{(x - D_{12}/2)^2 + (y - 0)^2} \]
\[ \Delta d_{12} = d_1 - d_2 = v(t_1 - t_2) = v\Delta t_{12} \]

\[ \Delta d_{12} = \sqrt{(x + D_{12}/2)^2 + y^2} - \sqrt{(x - D_{12}/2)^2 + y^2} \]

where \( t_1 \) is the time of signal arrival at \( P_1 \), and \( t_2 \) is the time of signal arrival at \( P_2 \).

Rearranging and squaring,

\[ \Delta d_{12}^2 + 2\Delta d_{12}\sqrt{(x - D_{12}/2)^2 + y^2} + (x - D_{12}/2)^2 + y^2 = (x + D_{12}/2)^2 + y^2 \]

\[ 2\Delta d_{12}\sqrt{(x - D_{12}/2)^2 + y^2} = 2xD_{12} - \Delta d_{12}^2 \]

and then squaring again,

\[ 4\Delta d_{12}^2 \left[ (x - D_{12}/2)^2 + y^2 \right] = 4x^2D_{12}^2 - 4xD_{12}\Delta d_{12}^2 + \Delta d_{12}^4 \]

\[ 4\Delta d_{12}^2 \left[ x^2 - xD_{12} + D_{12}^2/4 + y^2 \right] = 4x^2D_{12}^2 - 4xD_{12}\Delta d_{12}^2 + \Delta d_{12}^4 \]

\[ 4x^2\Delta d_{12}^2 - 4xD_{12}\Delta d_{12}^2 + D_{12}^2\Delta d_{12}^2 + 4y^2\Delta d_{12}^2 = 4x^2D_{12}^2 - 4xD_{12}\Delta d_{12}^2 + \Delta d_{12}^4 \]

\[ 4x^2(D_{12}^2 - \Delta d_{12}^2) - 4y^2\Delta d_{12}^2 = \Delta d_{12}^2(D_{12}^2 - \Delta d_{12}^2) \] (2.8)

and finally, dividing by the terms on the right hand side of Equation 2.8 results in the well known constant TDOA hyperbolic equation in \( \mathbb{R}^2 \), Equation 2.9, as reported in [Gustafsson and Gunnarsson, 2003]. The constant distance difference, \( \Delta d_{12} \), is formed from the product of the constant TDOA, \( \Delta t_{12} \), and the signal speed, \( v \), \( \Rightarrow \Delta d_{12} = v\Delta t_{12} \).

\[ \Rightarrow \frac{x^2}{\Delta d_{12}^2/4} - \frac{y^2}{D_{12}^2/4 - \Delta d_{12}^2/4} = 1 \] (2.9)
Letting $D_{12} = 10$ and $\Delta d_{12} = \pm 6$, the sensors are then located at $P_1 = (-5, 0)$ and at $P_2 = (5, 0)$ and Equation 2.9 becomes

$$\frac{x^2}{9} - \frac{y^2}{16} = 1$$

with asymptotes at $y = \pm \frac{4}{3}x$. The locus of points satisfying $\Delta d_{12} = -6$ in $\mathbb{R}^2$ is shown in blue in Figure 2.4, and the locus of points satisfying $\Delta d_{12} = +6$ is shown in Figure 2.5 with the asymptotes plotted with red dashed lines.

Figure 2.4: Points in $\mathbb{R}^2$ for $\Delta d_{12} = -6$
The locus of constant TDOA points in $\mathbb{R}^3$ can be viewed by rotating the blue hyperbolas in Figures 2.4 and 2.5 about the x-axis using $R_x(\phi)$ in Equation 2.5, as shown in Figures 2.6 and 2.7. In this case, the rotation is about the x-axis, since this is the line on which the sensor pair lies. In general, the rotation would be about the line passing through the two sensor positions.
Figure 2.6: Points in $\mathbb{R}^3$ for $\Delta d_{12} = -6$
The cosine of the angle between the line passing through the sensor pair and the line passing through the midpoint of the sensor pair and a point on the constant TDOA hyperbola is used to create the functions that are minimized in a least squares sense. Referencing Figure 2.8, \( \cos(\theta) = \frac{x_0}{R_{12}} \), where \( R_{12} \) is the distance between the sensors’ midpoint, \( M_{12} = (P_1 + P_2)/2 = (0, 0) \), and a point on the hyperbola, \( S_0 = (x_0, y_0) \).
\[
\cos(\theta) = \frac{x_0}{R_{12}}
\]
\[
R_{12} = \sqrt{x_0^2 + y_0^2} = |S_0 - M_{12}|
\]
\[
M_{12} = \frac{(P_1 + P_2)}{2}
\]

Figure 2.8: Constant TDOA Hyperbola in \( \mathbb{R}^2 \)

Recognizing \( R_{12} = \sqrt{x_0^2 + y_0^2} \), \( \cos(\theta) \) is derived from Equation 2.9 as follows.

\[
1 = \frac{x_0^2}{(\Delta d_{12}^2/4)} - \frac{y_0^2}{(D_{12}^2/4 - \Delta d_{12}^2/4)}
\]
\[
y_0^2 = \left[ (D_{12}^2 - \Delta d_{12}^2)/4 \right] \left[ \frac{x_0^2}{(\Delta d_{12}^2/4)} - 1 \right]
\]
\[
y_0^2 = (D_{12}^2/\Delta d_{12}^2)x_0^2 - x_0^2 - (D_{12}^2 - \Delta d_{12}^2)/4
\]
\[ R_{12}^2 = x_0^2 + y_0^2 \]

\[ R_{12}^2 = (D_{12}^2/\Delta d_{12}^2)x_0^2 - (D_{12}^2 - \Delta d_{12}^2)/4 \]

\[ \Rightarrow \cos (\theta) = \frac{x_0}{R_{12}} = \frac{\Delta d_{12}}{D_{12}} \sqrt{1 + \frac{(D_{12}^2 - \Delta d_{12}^2)}{4R_{12}^2}} \] (2.10)

Note that Equation 2.10 holds for all points on the constant TDOA hyperbola in \( \mathbb{R}^2 \) other than the midpoint of the sensor pair, where \( R_{12} = 0 \). It also holds for all points on a constant TDOA hyperboloid in \( \mathbb{R}^3 \), formed by rotating the constant TDOA hyperbola in \( \mathbb{R}^2 \) around the line passing through the sensor pair, as illustrated in Figure 2.9, with the same midpoint exception. The approximation, \( \cos(\theta) \approx \Delta d_{12}/D_{12} \), applies to the asymptotes, and the sign of the square root is included in the sign of \( \Delta d_{12} \).
\[
\cos(\theta) = \frac{x_0}{R_{12}} = \frac{(\Delta d_{12}/D_{12})}{\sqrt{1 + (D_{12}^2 - \Delta d_{12}^2)/(4R_{12}^2)}}
\]  

\[
R_{12} = \sqrt{x_0^2 + y_0^2 + z_0} = |S_0 - (P_1 + P_2)/2| = |S_0 - M_{12}|
\]

\[
\Delta d_{12} = |S_0 - P_1| - |S_0 - P_2| = v \Delta t_{12}
\]

\[
D_{12} = |P_1 - P_2|
\]

Figure 2.9: Constant TDOA Hyperboloid in \(\mathbb{R}^3\)
2.5 Ezzat G. Bakhoum’s 5-Sensor Solution

Ezzat G. Bakhoum [Bakhoum, 2006] derived a closed-form solution for 3D geolocation of a signal source position using only TDOAs and five known sensor positions. His solution, $\vec{S}_0$, is independent of direction and range of the signal source location. 5 sensors are generally required to unambiguously geolocate a source signal in 3 dimensions using only TDOAs between sensor pairs with known sensor positions. This can be seen from Bakhoum’s starting point.

Given a source position vector, $\vec{S}_0 = [x_0, y_0, z_0]^T$, from which a signal is emitted at time $t_0$, and given 2 sensor position vectors, $\vec{P}_1$ and $\vec{P}_2$, and the corresponding times of signal arrival at each sensor, $t_1$ and $t_2$, respectively, the difference of the square of the signal path distance between the source and each sensor, under the direct signal path assumption, is written as:

$$d_2^2 - d_1^2 = v^2 \left[ (t_2 - t_0)^2 - (t_1 - t_0)^2 \right], \text{ where}$$

$$d_2^2 = \left| \vec{P}_2 - \vec{S}_0 \right|^2,$$

$$d_1^2 = \left| \vec{P}_1 - \vec{S}_0 \right|^2, \text{ and}$$

$$v = c = \text{the signal speed}$$

Expanding the squares,

$$\left| \vec{P}_2 - \vec{S}_0 \right|^2 - \left| \vec{P}_1 - \vec{S}_0 \right|^2 = v^2 \left[ t_2^2 - 2t_2t_0 + t_0^2 - t_1^2 - 2t_1t_0 + t_0^2 \right]$$
\[
\left| \mathbf{P}_2 \right|^2 - \left| \mathbf{P}_1 \right|^2 - 2 \left( \mathbf{P}_2^T - \mathbf{P}_1^T \right) \mathbf{S}_0 = v^2 \left[ (t_2^2 - t_1^2) - 2 (t_2 - t_1) t_0 \right]
\]

and then dividing by \((t_2 - t_1)\), results in the first sensor pair equation.

\[
\frac{1}{(t_2 - t_1)} \left[ \left| \mathbf{P}_2 \right|^2 - \left| \mathbf{P}_1 \right|^2 - 2 \left( \mathbf{P}_2^T - \mathbf{P}_1^T \right) \mathbf{S}_0 \right] = v^2 [(t_2 + t_1) - 2t_0] \tag{2.12}
\]

To eliminate \(t_0\) and produce a form having only sensor positions and TDOAs, a second equation is formed from a second sensor pair, \(\mathbf{P}_3\) and \(\mathbf{P}_1\).

\[
\frac{1}{(t_3 - t_1)} \left[ \left| \mathbf{P}_3 \right|^2 - \left| \mathbf{P}_1 \right|^2 - 2 \left( \mathbf{P}_3^T - \mathbf{P}_1^T \right) \mathbf{S}_0 \right] = v^2 [(t_3 + t_1) - 2t_0] \tag{2.13}
\]

Equation 2.12 is then subtracted from Equation 2.13.

\[
\frac{2}{(t_2 - t_1)} \left( \mathbf{P}_2^T - \mathbf{P}_1^T \right) - \frac{2}{(t_3 - t_1)} \left( \mathbf{P}_3^T - \mathbf{P}_1^T \right) \mathbf{S}_0 = \frac{1}{(t_2 - t_1)} \left( \left| \mathbf{P}_2 \right|^2 - \left| \mathbf{P}_1 \right|^2 \right) - \frac{1}{(t_3 - t_1)} \left( \left| \mathbf{P}_3 \right|^2 - \left| \mathbf{P}_1 \right|^2 \right) + v^2 (t_3 - t_2) \tag{2.14}
\]

If the time difference of arrivals and the sensor positions are known, then it is clear from Equation 2.12, Equation 2.13, and Equation 2.14, that there are 4 unknowns, \(x_0, y_0, z_0,\) and \(t_0\). 4 independent equations are required to solve for 4 unknowns. Each equation requires an independent pair of sensors of the same form as Equations 2.12 and 2.13. With no additional information, 5 sensors are required for 4 independent sensor pairs to obtain 4 independent equations to locate the source.
in three dimensions.

Given 5 sensor position vectors, \( \vec{P}_1, \vec{P}_2, \ldots, \vec{P}_5 \), and 7 TDOAs, \((t_2 - t_1), (t_3 - t_1), (t_4 - t_1), (t_5 - t_1), (t_3 - t_2), (t_4 - t_2), (t_5 - t_2)\), then the following system of linear equations can be solved for \( \vec{S}_0 \).

\[
\begin{align*}
\frac{2}{(t_2 - t_1)} \left( \vec{P}_2^T - \vec{P}_1^T \right) - \frac{2}{(t_3 - t_1)} \left( \vec{P}_3^T - \vec{P}_1^T \right) \vec{S}_0 &= \\
\frac{1}{(t_2 - t_1)} \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - \frac{1}{(t_3 - t_1)} \left( \left| \vec{P}_3 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_3 - t_2), \\
\frac{2}{(t_2 - t_1)} \left( \vec{P}_2^T - \vec{P}_1^T \right) - \frac{2}{(t_4 - t_1)} \left( \vec{P}_4^T - \vec{P}_1^T \right) \vec{S}_0 &= \\
\frac{1}{(t_2 - t_1)} \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - \frac{1}{(t_4 - t_1)} \left( \left| \vec{P}_4 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_4 - t_2), \\
\frac{2}{(t_2 - t_1)} \left( \vec{P}_2^T - \vec{P}_1^T \right) - \frac{2}{(t_5 - t_1)} \left( \vec{P}_5^T - \vec{P}_1^T \right) \vec{S}_0 &= \\
\frac{1}{(t_2 - t_1)} \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - \frac{1}{(t_5 - t_1)} \left( \left| \vec{P}_5 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_5 - t_2)
\end{align*}
\]

It is possible for the TDOAs to be zero. To avoid division by 0, the time differences are moved to the numerators.

\[
\begin{align*}
2(t_3 - t_1) \left( \vec{P}_2^T - \vec{P}_1^T \right) - 2(t_2 - t_1) \left( \vec{P}_3^T - \vec{P}_1^T \right) \vec{S}_0 &= \\
(t_3 - t_1) \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - (t_2 - t_1) \left( \left| \vec{P}_3 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_3 - t_2)(t_2 - t_1)(t_3 - t_1), \\
2(t_4 - t_1) \left( \vec{P}_2^T - \vec{P}_1^T \right) - 2(t_2 - t_1) \left( \vec{P}_4^T - \vec{P}_1^T \right) \vec{S}_0 &= \\
(t_4 - t_1) \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - (t_2 - t_1) \left( \left| \vec{P}_4 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_4 - t_2)(t_2 - t_1)(t_4 - t_1), \\
2(t_5 - t_1) \left( \vec{P}_2^T - \vec{P}_1^T \right) - 2(t_2 - t_1) \left( \vec{P}_5^T - \vec{P}_1^T \right) \vec{S}_0 &=
\end{align*}
\]
\[(t_5 - t_1) \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - (t_2 - t_1) \left( \left| \vec{P}_3 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_5 - t_2)(t_2 - t_1)(t_5 - t_1) \]

These equations can be written as:

\[
\begin{align*}
\vec{a}_1^T &= \left[ 2(t_3 - t_1) \left( \vec{P}_2^T - \vec{P}_1^T \right) - 2(t_2 - t_1) \left( \vec{P}_3^T - \vec{P}_1^T \right) \right], \\
\vec{a}_2^T &= \left[ 2(t_4 - t_1) \left( \vec{P}_2^T - \vec{P}_1^T \right) - 2(t_2 - t_1) \left( \vec{P}_4^T - \vec{P}_1^T \right) \right], \\
\vec{a}_3^T &= \left[ 2(t_5 - t_1) \left( \vec{P}_2^T - \vec{P}_1^T \right) - 2(t_2 - t_1) \left( \vec{P}_5^T - \vec{P}_1^T \right) \right], \\
b_1 &= (t_3 - t_1) \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - (t_2 - t_1) \left( \left| \vec{P}_3 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_3 - t_2)(t_2 - t_1)(t_3 - t_1), \\
b_2 &= (t_4 - t_1) \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - (t_2 - t_1) \left( \left| \vec{P}_4 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_4 - t_2)(t_2 - t_1)(t_4 - t_1), \\
b_3 &= (t_5 - t_1) \left( \left| \vec{P}_2 \right|^2 - \left| \vec{P}_1 \right|^2 \right) - (t_2 - t_1) \left( \left| \vec{P}_5 \right|^2 - \left| \vec{P}_1 \right|^2 \right) + v^2(t_5 - t_2)(t_2 - t_1)(t_5 - t_1),
\end{align*}
\]

\[
\begin{bmatrix}
\vec{a}_1^T \\
\vec{a}_2^T \\
\vec{a}_3^T
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
b_2 \\
b_3
\end{bmatrix}, 
\Rightarrow \begin{bmatrix}
\vec{a}_1^T \\
\vec{a}_2^T \\
\vec{a}_3^T
\end{bmatrix} = \begin{bmatrix}
b_1 \\
b_2 \\
b_3
\end{bmatrix}
\]

A 5-sensor source position solution, \( \vec{S}_0 \), to the linear system of equations, Equation

\[ (2.15) \]
2.15, is used as the starting point in the least squares algorithm, when $A_5$ is not ill-conditioned.

### 2.6 Binomial Theorem with Fractional Exponents

The binomial theorem with fractional exponents is an application of the Maclaurin series. This truncated series sum improves the accuracy of the distance difference between a source and two sensors. The series sum is used when the distance from a source to the midpoint of two sensors is greater than or equal to the distance between the two sensors. This distance difference calculation divided by the signal speed simulates the time difference of arrival of the source signal at the two sensors under the given assumptions.

Consider the binomial theorem application to $(1 - \alpha)^p$.

$$(1 - \alpha)^p = 1 - p\alpha +$$

$$\frac{1}{2!} p(p - 1)\alpha^2 - \frac{1}{3!} p(p - 1)(p - 2)\alpha^3 +$$

$$\frac{1}{4!} p(p - 1)(p - 2)(p - 3)\alpha^4 - \frac{1}{5!} p(p - 1)(p - 2)(p - 3)(p - 4)\alpha^5 +$$

$$\frac{1}{6!} p(p - 1)(p - 2)(p - 3)(p - 4)(p - 5)\alpha^6 - \ldots + \ldots$$

Choosing $p = 1/2$ for the square root function, then

$$\sqrt{1 - \alpha} = 1 + \sum_{n=1}^{\infty} (-1)^n a_n \alpha^n$$

where

$$a_0 = 1,$$
\[ a_n = (1.5/n - 1) a_{n-1}, \quad \forall n \geq 1 \]

The sum converges for \(|\alpha| < 1\) per the Ratio test.

### 2.7 Machine Precision

An implementation parameter that affects power, resources, speed, and accuracy is the machine epsilon, \(\text{eps}\), defined as the smallest number, which, when added to 1, returns a value greater than 1. A smaller machine epsilon usually increases accuracy, power dissipation, resource consumption, and the computational time to reach convergence. For hardware implementations, the choice of machine epsilon can reduce power, speed, and resource consumption, without loss of accuracy, when limited by input data uncertainty.

Simulations were implemented in Matlab. Matlab uses double precision floating point binary numbers. Per the IEEE 754 standard, double precision floating point numbers are stored in a 64-bit [63-0] binary format, as shown in Figure 2.10. 52 bits are allocated to the mantissa, \(M_b = [51 : 0] := (m_{51} : m_0)_2\), 11 bits to the exponent, \(E_b = [62 : 53] := (e_{10} : e_0)_2\), and 1 bit to the sign, \(S_b = [63] := (s_0)_2\).

The smallest number greater than 1, \(1 + \text{eps}\), is determined by the number of mantissa bits, \(N_m\). For double precision floating point numbers, with \(N_m = 52\), \(\text{eps} = 2^{-N_m} = 2^{-52} \approx 2.220446049250313\times10^{-16}\). Simulation results here are consistent with accuracy limitations of double precision floating point binary numbers.
Double Precision Floating Point Binary Format

\[ S_b = [63] := (s_0)_2, \quad s_0 \in \{0, 1\} \]
\[ E_b = [62 : 52] := (e_{10} : e_0)_2, \quad e_i \in \{0, 1\}, \quad \forall i = 0, 1, \ldots, 10 \]
\[ M_b = [51 : 0] := (m_{51} : m_0)_2, \quad m_j \in \{0, 1\}, \quad \forall j = 0, 1, \ldots, 51 \]

Double Precision Floating Point Binary to Decimal Conversion

\[ S_d = (-1)^{s_0} \]
\[ E_d = \sum_{i=0}^{10} e_i 2^i \]
\[ M_d = \sum_{j=0}^{51} m_j 2^{(j-52)} \]

\[ \Rightarrow N_d = \begin{cases} 
  S_d \left[1 + M_d\right] 2^{(E_d-1023)}, & \text{when } E_d > 0 \text{ for normal numbers} \\
  S_d \left[M_d\right] 2^{-1022}, & \text{when } E_d = 0 \text{ for subnormal numbers}
\end{cases} \]

Figure 2.10: IEEE 754 Double Precision Floating Point Format

The machine epsilon of some IEEE 754 standard floating point binary formats are shown in Table 2.1.

Table 2.1: IEEE 754 Standard Floating Point Binary Formats

<table>
<thead>
<tr>
<th>Floating Point Precision</th>
<th>Total Bits</th>
<th>Sign Bits</th>
<th>Exponent Bits</th>
<th>Mantissa Bits</th>
<th>Machine Epsilon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half Precision</td>
<td>16</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>(2^{-10})</td>
</tr>
<tr>
<td>Single Precision</td>
<td>32</td>
<td>1</td>
<td>8</td>
<td>23</td>
<td>(2^{-23})</td>
</tr>
<tr>
<td>Double Precision</td>
<td>64</td>
<td>1</td>
<td>11</td>
<td>52</td>
<td>(2^{-52})</td>
</tr>
<tr>
<td>Quadruple Precision</td>
<td>128</td>
<td>1</td>
<td>15</td>
<td>112</td>
<td>(2^{-112})</td>
</tr>
<tr>
<td>Octuple Precision</td>
<td>256</td>
<td>1</td>
<td>19</td>
<td>236</td>
<td>(2^{-236})</td>
</tr>
</tbody>
</table>
Advancements in digital technology, particularly in field-programmable gate arrays (FPGAs), allow algorithm implementations using custom numerical machine precisions optimized for specific applications without restriction to the standard.

2.8 Modified Levenberg-Marquardt Algorithm

The formulation of a non-linear least squares minimization problem is presented in Section 3.6 for the geolocation of a signal source in 3D using an over constrained set of TDOA equations. The Levenberg-Marquardt (LM) algorithm, [Levenberg, 1944] and [Marquardt, 1963], is commonly used in nonlinear least squares numerical solutions.

The LM algorithm is a blend of the gradient method, also known as the steepest descent method, and the Gauss-Newton method. The gradient method moves the iterative solution in the direction opposite to that of the gradient of a cost function. For a least squares fit to an over constrained number of analytic functions, the cost function is typically the sum of the square of the real valued functions. The Gauss-Newton method attempts to move the iterative solution in a direction that includes second order effects and is often unstable without protection from matrix singularities.

At large distances from the optimal solution, the LM algorithm adopts steepest descent characteristics. As the iterative updates approach the least squares solution, it adopts Gauss-Newton characteristics and improves the rate of convergence. The transition between the two is accomplished by means of a scalar, \( \mu \). The scalar is multiplied by a diagonal matrix, \( D \), with positive diagonal elements, and added to
an estimate of the second derivative of the cost function using the Jacobian, $J$. This forms a nonsingular matrix, $[J^T J + \mu D]$, $\forall \mu > 0$.

For real valued functions, the LM algorithm is generally applied to minimizing the sum of their squares, $F_r(S) = \frac{1}{2} \sum_{i=1}^{N} [f_i^2(S)]$. For the application herein, it is possible for the functions, defined in Chapter 3, to assume complex values. To minimize a real valued cost function, the sum of the square of their absolute values is used instead, $F_c(S) = \frac{1}{2} \sum_{i=1}^{N} \left| f_i(S) f_i^*(S) \right|$. A variation of the LM algorithm used to minimize $F_c(S)$ with respect to $S$, a 3x1 real valued vector in Cartesian coordinates, is summarized as follows.

Given an existing or starting source location vector, $\hat{S}$, it is desired to find an incremental vector, $\delta \hat{S}$, that approaches the least squares solution in some optimal way. Each analytic function, $f_i(\hat{S} + \delta \hat{S})$, can be expanded in a Taylor series. Retaining only second order terms of $\delta \hat{S}$ in the initial expansion,

$$f_i(\hat{S} + \delta \hat{S}) = f_i(\hat{S}) + \delta \hat{S}^T \nabla_{\hat{S}} f_i(\hat{S}) + \frac{1}{2} \delta \hat{S}^T H_i \delta \hat{S},$$

(2.16)

where $H_i = \begin{bmatrix} \frac{\partial^2 f_i(\hat{S})}{\partial x^2} & \frac{\partial^2 f_i(\hat{S})}{\partial x \partial y} & \frac{\partial^2 f_i(\hat{S})}{\partial x \partial z} \\ \frac{\partial^2 f_i(\hat{S})}{\partial y \partial x} & \frac{\partial^2 f_i(\hat{S})}{\partial y^2} & \frac{\partial^2 f_i(\hat{S})}{\partial y \partial z} \\ \frac{\partial^2 f_i(\hat{S})}{\partial z \partial x} & \frac{\partial^2 f_i(\hat{S})}{\partial z \partial y} & \frac{\partial^2 f_i(\hat{S})}{\partial z^2} \end{bmatrix}$ is the Hessian matrix of $f_i(\hat{S})$.

$$F_c(\hat{S} + \delta \hat{S}) = \frac{1}{2} \sum_{i=1}^{N} \left| f_i(\hat{S} + \delta \hat{S}) f_i^*(\hat{S} + \delta \hat{S}) \right|$$

(2.17)
Plug Equation 2.16 into Equation 2.17.

$$\Rightarrow F_c(\vec{S} + \delta \vec{S}) = \frac{1}{2} \sum_{i=1}^{N} \left[ \left( f_i(\vec{S}) + \delta \vec{S}^T \nabla_{\vec{S}} f_i(\vec{S}) + \frac{1}{2} \delta \vec{S}^T H_i \delta \vec{S} \right) + \left( f_i(\vec{S}) + \delta \vec{S}^T \nabla_{\vec{S}} f_i(\vec{S}) + \frac{1}{2} \delta \vec{S}^T H_i \delta \vec{S} \right)^* \right]$$

To determine the optimal increment, $\delta \vec{S}$, the derivative of $F_c(\vec{S} + \delta \vec{S})$ is taken with respect to $\delta \vec{S}$ and set to 0.

$$0 = \nabla_{\delta \vec{S}} F_c(\vec{S} + \delta \vec{S})$$

$$0 = \frac{1}{2} \sum_{i=1}^{N} \left[ \left( \nabla_{\vec{S}} f_i(\vec{S}) + H_i \delta \vec{S} \right) \left( f_i(\vec{S}) + \delta \vec{S}^T \nabla_{\vec{S}} f_i(\vec{S}) + \frac{1}{2} \delta \vec{S}^T H_i \delta \vec{S} \right)^* \right]$$

$$+ \left( f_i(\vec{S}) + \delta \vec{S}^T \nabla_{\vec{S}} f_i(\vec{S}) + \frac{1}{2} \delta \vec{S}^T H_i \delta \vec{S} \right) \left( \nabla_{\vec{S}} f_i(\vec{S}) + H_i \delta \vec{S} \right)^*$$

Retaining only first order terms of $\delta \vec{S}$ in the first vector derivative,

$$0 = \frac{1}{2} \sum_{i=1}^{N} \left[ \left( \nabla_{\vec{S}} f_i(\vec{S}) f_i^*(\vec{S}) + \nabla_{\vec{S}} f_i^*(\vec{S}) f_i(\vec{S}) \right) \right]$$

$$+ \left[ \nabla_{\vec{S}} f_i(\vec{S}) \nabla_{\vec{S}} f_i^*(\vec{S}) \right]^T + \left[ \nabla_{\vec{S}} f_i^*(\vec{S}) \nabla_{\vec{S}} f_i(\vec{S}) \right]^T \delta \vec{S}$$

$$+ \left[ H_i f_i^*(\vec{S}) + H_i^* f_i(\vec{S}) \right] \delta \vec{S}$$

$$0 = \frac{1}{2} \left[ J^T [\vec{f}(\vec{S})]^* + [J^*]^T \vec{f}(\vec{S}) \right]$$

$$+ \frac{1}{2} \left[ J^T J^* + [J^*]^T J \right] + \sum_{i=1}^{N} \left[ H_i f_i^*(\vec{S}) + H_i^* f_i(\vec{S}) \right] \delta \vec{S}$$

where $J = \left[ \nabla_{\vec{S}} f_1(\vec{S}), \nabla_{\vec{S}} f_2(\vec{S}), \ldots, \nabla_{\vec{S}} f_N(\vec{S}) \right]^T$
\[ \frac{\partial f_1(S)}{\partial x} \; \frac{\partial f_1(S)}{\partial y} \; \frac{\partial f_1(S)}{\partial z} \\
\frac{\partial f_2(S)}{\partial x} \; \frac{\partial f_2(S)}{\partial y} \; \frac{\partial f_2(S)}{\partial z} \\
\vdots \; \vdots \; \vdots \\
\frac{\partial f_N(S)}{\partial x} \; \frac{\partial f_N(S)}{\partial y} \; \frac{\partial f_N(S)}{\partial z} \]  

\[ J = \begin{bmatrix}
J_{11} & J_{12} & J_{13} \\
J_{21} & J_{22} & J_{23} \\
J_{31} & J_{32} & J_{33}
\end{bmatrix}, \]  

the Jacobian matrix of \( \vec{f}(\vec{S}) \), and

\[ \vec{f}(\vec{S}) = \left[ f_1(\vec{S}), f_2(\vec{S}), \ldots, f_N(\vec{S}) \right]^T \]

The optimal increment, \( \vec{\delta S} \), including the second order terms of the initial expansion, Equation 2.16, is then the solution to the following system of linear equations.

\[
\frac{1}{2} \left[ J^T J^* + [J^*]^T J \right] + \sum_{i=1}^{N} \left[ H_i f_i^*(\vec{S}) + H_i^* f_i(\vec{S}) \right] \vec{\delta S} = -\frac{1}{2} \left[ J^T \left[ \vec{f}(\vec{S}) \right]^* + [J^*]^T \vec{f}(\vec{S}) \right]
\]

This solution is a minimum if the second derivative of \( F_c(\vec{S}) \) with respect to \( \vec{\delta S} \) is positive definite.

\[
\nabla^2_{\vec{\delta S}} F_c(\vec{S} + \vec{\delta S}) = \frac{1}{2} \left[ J^T J^* + [J^*]^T J \right] + \sum_{i=1}^{N} \left[ H_i f_i^*(\vec{S}) + H_i^* f_i(\vec{S}) \right]
\]

\[
\Rightarrow \quad \vec{\delta S}^T \left[ \nabla^2_{\vec{\delta S}} F_c(\vec{S} + \vec{\delta S}) \right] \vec{\delta S} > 0, \quad \forall \vec{\delta S} \neq \vec{0}
\]

\[
\Rightarrow \quad \vec{\delta S}^T \left[ J^T J^* + [J^*]^T J \right] + \sum_{i=1}^{N} \left[ H_i f_i^*(\vec{S}) + H_i^* f_i(\vec{S}) \right] \vec{\delta S} > 0, \quad \forall \vec{\delta S} \neq \vec{0}
\]
It is customary to disregard the Hessian, since its elements are usually small compared to those in the matrix formed from the Jacobian, \([J^T J^* + [J^*]^T J]\). For this application, the Hessian was computed to be typically three orders of magnitude smaller and is ignored in the customary manner.

\[
\Rightarrow \frac{1}{2} \left[ J^T J^* + [J^*]^T J \right] \delta S \approx -\frac{1}{2} \left[ J^T [\overset{\sim}{f}(S)]^* + [J^*]^T \overset{\sim}{f}(S) \right]
\]

\[
\Rightarrow \Re(J^T J^*) \delta S = -\Re \left( J^T \left[ \overset{\sim}{f}(S) \right]^* \right)
\]

The left matrix, denoted as \(A = \Re(J^T J^*)\), causes instability in the Gauss-Newton method, when it is nearly singular or ill-conditioned. \(\forall \mu > 0\), the matrix, denoted as \(A_D = [\Re(J^T J^*) + \mu D]\), is positive definite, where the damping matrix, \(D\), is a diagonal matrix with positive diagonal elements. Adding this term avoids the instability of the Gauss-Newton method for appropriate values of the damping parameter, \(\mu\), and the diagonal elements of \(D\).

There are many ways to update the damping parameter and the damping matrix. The best method appears to depend on the application. The damping matrix update used here is similar to that described in [Transtrum and Sethna, 2012]. Updates to the damping parameter is based on a comparison of the cost function before and after the potential new solution vector. If there is no improvement in the cost function, then the updated solution vector is rejected and the damping parameter is moved in the direction of steepest descent. If there is improvement, then the updated solution
is accepted and the damping parameter is more slowly moved in the direction of Gauss-Newton. The following approach worked for this application.

At each step, the Jacobian, $J$, and the matrix, $A = \Re(J^TJ^*)$, are computed using the initial starting point or the latest accepted update. The maximum of the diagonal elements of $A$ is determined, $M_A = \max[a_{11}, a_{22}, a_{33}]$. A scaling factor, $C_D$, initially representing the maximum condition number for $D$, is chosen and each diagonal element of $D$ is set to the corresponding diagonal element of $A$, $M_A/C_D$, or the previous value, whichever is greater.

The increment, $\delta S$, is computed from the following system of linear equations.

$$
\begin{align*}
\left[\Re(J^TJ^*) + \mu D\right]\delta S &= -\Re\left(J^T\left[f(\hat{S})\right]^*\right)\mu \lambda \\
\left[A + \mu D\right]\delta S &= -\Re\left(J^T\left[f(\hat{S})\right]^*\right)\mu \lambda
\end{align*}
$$

where the initial values for the scalars are $\mu = M_A/20$ and $\lambda = 1$. The increment is added to $\hat{S}$ and the cost function, $F_c(\hat{S} + \delta S)$, is evaluated and compared with its prior value, $F_c(\hat{S})$.

If $F_c(\hat{S}) > F_c(\hat{S} + \delta S)$, then the update is accepted, the value of $\mu$ is reduced by the factor $(1/\mu_{\text{down}})$, and $\lambda$ is increased by the factor $\mu_{\text{down}}$. If $F_c(\hat{S}) < F_c(\hat{S} + \delta S)$, then the update is rejected, the value of $\mu$ is increased by the factor $\mu_{\text{up}}$, and $\lambda$ is reduced by the factor $(1/\mu_{\text{up}})$. This simple update replaces the positive definite computation of matrix $A$, moves toward second order convergence of Gauss-Newton, when possible, and retracts to the more stable first order convergence of steepest
Simulation results here reach the limits of numerical precision with a machine epsilon of $\epsilon = 2^{-52} \approx 2.220446049250313e-16$, for double precision floating point numbers, as described in Section 2.7. Additional checks are included to reach the numerical limits without loss of algorithm stability within a reasonable number of iterations and may not be required in typical applications.

The product, $\mu \lambda$, is the gradient gain. It remains a constant until the limits of machine numerical precision are approached. After each $\mu$ update, the ratio of $M_A / \mu$ is tested for the extremities. If $M_A / \mu > 1/(10 \times \epsilon)$, then $\mu$ is lower bounded and set to $\mu = M_A (10 \times \epsilon)$. If $M_A / \mu < (10 \times \epsilon)$, then $\mu$ is upper bounded and set to $\mu = M_A / (10 \times \epsilon)$.

Exceeding limits imposed on $\mu$ can result in loss of numerical precision, when adding the $A$ and $\mu D$ matrices. Bounding $\mu$ and allowing $\lambda$ to update without constraint has the effect of increasing the gradient gain, when $\mu$ is approaching the lower end of numerical precision, and reducing the gradient gain, when $\mu$ is approaching the upper end of numerical precision. Gradient gain adjustments compensate for the $\mu$-limiter and allow the algorithm to quickly exit.

At the extremes of error uncertainties and distances, the $A_D$ matrix can be nearly singular. To avoid the singularities, the condition number of the $A_D$ matrix, $\kappa(A_D)$, is computed, prior to solving for $\delta S$, and the diagonal elements of $D$ are adjusted until the condition number is no greater than $1/(10 \times \epsilon)$. $C_D$ is reduced by a factor of $10^3$. 

36
and the diagonal elements of $D$ are updated to $d_{ii} = \max(a_{ii}, M_{A}/C_{D}), \forall i = 1, 2, 3$. This is repeated until $\kappa(A_{D}) \leq 1/(10 \ast \mathbf{eps})$.

Given an acceptable convergence threshold, $\epsilon$, at each step, $n$, the search for a solution, $\hat{S}_{n}$, that lowers the cost function, $F_{c}$, halts, when one of the following three conditions is met.

- $|\delta S_{n}| / |\hat{S}_{n}| < \epsilon$, where $|\hat{S}_{n}|$ is a normalization factor, per Section 3.8
- $F_{c}(\hat{S}_{n})/R_{\text{max},n}^{2} < \epsilon^{2}$, where $R_{\text{max},n}$ is a normalization factor, per Section 3.8
- $n = 10,000$

Given $\hat{S}_{n}, F_{c}(\hat{S}_{n}), n \geq 0$, $\mu_{\text{down}} = 1.3$, $\mu_{\text{up}} = 2$, $\lambda_{0} = 1$, $C_{D} = 1,000$, $a_{11,0} = a_{22,0} = a_{33,0} = 0$, and $\epsilon = (10^{k} \ast \mathbf{eps})$, where $k \in \{0,1,2,\ldots\}$ is optimally chosen for the best rate of convergence without loss of precision as described in Section 3.9, the steps used in the modified Levenberg-Marquardt algorithm to compute the next update, $\hat{S}_{n+1}$, are as follows:

1. Compute the Jacobian, $\mathbf{J}_{n} = [\nabla_{\hat{S}_{n}} f_{1}(\hat{S}_{n}), \nabla_{\hat{S}_{n}} f_{2}(\hat{S}_{n}), \cdots, \nabla_{\hat{S}_{n}} f_{N}(\hat{S}_{n})]^{T}$

2. Compute $\mathbf{A}_{n} = \Re [\mathbf{J}_{n}^{T} \mathbf{J}_{n}] = [a_{ij,n}]_{i,j=1,2,3}$

3. Find the maximum diagonal element $M_{A_{n}} = \max[a_{11,n}, a_{22,n}, a_{33,n}]$

4. If $n = 0$, then set $\mu_{n} = \mu_{0} = M_{A_{n}}/20$

5. Form $\mathbf{D}_{n}$: $d_{ii,n} = \max[a_{ii,n}, M_{A_{n}}/C_{D}, a_{ii,n-1}], i = 1,2,3; d_{ij,n} = 0, \forall i \neq j$
6. Form $A_{Dn} = [A_n + \mu_n D_n]$ and compute $\kappa(A_{Dn})$

7. If $\kappa(A_{Dn}) > 1/(10*\text{eps})$, then set $C_D = C_D/10$ and go to step 5.

8. Else if $\kappa(A_{Dn}) \leq 1/(10*\text{eps})$, then go to step 9.

9. Solve $[A_{Dn}] \delta S_n = -\Re \left( J_n^T \left[ \hat{f}(S_n) \right]^* \right) \mu_n \lambda_n$ for $\delta S_n$

10. Compute $F_c(\hat{S}_n + \delta S_n)$

11. If $F_c(\hat{S}_n + \delta S_n) < F_c(\hat{S}_n)$, then

   11.1. Set $\hat{S}_{n+1} = \hat{S}_n + \delta S_n$

   11.2. Set $\lambda_{n+1} = \lambda_n \mu_{down}$

   11.3. Set $\mu_{n+1} = \mu_n / \mu_{down}$ and check $\mu_{n+1}$ for lower limit.

   11.4. If $\mu_{n+1} < M_A (10*\text{eps})$, then set $\mu_{n+1} = M_A (10*\text{eps})$

12. If $F_c(\hat{S}_n + \delta S_n) \geq F_c(\hat{S}_n)$, then

   12.1. Set $\hat{S}_{n+1} = \hat{S}_n$

   12.2. Set $\lambda_{n+1} = \lambda_n / \mu_{up}$

   12.3. Set $\mu_{n+1} = \mu_n \mu_{up}$ and check $\mu_{n+1}$ for upper limit.

   12.4. If $\mu_{n+1} > M_A/(10*\text{eps})$, then set $\mu_{n+1} = M_A/(10*\text{eps})$

13. Check exit criteria:

   13.1. If $|\delta S_n| / |\hat{S}_n| < \epsilon$, or $F_c(\hat{S}_n)/R_{max,n}^2 < \epsilon^2$, or $n = 10,000$, then
13.1.1. Set $\hat{S}_{LS} = \hat{S}_{n+1} = \text{the Least Squares Solution, and}$

13.1.2. Exit.

13.2. Else, increment $n$ and go to step 1.
Chapter 3

Models

The models used in the least squares solution for the geolocation of a signal source are described in this chapter. The time difference of arrival (TDOA) algorithm presented here generally works with five or more sensors using TDOAs at various sensor pairs. No knowledge of the source direction or range is assumed.

Seven sensors are used in the geolocation simulations, for reasons described in Section 3.7. Sensor positions and TDOAs of a source signal at sensor pairs are assumed known with various levels of maximum uncertainty. From this knowledge, the source signal position is iteratively solved in a “least squares” sense and the maximum distance between the least squares solution and the true source position is determined over all iterations with common parameters.
3.1 Error Sources

The accuracy of the three dimensional geolocation solution depends on the convergence threshold used in the Levenberg-Marquardt algorithm, the machine numerical precision, and uncertainties in sensor position knowledge and TDOA measurements. It is the goal of this thesis to quantify them and to serve as a system engineering requirement for each, given a desired maximum three dimensional signal source location error. The algorithm is independent of phenomenology, but simulation results here apply to electro-magnetic signals, that are either optical or radio frequency (RF) in nature, with only a direct signal path from source to each sensor traveling at a constant signal speed of \( v = c = 299,792,458 \text{ meters/second} \). Seismic, sonar, acoustic, or other signal propagating environments would use the appropriate signal speed for the application.

There are many possible sources of TDOA errors, including uncertainty in the distance between two sensors. TDOA measurements are often computed from the cross correlation of time stamped sensor input signals. Parameters contributing to TDOA errors include the signal to noise ratio and frequency content of received signals, errors in clock synchronizations between the sensors, quantization levels and sample rates in signal digitizations, amplifier and digitizer nonlinearities, observed bandwidth, among others. Violations of constant signal speed and direct signal path assumptions impact TDOA measurements and contribute to the source location error. Implementations required to minimize sensor position uncertainty and TDOA mea-
surement errors, and algorithm adjustments for signal speed fluctuations, multi-path environments, and source or sensor motion are beyond the scope of this work.

Input errors are modeled as uncertainties in either sensor positions, TDOA measurements, or both. Sensor position errors are modeled in one of two ways, depending on whether or not the source location is referenced to one of the sensors, as described in Section 3.3. TDOA measurement errors can be modeled as either random or correlated, depending on the prevailing nature of the noise. Since TDOA errors in the random noise model are at least as large as the correlated noise model, only the random noise model is used to find the largest worst case geolocation error, as described in Section 3.5.

Source location accuracy is the distance between a computed position and the true source position. It is determined over a range of parameter values effecting it, which include the distance between the unperturbed reference frame sensor and all other sensors, the signal source distance from the unperturbed reference frame sensor, the maximum uncertainty in the relative sensor angle positions, and the maximum uncertainty in the TDOAs.

\section{3.2 True Source Positions}

The true source positions, $\hat{R}_s$, are a product of the source distance, $R$, from the origin and a random source position directional cosine vector, $\hat{r}_s$, as described in Section 2.2. $R$ values start at 1 centimeter, are incremented by factors of 10, and
stop at a maximum value, that depends on the distance between the sensors, $B$, and
the magnitude of input uncertainties, described in Sections 3.3 and 3.5. The same
random source position directional cosine vectors are used for each value of $R$ so the
effects of other parameter changes are not skewed by changes in the source directions.

Source position directional cosine vectors are chosen from random directions over
$4\pi$ steradians and are located on the unit sphere centered at the origin. The direc-
tional cosine constraint of Equation 2.4, $\cos^2(\alpha_s) + \cos^2(\beta_s) + \cos^2(\gamma_s) = 1$, is used
to construct the directional cosine vector for each source position direction index,
$s = 1, 2, \ldots, 100$, as follows.

The first directional cosine angle, $\gamma_s$, is chosen from a uniform distribution that
is unrestricted over the entire range from $0$ to $\pi$. Given $\gamma_s$, the next directional
cosine angle, $\beta_s$, is chosen from a uniform distribution that is restricted over the
remaining solid angle not taken by $\gamma_s$. The final directional cosine angle, $\alpha_s$, consumes
the remainder of the unused solid angle and is restricted by the directional cosine
constraint, where the sign of its cosine is chosen randomly, either $+$ or $-$, with equal
probability.

A 3D plot of 100 random directional cosine vectors, $\vec{r}_s$, constructed from this
method, pointing from the origin to a point on the unit sphere, and used for the
source positions in the simulations, are shown in Figure 3.1.
\[ \vec{r}_s = [\cos(\alpha_s), \cos(\beta_s), \cos(\gamma_s)]^T, \text{ where} \]
\[ \gamma_s = \text{rand} \{0 : \pi\} \text{ radians}, \]
\[ \beta_s = \text{rand} \left[ \arccos \left( \sqrt{1 - \cos^2(\gamma_s)} \right) : \pi - \arccos \left( \sqrt{1 - \cos^2(\gamma_s)} \right) \right] \text{ radians}, \]
\[ \alpha_s = \begin{cases} 
\arccos \left( \sqrt{1 - \cos^2(\beta_s) - \cos^2(\gamma_s)} \right) & \text{ if } \rho_s > 0 \\
\pi - \arccos \left( \sqrt{1 - \cos^2(\beta_s) - \cos^2(\gamma_s)} \right) & \text{ if } \rho_s \leq 0
\end{cases}, \]
\[ \rho_s = \text{rand} \{-0.5 : 0.5\}, \]
\[ \Rightarrow \vec{R}_s = R \ast \vec{r}_s, \]
\[ R \in \{1e-2, 1e-1, 1e0, 1e1, \ldots\} \text{ meters}, \]
\[ s = 1, 2, \ldots, 100 = \text{the random source position direction index} \]

Figure 3.1: Source Directional Cosine Vectors

### 3.3 Sensor Positions and Perturbations

Simulations use seven sensors, \( N_S = 7 \), for reasons described in Section 3.7. There are two different models used for sensor perturbations, a relative sensor perturbation
model and a general sensor perturbation model.

The relative sensor perturbation model applies, when it is desired to locate a signal source position relative to one of the sensor positions, chosen as the reference frame and assumed to be fixed at the origin with no location uncertainty. An application example for this model may be a sensor mounted on a satellite with six other sensors mounted on the end of a boom attached to the same satellite. Another application may be where one of the sensors is located in a fixed position on the ground and the others are in the air or in space. And yet another may be where all of the sensors are mounted on independent satellites and one of the sensor positions may be known with a much higher accuracy than the others, or a signal source may be approaching one of the sensors. In this model, it is desired to find the source location relative to a reference frame attached to one sensor’s position.

The general sensor perturbation model can be used, when it is desired to locate the absolute source position with uncertainty in every sensor position. In this case, the source location is unlikely to be near any of the sensor positions. An application example for this model may be a sensor mounted on independent satellites with equal position uncertainty, and the absolute location of a distant source is desired. It is possible to use multiple applications of the relative sensor perturbation model to achieve the absolute location of the source, but this approach is beyond the scope of this paper. In this model, it is desired to find the source location relative to a reference frame that is not attached to any of the sensors’ positions.
In both models, the unperturbed reference frame sensor is located at the origin and the other six unperturbed sensors are located on a sphere of radius $B$ from the origin. $B$ values start at 1 meter, are incremented by factors of 10, and stop at 100,000 km.

The six unperturbed non-reference frame sensor positions, $P_2$ through $P_7$, are assumed to be oriented in the same direction from the center reference frame sensor, $P_1$, $\forall B$, approximately as shown in Figure 3.2. The unperturbed sensor positions are modeled as $B$ multiplied by their corresponding directional cosines, described in Section 2.2.

\[
P_1 = (0, 0, 0)
\]

\[
P_2 = B \times (-0.7071067811865476, 0, -0.7071067811865475)
\]

\[
P_3 = B \times (0.3535533905932737, 0.6123724356957946, -0.7071067811865475)
\]

\[
P_4 = B \times (0.3535533905932737, -0.6123724356957946, -0.7071067811865475)
\]

\[
P_5 = B \times (-0.7071067811865476, 0, 0.7071067811865475)
\]

\[
P_6 = B \times (0.3535533905932737, 0.6123724356957946, 0.7071067811865475)
\]

\[
P_7 = B \times (0.3535533905932737, -0.6123724356957946, 0.7071067811865475)
\]

$B \in \{1e0, 1e1, \ldots, 1e8\}$ meters
The corresponding vectors pointing to the sensor positions are formed from the matrix rotations, $\mathbf{R}_y(\psi)$ and $\mathbf{R}_z(\zeta)$, per Equation 2.6 and Equation 2.7, as follows.

\[
\vec{P}_1 = [0, 0, 0]^T \\
\vec{P}_2 = B \ast \mathbf{R}_y(-\psi)[-1, 0, 0]^T \\
\vec{P}_3 = B \ast \mathbf{R}_z(\zeta)\mathbf{R}_y(\psi)[1, 0, 0]^T \\
\vec{P}_4 = B \ast \mathbf{R}_z(-\zeta)\mathbf{R}_y(\psi)[1, 0, 0]^T \\
\vec{P}_5 = B \ast \mathbf{R}_y(\psi)[-1, 0, 0]^T \\
\vec{P}_6 = B \ast \mathbf{R}_z(\zeta)\mathbf{R}_y(-\psi)[1, 0, 0]^T \\
\vec{P}_7 = B \ast \mathbf{R}_z(-\zeta)\mathbf{R}_y(-\psi)[1, 0, 0]^T ,
\]

where $B \in \{1e0, 1e1, \ldots, 1e8\}$ meters, $\psi = \pi/4 \text{ radians}$, and $\zeta = \pi/3 \text{ radians}$
3.3.1 Relative Sensor Perturbation Model

In the relative sensor perturbation model, also referred to as the fixed reference frame model, the sensor located at the origin, \( P_1 = (0, 0, 0) \), remains unperturbed. The other six non-reference frame sensors are perturbed in two dimensions. Uncertainty in the third dimension, if any, is handled by TDOA perturbation models described in Section 3.5. The 2D angular uncertainty for the non-reference frame sensor positions is modeled as two successive rotations from their unperturbed position.

The first rotation is away from the sensor by the maximum assumed perturbation angle, denoted as \( \xi \). Non-zero values for \( \xi \) start at 1e-15 radians, are incremented by factors of 10, and stop at 1e-1 radians. The second rotation is around the unperturbed sensor location, denoted as \( \eta \). \( \eta \) is randomly chosen from a uniform distribution ranging from 0 to 2\( \pi \) radians. These two rotations are illustrated in Figure 3.3 starting from an unperturbed sensor located on the x-axis. \( \xi = 0 \) radians for the initial unperturbed sensor position.
\( \xi \in \{0, 1e-15, 1e-14, \ldots, 1e-1\} \) radians
\( \eta = \text{rand}[0 : 2\pi] \) radians
\( B \in \{1e0, 1e1, \ldots, 1e8\} \) meters

Figure 3.3: Non-Reference Frame Relative Sensor Perturbation

The corresponding vectors pointing to the perturbed sensor positions, \( \vec{P}_{t2} \) through \( \vec{P}_{t7} \), are formed in a similar manner as the unperturbed sensor positions, but include the two additional angular uncertainty matrix rotations described above. These transformations use the matrix rotations, \( R_y(\xi), R_x(\eta), R_y(\psi) \) and \( R_z(\zeta) \), per Equation 2.5, Equation 2.6 and Equation 2.7, in the following matrix multiplication order.

\[
\vec{P}_{t1} = \vec{P}_1 = [0, 0, 0]^T
\]

\[
\vec{P}_{t2} = B \ast R_y(-\psi)R_x(\eta_2)R_y(\xi)[-1, 0, 0]^T
\]

\[
\vec{P}_{t3} = B \ast R_z(\zeta)R_y(\psi)R_x(\eta_3)R_y(\xi)[1, 0, 0]^T
\]
\[ \vec{P}_t = B \ast \mathbf{R}_z(-\zeta)\mathbf{R}_y(\psi)\mathbf{R}_x(\eta_{kj})\mathbf{R}_y(\xi)[1, 0, 0]^T \]

\[ \vec{P}_{t5} = B \ast \mathbf{R}_y(\psi)\mathbf{R}_x(\eta_{kj})\mathbf{R}_y(\xi)[-1, 0, 0]^T \]

\[ \vec{P}_{t6} = B \ast \mathbf{R}_z(\zeta)\mathbf{R}_y(-\psi)\mathbf{R}_x(\eta_{kj})\mathbf{R}_y(\xi)[1, 0, 0]^T \]

\[ \vec{P}_{t7} = B \ast \mathbf{R}_z(-\zeta)\mathbf{R}_y(-\psi)\mathbf{R}_x(\eta_{kj})\mathbf{R}_y(\xi)[1, 0, 0]^T, \text{where} \]

\[ B \in \{1e0, 1e1, \ldots, 1e8\} \text{ meters}, \]

\[ \xi \in \{0, 1e-15, \ldots, 1e-1\} \text{ radians}, \]

\[ \eta_{kj} = \text{rand}[0 : 2\pi] \text{ radians, where} \]

\[ k = 2, 3, \ldots, 7, \text{ the non-reference frame sensor index}, \]

\[ j = 1, 3, \ldots, 20, \text{ the random } \eta \text{ rotation index}, \]

\[ j \text{ is independently chosen for each } k, \]

\[ \psi = \pi/4 \text{ radians, and } \zeta = \pi/3 \text{ radians} \]

It should be noted that geolocation errors associated with sensor pairs, that do not include the fixed reference frame sensor, are greater than those that do. For this reason, the least squares algorithm, in this model, uses only the \( N_S - 1 = 6 \) sensor pairs that include the reference frame sensor. Including other pairs produces larger source geolocation errors relative to the chosen sensor.
3.3.2 General Sensor Perturbation Model

In the general sensor perturbation model, also referred to as the floating reference frame model, every sensor position is perturbed in three dimensions, relative to a reference frame that is not attached to any sensor position. The 3D sensor position uncertainty is modeled as a sensor position perturbation vector, $\vec{p}_{tk}$, added to each unperturbed sensor position vector, $\vec{P}_k$, to form the perturbed sensor position vector, $\vec{P}_{tk} = \vec{P}_k + \vec{p}_{tk}$.

The sensor position perturbation vector, $\vec{p}_{tk}$, is the product of the error magnitude, $\varrho$, derived from the maximum assumed relative uncertainty angle, $\xi$, and a random 3D sensor position perturbation directional cosine vector, $\vec{\varphi}_{kj}$, where $k = 1, 2, \ldots, 7$, is the sensor position index, and $j = 1, 2, \ldots, 20$, is the random vector direction index. The sensor position perturbation directional cosine vector is formed in the same manner as the source position directional cosine vector, described in Section 3.2.

The error magnitude is the product of the unperturbed distance from the origin of the non-reference frame sensor positions, $B$, and half of the maximum uncertainty angle, $\xi/2$, as illustrated in Figure 3.4. In this model, the sensor position error magnitude, $\varrho$, is assumed to be equal for all sensors. In a practical application, the sensor position error magnitude could be tailored to match the knowledge of each sensor’s positional error accordingly.
\[ \varphi = B \times \xi/2, \]
\[ \xi \in \{0, 1e-15, 1e-14, \ldots, 1e-1\} \text{ radians}, \]
\[ B \in \{1e0, 1e1, \ldots, 1e8\} \text{ meters} \]

Figure 3.4: General Sensor Perturbation Amplitude

Sensor position perturbation directional cosine vectors, \( \varphi_{kj} \), are chosen from \( j \) random directions over \( 4\pi \) steradians independently for each \( k \), and are located on the unit sphere centered at the corresponding unperturbed sensor position, \( \hat{P}_k \). The directional cosine constraint of Equation 2.4, \( \cos^2(\alpha_{kj}) + \cos^2(\beta_{kj}) + \cos^2(\gamma_{kj}) = 1 \), is used to construct the directional cosine vector for each sensor position index, \( k = 1, 2, \ldots, 7 \), and for each random direction index, \( j = 1, 2, \ldots, 20 \), in the same manner as the source position directional cosine vectors.

The first sensor position perturbation directional cosine angle, \( \gamma_{kj} \), is chosen from
a random uniform distribution, that is unrestricted over the entire range from 0 to $\pi$. Given $\gamma_{kj}$, the next directional cosine angle, $\beta_{kj}$, is chosen from a random uniform distribution, that is restricted over the remaining solid angle not taken by $\gamma_{kj}$. The final directional cosine angle, $\alpha_{kj}$, consumes the remainder of the unused solid angle and is restricted by the directional cosine constraint, where the sign of its cosine is chosen randomly, either + or -, with equal probability.

A 3D plot of 20 random directional cosine vectors, $\vec{\varphi}_{1j}, j = 1, 2, \ldots, 20$, constructed from this method, pointing from the origin to a point on the unit sphere, and used for the $P_1$ sensor position perturbations in the simulations, are shown in Figure 3.5. Directional cosine vectors for the other sensor positions, $k = 2, 3, \ldots, 7$, are chosen similarly, but centered at their corresponding unperturbed sensor position.

The maximum least squares errors are computed over all sensor perturbation directions, $j = 1, 2, \ldots, 20$, and source directions, $s = 1, 2, \ldots, 100$, for each parameter set. The parameter set includes the unperturbed distance of all sensors, except $P_1$, from the origin, $B$, the maximum uncertainty angle of all sensors with respect to $P_1$, $\xi$, and the source distance from the origin, $R$, described in Section 3.2.
\[\vec{\varphi}_{kj} = [\cos(\alpha_{kj}), \cos(\beta_{kj}), \cos(\gamma_{kj})]^T,\]
\[\gamma_{kj} = \text{rand}[0 : \pi] \text{ radians},\]
\[\beta_{kj} = \text{rand} \left[ \arccos \left( \sqrt{1 - \cos^2(\gamma_{kj})} \right), \pi - \arccos \left( \sqrt{1 - \cos^2(\gamma_{kj})} \right) \right] \text{ radians},\]
\[\alpha_{kj} = \begin{cases} 
\arccos \left( \sqrt{1 - \cos^2(\beta_{kj}) - \cos^2(\gamma_{kj})} \right) \text{ radians, if } \rho_{kj} > 0 \\
\pi - \arccos \left( \sqrt{1 - \cos^2(\beta_{kj}) - \cos^2(\gamma_{kj})} \right) \text{ radians, if } \rho_{kj} \leq 0 
\end{cases},\]
\[\rho_{kj} = \text{rand}[-0.5 : 0.5],\]
\[\Rightarrow \vec{p}_{tk} = \vec{\varphi} * \vec{\varphi}_{kj}, \text{ the } k-\text{th indexed sensor position perturbation vector},\]
\[\vec{\varphi} = B * \xi / 2, \text{ the sensor position error magnitude},\]
\[B \in \{1e0, 1e1, \ldots, 1e8\} \text{ meters, the unperturbed distance from reference},\]
\[\xi \in \{0, 1e-15, \ldots, 1e-1\} \text{ radians, the max relative sensor perturbation angle},\]
\[k = 1, \text{ the first sensor position index shown in this Figure}\]
\[j = 1, 2, \ldots, 20, \text{ the random sensor position perturbation direction index}\]

Figure 3.5: Sensor Perturbation Directional Cosine Vectors for \(P_1\)

In practical applications, the actual sensor locations are known with limited accuracy. An upper bound may be known for the positional deflection uncertainty, but
the accuracy of the true position knowledge remains limited, even with calibration corrections. In all of the iterative solutions for the signal source position, the unperturbed sensor locations, \( P_1, P_2, \ldots, P_7 \), are used as the sensor position input to the least squares algorithm, since the perturbed sensor positions are unknown. Unperturbed TDOAs are computed from the true source position, \( R_s \), and the perturbed sensor positions.

### 3.4 Unperturbed TDOA Computations

Unperturbed TDOAs are computed from the distances between the true source position, described in Section 3.2, and the true sensor positions, described in Section 3.3, whether perturbed or not. Various levels of maximum uncertainty are added to these unperturbed TDOA computations and used as input to the least squares algorithm, described in Section 2.8. TDOA perturbations are described in Section 3.5.

There are two methods used to compute the unperturbed TDOA for a sensor pair. One method subtracts the distances between the true source position and each sensor pair, \( d_1 - d_k \), using Matlab’s ‘sqrt’ function as shown in Equation 3.1 of Figure 3.6. This method produces satisfactory results, when \( R_{1k} < D_{1k} \). However, when \( R_{1k} \geq D_{1k} \), the computational accuracy is inconsistent.

To improve the TDOA computational accuracy, an alternate method is used for computing unperturbed TDOAs, when \( R_{1k} \geq D_{1k} \). The alternate method provides
more consistent accuracy results and has demonstrated an improvement in accuracy over the difference of square roots by as much as 8 orders of magnitude using double precision floating point numbers. The second method uses the constant TDOA hyperbola, Equation 2.11, and the dot product, described in Section 2.1, to compute the TDOA and is derived as follows.

Given the true source position, \( R_s \), and true perturbed sensor positions, \( P_{t1} \) and \( P_{tk} \), the cosine of the relevant angle, \( \cos(\theta_{1k}) \), is computed from the dot product cosine, per Equation 2.2, as shown in Figure 3.6.

\[
\cos(\theta_{1k}) = \frac{(\vec{R}_s - \vec{M}_{1k})^T (\vec{P}_{tk} - \vec{P}_{t1})}{|\vec{R}_s - \vec{M}_{1k}| |\vec{P}_{tk} - \vec{P}_{t1}|}
\]

This cosine is used in the constant TDOA hyperbola and solved for \( \Delta d_{1k} \).

\[
\cos(\theta_{1k}) = \frac{\Delta d_{1k}}{D_{1k}} \sqrt{1 + \frac{(D_{1k}^2 - \Delta d_{1k}^2)}{4R_{1k}^2}}
\]

\[
D_{1k} \cos^2(\theta_{1k}) = \Delta d_{1k}^2 \left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right) - \Delta d_{1k}^4 \left(\frac{1}{4R_{1k}^2}\right)
\]

\[
\Rightarrow \Delta d_{1k}^2 = 2R_{1k}^2 \left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right) \left[1 \pm \sqrt{1 - \left(\frac{D_{1k}^2 \cos^2(\theta_{1k})}{R_{1k}^2}\right) \left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right)^2}\right]
\]

To determine the \( \pm \) sign, consider \( \theta_{1k} = 0 \), or \( \cos(\theta_{1k}) = 1 \).

\[
\Rightarrow \Delta d_{1k}^2 = 2R_{1k}^2 \left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right) \left[1 \pm \sqrt{1 - \left(\frac{D_{1k}^2}{R_{1k}^2}\right) \left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right)^2}\right]
\]
\[ d_1 = |\vec{R}_s - \vec{P}_{t1}| = \sqrt{\left(\vec{R}_s - \vec{P}_{t1}\right)^T \left(\vec{R}_s - \vec{P}_{t1}\right)} \]
\[ d_k = |\vec{R}_s - \vec{P}_{tk}| = \sqrt{\left(\vec{R}_s - \vec{P}_{tk}\right)^T \left(\vec{R}_s - \vec{P}_{tk}\right)} \]
\[ \Delta d_{1k} = \sqrt{\left(\vec{R}_s - \vec{P}_{t1}\right)^T \left(\vec{R}_s - \vec{P}_{t1}\right)} - \sqrt{\left(\vec{R}_s - \vec{P}_{tk}\right)^T \left(\vec{R}_s - \vec{P}_{tk}\right)} \quad (3.1) \]
\[ \vec{M}_{1k} = \left(\vec{P}_{t1} + \vec{P}_{tk}\right)/2 \]
\[ D_{1k} = |\vec{P}_{tk} - \vec{P}_{t1}| = \sqrt{\left(\vec{P}_{tk} - \vec{P}_{t1}\right)^T \left(\vec{P}_{tk} - \vec{P}_{t1}\right)} \]
\[ \vec{u}_{1k} = \left(\vec{P}_{tk} - \vec{P}_{t1}\right)/D_{1k} \]
\[ R_{1k} = |\vec{R}_s - \vec{M}_{1k}| = \sqrt{\left(\vec{R}_s - \vec{M}_{1k}\right)^T \left(\vec{R}_s - \vec{M}_{1k}\right)} \]
\[ \vec{v}_{1k} = \left(\vec{R}_s - \vec{M}_{1k}\right)/R_{1k} \]
\[ \cos(\theta_{1k}) = \vec{u}_{1k} \cdot \vec{v}_{1k} = \vec{u}_{1k}^T \vec{v}_{1k} \]

Figure 3.6: Unperturbed TDOA Computation for \(P_{t1}\) and \(P_{tk}\)
\[\Rightarrow \Delta d_{1k}^2 = 2R_{1k}^2 \left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right) \left[1 \pm \frac{\left(1 - \frac{D_{1k}^2}{4R_{1k}^2}\right)}{\left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right)}\right]\]

\[\Rightarrow \Delta d_{1k}^2 = 2R_{1k}^2 \left(1 + \frac{D_{1k}^2}{4R_{1k}^2}\right) \pm \left(1 - \frac{D_{1k}^2}{4R_{1k}^2}\right)\]

\[\Rightarrow \Delta d_{1k}^2 = \begin{cases} 4R_{1k}^2, & \text{for plus sign} \\ D_{1k}^2, & \text{for minus sign} \end{cases}\]

Clearly, the minus sign is the correct choice.

\[\Rightarrow \Delta d_{1k}^2 = 2R_{1k}^2 \left[1 \pm \left(1 - \alpha\right)\right] \left[1 - \left(1 - \alpha\right)^2\right] \left[1 - \left(1 - \alpha\right)^3\right] \cdots \tag{3.2}\]

Using the results from Section 2.6, the square root can be written as

\[\sqrt{1 - \alpha} = 1 - \frac{1}{2} \alpha - \frac{1}{2!} \alpha^2 - \frac{1}{3!} \alpha^3 - \cdots\]

\[\sqrt{1 - \alpha} = 1 + \sum_{n=1}^{\infty} (-1)^n a_n \alpha^n, \text{ where}\]

\[\alpha = \left[\frac{\left(D_{1k} \cos(\theta_{1k})\right)}{\left(R_{1k}\right)}\right]^2 \left[1 + \frac{D_{1k}^2}{4R_{1k}^2}\right]\]

\[a_0 = 1,\]

\[a_n = (1.5/n - 1) a_{n-1}, \forall n \geq 1\]
Plugging this expansion into Equation 3.2, and then factoring out \( \alpha/2 \),

\[
\Delta d_{1k}^2 = 2R_{1k}^2 \left( 1 + \frac{D_{1k}^2}{4R_{1k}^2} \right) \left[ \left( \frac{1}{2} \right) \alpha + \left( \frac{1}{2! \cdot 4} \right) \alpha^2 + \left( \frac{1}{3! \cdot 8} \right) \alpha^3 + \cdots \right]
\]

\[
\Delta d_{1k}^2 = D_{1k}^2 \cos^2(\theta_{1k}) \left[ \frac{1 + \left( \frac{1}{2! \cdot 2} \right) \alpha + \left( \frac{1}{3! \cdot 2^2} \right) \alpha^2 + \left( \frac{1}{4! \cdot 2^3} \right) \alpha^3 + \cdots}{1 + \frac{D_{1k}^2}{4R_{1k}^2}} \right]
\]

Taking the square root of both sides and recognizing the sign is included in \( \cos(\theta_{1k}) \) results in the alternate form for computing \( \Delta d_{1k} \). This form is used when \( R_{1k} \geq D_{1k} \).

\[
\Delta d_{1k} = D_{1k} \cos(\theta_{1k}) \left[ \left( 1 + S_\alpha \right) \right], \text{ where}
\]

\[
S_\alpha = \left( \frac{1}{2! \cdot 2^1} \right) \alpha + \left( \frac{1}{3! \cdot 2^2} \right) \alpha^2 + \left( \frac{1}{4! \cdot 2^3} \right) \alpha^3 + \left( \frac{1}{5! \cdot 2^4} \right) \alpha^4 + \cdots
\]

\[
S_\alpha = \sum_{n=1}^{N} b_n \alpha^n, \text{ where } b_{N+1} \alpha^{N+1} < \text{eps}^2
\]

\[
\text{eps} = \text{the machine epsilon},
\]

\[
b_0 = 1, \quad b_n = \left( \frac{n - 0.5}{n + 1} \right) b_{n-1}, \forall n \geq 1
\]

\[
\alpha = \left[ \left( \frac{D_{1k} \cos(\theta_{1k})}{R_{1k}} \right) \right]^2 = \left[ \frac{4R_{1k}D_{1k} \cos(\theta_{1k})}{4R_{1k}^2 + D_{1k}^2} \right]^2
\]

Note that \( \alpha < 1 \), when \( \frac{D_{1k}}{R_{1k}} \leq 1 \), and this alternate method for computing \( \Delta d_{1k} \) converges per the Ratio test. The order of mathematical computations effects the
accuracy of results. The sum, $S_\alpha$, is computed prior to adding it to 1 in Equation 3.3. The order of the rightmost term in Equation 3.4 is more accurate than the left. Detailed analyses of roundoff error propagation is beyond the scope of this work.

To summarize, the unperturbed TDOAs, $\Delta t_{1k}$, are computed from the true source position and true perturbed sensor positions as follows.

If $\frac{R_{1k}}{D_{1k}} < 1$, then

$$\Delta d_{1k} = \sqrt{\left( \vec{R}_s - \vec{P}_{t1} \right)^T \left( \vec{R}_s - \vec{P}_{t1} \right) - \sqrt{\left( \vec{R}_s - \vec{P}_{tk} \right)^T \left( \vec{R}_s - \vec{P}_{tk} \right)}}$$

If $\frac{R_{1k}}{D_{1k}} \geq 1$, then

$$\Delta d_{1k} = D_{1k} \cos(\theta_{1k}) \left[ \frac{1 + S_\alpha}{1 + \frac{D_{1k}^2}{4R_{1k}^2}} \right],$$

where

$$S_\alpha = \left( \frac{1}{2! \cdot 2^1} \right) \alpha + \left( \frac{1 \cdot 3}{3! \cdot 2^2} \right) \alpha^2 + \left( \frac{1 \cdot 3 \cdot 5}{4! \cdot 2^3} \right) \alpha^3 + \left( \frac{1 \cdot 3 \cdot 5 \cdot 7}{5! \cdot 2^4} \right) \alpha^4 + \cdots,$$

$$\Rightarrow S_\alpha = \sum_{n=1}^{N} b_n \alpha^n, \quad \text{where } b_{N+1} \alpha^{N+1} < \text{eps}^2,$$

$\text{eps} = \text{the machine epsilon},$

$$b_0 = 1,$$

$$b_n = \left( \frac{n - 0.5}{n + 1} \right) b_{n-1}, \quad \forall n \geq 1$$

$$\alpha = \left[ \frac{4R_{1k}D_{1k} \cos(\theta_{1k})}{4R_{1k}^2 + D_{1k}^2} \right]^2,$$

$$\Delta t_{1k} = \Delta d_{1k}/v$$
### 3.5 TDOA Perturbations

Similar to sensor position knowledge, TDOA measurements are also known with limited accuracy. TDOAs are often determined from cross-correlations of time tagged sensor signals. There are two possible models for TDOA perturbations, a random TDOA perturbation model and a correlated TDOA perturbation model.

The random TDOA perturbation model pertains to low signal to noise ratio (SNR) environments, where the cross correlation peak is heavily impacted by the random noise in two sensor data streams. Perturbations in this model are randomly added to or subtracted from the computed unperturbed TDOAs between each sensor pair.

The correlated TDOA perturbation model applies when the primary error source alters the perceived time of arrival (TOA) at each sensor. This can be caused by clock synchronization errors and can occur in high SNR environments. Perturbations are added to or subtracted from the computed unperturbed TDOAs according to their altered errors in the corresponding TOAs. It is possible for the error term in some perturbed TDOAs to be zero, when the signs of the TOA errors are the same, and twice that of randomly perturbed TDOAs, when the signs are opposite.

One objective of this work is to determine the maximum TDOA errors allowed for a maximum prescribed source location error. The worst case random TDOA perturbations are greater than or equal to those of the correlated TDOA perturbations, if the maximum correlated error term is assumed to be half of the maximum random error term. Therefore, only random TDOA perturbations are used here.
In the random TDOA perturbation model, a maximum TDOA error, $\Delta t_{err}$, is added to or subtracted from each unperturbed TDOA, $\Delta t_{1k}$, to form the corresponding perturbed TDOA, $\Delta \tilde{t}_{1k} = \Delta t_{1k} \pm \Delta t_{err}$, where the + or - sign is randomly chosen with equal probability. The values for $\Delta t_{err}$ start at $1e-24$ seconds, are incremented by factors of 10, and stop at a maximum error, $\Delta t_{max}$. $\Delta t_{max}$ ranges in value from $1e-10$ to $1e-2$ seconds and depends on the sensor distance parameter, $B$, and the source distance from the origin, $R$.

For $N_S = 7$ sensors, there are a maximum of $\binom{N_S}{2} = \binom{7}{2} = 21$ sensor pairs and TDOAs. The number of pairs used in the least squares solution for the random TDOA perturbation model with unperturbed sensors is $N_S - 1 = 6$, the number of pairs that include the reference sensor. When combining TDOA perturbations with sensor perturbations, the number of TDOA pairs used is the same as that used for perturbed sensor solutions.

**TDOA Perturbations for Unperturbed or Relatively Perturbed Sensors**

\[
\Delta \tilde{t}_{1k} = \Delta t_{1k} + \text{sgn}(\rho_{1k}) \times \Delta t_{err}
\]

\[
\Delta t_{1k} = \text{the unperturbed TDOA between } P_1 \text{ and } P_k,
\]

$\Delta t_{err} \in \{1e-24, 1e-23, 1e-22, 1e-21, \ldots, \Delta t_{max} \}$ seconds,

$\Delta t_{max} \in \{1e-10, 1e-9, \ldots, 1e-2\}$ seconds,

\[
\text{sgn}(\rho_{1k}) = \begin{cases} 
-1, & \text{if } \rho_{1k} \leq 0 \\
+1, & \text{if } \rho_{1k} > 0 
\end{cases}
\]

where $\rho_{1k} = \text{rand}[-0.5 : 0.5]$, for $k = 2, 3, \ldots, N_S$, where $N_S = 7$ is the number of sensors.
\[ \Delta \tilde{t}_{ik} = \Delta t_{ik} + \text{sgn}(\rho_{ik}) \times \Delta t_{err} \]

\[ \Delta t_{ik} \] is the unperturbed TDOA between \( P_i \) and \( P_k \),

\[ \Delta t_{err} \in \{ 1e-24, 1e-23, 1e-22, 1e-21, \ldots, \Delta t_{max} \} \] seconds,

\[ \Delta t_{max} \in \{ 1e-10, 1e-9, \ldots, 1e-2 \} \] seconds,

\[ \text{sgn}(\rho_{ik}) = \begin{cases} 
-1, & \text{if } \rho_{ik} \leq 0 \\
+1, & \text{if } \rho_{ik} > 0 
\end{cases} \]

where \( \rho_{ik} = \text{rand}[-0.5 : 0.5] \),

\[ i = 1, 2, \ldots, N_S - 1, \]

\[ k = i, i + 1, \ldots, N_S, \] where \( N_S = 7 \) is the number of sensors

### 3.6 Least Squares Formulation

The least squares solution for the multi-sensor TDOA problem minimizes the sum of products, which are formed from the multiplication of an error with its complex conjugate. Each error is a difference of two expressions for the cosine of the angle between two intersecting line segments.

The first line segment connects the source position and the midpoint of two sensors. The second line segment connects the positions of the same two sensors. One cosine expression is derived from geometry using the dot product as described in Section 2.1. The second cosine expression is derived from the constant TDOA hyperbola using the two sensor positions and the time difference of signal arrival between the
two sensors, as shown in Figure 2.9 of Section 2.4.

Consider two sensors as shown in Figure 3.7. The cosine of the angle of signal arrival is derived from the constant TDOA hyperbola for the sensor pair per Equation 2.11.

\[
\cos(\theta_{1k}) = \frac{x_0}{R_{1k}} = \frac{\Delta d_{1k}}{D_{1k}} \sqrt{1 + \frac{(D_{1k}^2 - \Delta d_{1k}^2)}{4R_{1k}^2}}
\]

The following equations hold for a general translated and rotated 2-sensor line, with sensors located at the positions pointed to by vectors \( \vec{P}_1 \) and \( \vec{P}_k \), and a signal source located at the end of vector \( \vec{S}_0 \).

\[
\vec{M}_{1k} = \left( \frac{\vec{P}_1 + \vec{P}_k}{2} \right) = \text{the vector pointing to the midpoint of the two sensors}
\]

\[
R_{1k} = |\vec{S}_0 - \vec{M}_{1k}| = \text{the distance between the signal source and the midpoint}
\]

\[
D_{1k} = |\vec{P}_k - \vec{P}_1| = \text{the distance between the sensors}
\]

\[
\Delta d_{1k} = d_1 - d_k = |\vec{S}_0 - \vec{P}_1| - |\vec{S}_0 - \vec{P}_k| = v(t_1 - t_k) = v \Delta t_{1k}, \text{where}
\]

\[
\Delta t_{1k} = (t_1 - t_k) = \text{the TDOA between } P_1 \text{ and } P_k, \text{ and}
\]

\[v = \text{the signal speed.}\]

The cosine derived from the constant TDOA hyperbola then becomes

\[
\cos(\theta_{1k}) = \frac{\Delta d_{1k}}{D_{1k}} \sqrt{1 + \frac{(D_{1k}^2 - \Delta d_{1k}^2)}{4R_{1k}^2}}
\]

\[
\Rightarrow \cos(\theta_{1k}) = \frac{\Delta d_{1k}}{|\vec{P}_k - \vec{P}_1|} \sqrt{1 + \frac{(|\vec{P}_k - \vec{P}_1|^2 - \Delta d_{1k}^2)}{4|\vec{S}_0 - \vec{M}_{12}|^2}}
\]
\[ d_1 = |\vec{S}_0 - \vec{P}_1| = \sqrt{\left( \vec{S}_0 - \vec{P}_1 \right)^T \left( \vec{S}_0 - \vec{P}_1 \right)} \]

\[ d_k = |\vec{S}_0 - \vec{P}_k| = \sqrt{\left( \vec{S}_0 - \vec{P}_k \right)^T \left( \vec{S}_0 - \vec{P}_k \right)} \]

\[ \vec{M}_{1k} = \left( \frac{\vec{P}_1 + \vec{P}_k}{2} \right) \]

\[ D_{1k} = |\vec{P}_k - \vec{P}_1| = \sqrt{\left( \vec{P}_k - \vec{P}_1 \right)^T \left( \vec{P}_k - \vec{P}_1 \right)} \]

\[ \vec{u}_{1k} = \left( \vec{P}_k - \vec{P}_1 \right) / D_{1k} \]

\[ R_{1k} = |\vec{S}_0 - \vec{M}_{1k}| = \sqrt{\left( \vec{S}_0 - \vec{M}_{1k} \right)^T \left( \vec{S}_0 - \vec{M}_{1k} \right)} \]

\[ \vec{v}_{1k} = \left( \vec{S}_0 - \vec{M}_{1k} \right) / R_{1k} \]

\[ \cos(\theta_{1k}) = \frac{\vec{u}_{1k} \cdot \vec{v}_{1k}}{\vec{u}_{1k} \cdot \vec{v}_{1k}} \]

Figure 3.7: Least Squares Formulation for \( P_1 \) and \( P_k \)

The cosine of the same angle is formed from geometry using the dot product, as
shown in Equation 2.2.

$$\cos(\theta_{1k}) = \hat{v}_{1k} \cdot \hat{u}_{1k} = \frac{\left(\hat{S}_0 - \hat{M}_{1k}\right) \cdot \left(\hat{P}_k - \hat{P}_1\right)}{|\hat{S}_0 - \hat{M}_{1k}| |\hat{P}_k - \hat{P}_1|}$$

$$\Rightarrow \cos(\theta_{1k}) = \hat{v}_{1k}^T \hat{u}_{1k} = \frac{\left(\hat{S}_0 - \hat{M}_{1k}\right)^T \left(\hat{P}_k - \hat{P}_1\right)}{|\hat{S}_0 - \hat{M}_{1k}| |\hat{P}_k - \hat{P}_1|}$$

To avoid possible division by 0, $\cos(\theta_{1k})$ is multiplied by $R_{1k} = |\hat{S}_0 - \hat{M}_{1k}|$.

$$R_{1k} \cos(\theta_{1k}) = \left(\hat{S}_0 - \hat{M}_{1k}\right)^T \frac{\left(\hat{P}_k - \hat{P}_1\right)}{|\hat{P}_k - \hat{P}_1|} \quad (3.5)$$

$$R_{1k} \cos(\theta_{1k}) = \frac{\Delta d_{1k}}{|\hat{P}_k - \hat{P}_1|} \sqrt{|\hat{S}_0 - \hat{M}_{1k}|^2 + \left(\left|\hat{P}_k - \hat{P}_1\right|^2 - \Delta d_{1k}^2\right)/4} \quad (3.6)$$

For large uncertainties or errors in the sensor position knowledge or TDOA measurements, it is possible for the radicand under the radical in Equation 3.6 to be negative, resulting in complex roots. To avoid halting simulations due to an error arising from a negative radicand and to ensure the solution for the source position is real, the square of the absolute value is used as the cost function to be minimized instead of the square of the potentially complex valued function, in accord with the altered Levenberg-Marquardt algorithm described in Section 2.8. This precaution is unnecessary, if the uncertainties are known or designed to be sufficiently small to avoid complex roots in practical applications. However, for a source near a sensor pair midpoint with large sensor position or TDOA measurement uncertainties, complex roots do appear.
For relative sensor perturbations or for TDOA perturbations with unperturbed sensors, the square of the absolute value of the difference of (3.5) and (3.6) is summed over all pairs that include the reference sensor, arbitrarily chosen to be $P_1$. The least squares objective is then to minimize $F_c(S_0)$ with respect to $S_0$ given $\Delta d_{1k}$, for $k = 2, 3, \ldots, N_S$, where $N_S = 7$, the number of sensors.

$$F_c(S_0) = \frac{1}{2} \sum_{k=2}^{N_S} f_{1k}(S_0) f_{1k}^*(S_0) = \frac{1}{2} \left[ f(S_0) \right]^T \left[ f(S_0) \right]^*, \text{ where}$$

$$f_{1k}(S_0) = \left[ (S_0 - M_{1k})^T \left( \frac{\vec{P}_k - \vec{P}_1}{|\vec{P}_k - \vec{P}_1|} - \frac{\Delta d_{1k}}{|\vec{P}_k - \vec{P}_1|} \sqrt{|S_0 - M_{1k}|^2 + \left( |\vec{P}_k - \vec{P}_1|^2 - \Delta d_{1k}^2 \right)/4} \right] \right]$$

$$f(S_0) = \left[ f_{12}(S_0), f_{13}(S_0), \ldots, f_{1k}(S_0) \right]^T$$

The gradient of $f_{1k}(S_0)$ with respect to $S_0$ is used to form the Jacobian.

$$\nabla_{S_0} f_{1k}(S_0) = \left[ \frac{\vec{P}_k - \vec{P}_1}{|\vec{P}_k - \vec{P}_1|} - \frac{\Delta d_{1k}}{|\vec{P}_k - \vec{P}_1|} \sqrt{|S_0 - M_{1k}|^2 + \left( |\vec{P}_k - \vec{P}_1|^2 - \Delta d_{1k}^2 \right)/4} \right]$$

$$\Rightarrow J_0 = \left[ \nabla_{S_0} f_{12}(S_0), \nabla_{S_0} f_{13}(S_0), \ldots, \nabla_{S_0} f_{1k}(S_0) \right]^T$$

(3.9)
In Equation 3.9 is the Jacobian of $f(\vec{S}_0)$ for the source position starting point, $\vec{S}_0$, used in the first step of the modified Levenberg-Marquardt algorithm. $F_c(\vec{S}_0)$ in Equation 3.7 is the cost function used to determine whether to keep iterative updates and is one of the criteria used to exit the algorithm.

For general sensor perturbations, all $\binom{N_s}{2} = \binom{7}{2} = 21$ sensor pairs are used in the cost function sum and corresponding Jacobian.

### 3.7 Initial Starting Point

Like many non-linear least squares algorithms, the convergence rate for the 3D TDOA geolocation algorithm can be slow. If it does converge, it may not converge to a point near the true source position. Successful convergence to the true source position depends on the number of sensors and their positions, the source location with respect to the sensors, the initial starting point, the convergence threshold, the machine precision, uncertainties in sensor positions and TDOA measurements, and whether all of the available data is simultaneously used in the least squares updates.

The minimum number of sensors required to geolocate a source position depends on known source information and sensor directionality. Directional sensors and source location assumptions can reduce the minimum number of sensors required to locate the source. Most RF (radio frequency) and optical sensors can only detect signals within a limited field of view restricting the source search area. Sonar, acoustic and seismic sensors are generally omni-directional. The signal position sought may be
further assumed to lie in a particular region or on a surface reducing the minimum number of sensors required. With no knowledge about the direction or position of the source, as assumed here, at least five sensors are required to geolocate a signal using only TDOA information. One independent sensor pair is required for each of the four variables, \(x, y, z,\) and \(t\), as described in Section 2.5.

A simple closed form linear system solution was derived in [Bakhoum, 2006] for a 5-sensor TDOA problem, thanks to Ezzat G. Bakhoum’s clever insight to use the difference of the square of the times of arrival. The potential drawback to its use is the fact that the \(A_5\) matrix in Equation 2.15 can be badly conditioned. Using infinite precision math with no TDOA measurement or sensor position errors, an ill or a well conditioned \(A_5\) matrix is determined by the source position with respect to the sensor positions.

### 3.7.1 Symmetric Sensor Position Anomaly

There is an interesting anomaly with the TDOA problem that arises from symmetric sensor positions. Sensor symmetry creates a source position null space. A source located near a null point of symmetrically oriented sensor positions, at which contributions from multiple sensor pairs cancel or nearly cancel, render the \(A_5\) matrix singular or at least badly conditioned. The 5-sensor solution, \(\vec{S}_0\), from the \(A_5\) matrix cannot be used to estimate a source position that is in or near its null space.

Sreeram Potluri, and others, derived a closed-form solution for 3D geolocation of a
signal source position using TDOAs from four known sensor positions in [Potluri, 2002]. Additional information is required to eliminate ambiguity between two possible solutions. For a 5-sensor system, it is sometimes possible to apply a 4-sensor solution to two sets of 4 sensors and use the common solution as the initial starting point, when the 5-sensor solution fails. Unfortunately, the sensor position symmetry anomaly can defeat both methods, as the following example illustrates.

Consider a source, \( R_s \), located at \((0, 0, 2)\), in a 5-sensor system with sensors at the following locations.

\[
P_1 = (0, 0, 0) \\
P_2 = (1, 0, 1) \\
P_3 = (0, 1, 1) \\
P_4 = (-1, 0, 1) \\
P_5 = (0, -1, 1)
\]

The \( A_5 \) matrix for this source and these five sensor positions is singular. In fact, the \( A_5 \) matrix is singular for any source located on the z-axis with sensors at these positions. This can be easily seen in Figure 3.8.
The contribution of $(\vec{P}_5 - \vec{P}_1)$ in the y-direction cancels with that of $(\vec{P}_3 - \vec{P}_1)$. Similarly, the contribution of $(\vec{P}_4 - \vec{P}_1)$ in the x-direction cancels with that of $(\vec{P}_2 - \vec{P}_1)$. Only one cancellation is sufficient to render the $A_5$ matrix singular.

Two sets of 4-sensor solutions are required to eliminate the 4-sensor closed form dual solution ambiguity to obtain the initial starting point estimate with no additional knowledge. With two cancellations from a source position on the z-axis, there is not even one set of 4 sensors from this 5-sensor system that can be used to compute the two potential source solutions using the 4-sensor closed form solution.

Similar to the source position null space, there is also a gradient null space due to sensor symmetry. It is sometimes possible to eliminate one of the two gradient pairs.
causing the cancellation or to use only one of the two pairs in every other least squares update to achieve convergence. Solving special cases with sensor position symmetry nulls is beyond the scope of this work. To avoid the sensor position symmetry nulls and to evaluate the least squares improvement with more than 5 sensors, 7 sensors are assumed here.

### 3.7.2 TDOA Dilution of Precision Metric

The increase in the TDOA geolocation error from input data errors is referred to in literature as “dilution of precision”. It is well known to be dependent on the source and sensor geometry, without concern for symmetry nulls. It is a measure of the error gain, that is, how badly input errors propagate to the source location solution error, and ranges in value from 1 to $\infty$.

Some published metrics that quantify the dilution of precision include moments and products of inertia in [Lee, 1975a] and [Lee, 1975b], the error covariance matrix about the line of sight vector in [Chaffee and Abel, 1994], and the condition number of the visibility matrix in [Mckay and Pachter, 1997]. For a 5-sensor system, we suggest the condition number of the $A_5$ matrix, $\kappa(A_5)$, as yet another. It is only used here to avoid sensor position symmetry nulls and to obtain a suitable starting point for the algorithm’s speedy convergence.

When there are more than 5 sensors in the data collection system, multiple sets of 5 sensors exist, from which a possible starting point could be computed. It is very
unlikely that all sets of 5 sensor positions would suffer from a symmetry null for a
given source position. The choice of which set to use is determined by the condition
number of the corresponding matrix formed from each set of 5 sensor positions and
their corresponding TDOAs.

The solution from the $A_5$ matrix with the minimum condition number is used
as the initial starting point, $\vec{S}_0$. The least squares algorithm uses all information
from the pairs, that include the reference sensor, in every step. In the general sen-
or perturbation model, sensor pairs excluding the reference sensor can be used to
evaluate acceptable starting points, but for the sensors oriented as in Figure 3.2, this
was not required. No effort is made here to determine gradient cancellation pairs or
to restrict their contributions in each least squares update, as may be necessary for
sources located in the null space of symmetric sensor positions.

3.8 Scaling

Scaling is used in the check for satisfaction of the exit criteria. Without scaling,
adjustments in the convergence threshold are necessary for the algorithm to converge
in a timely manner as the source distance varies. The two measures that require
scaling are the magnitude of the solution update, $|\delta \vec{S}_n|$, and the cost function, $F_c(\vec{S}_n)$. The magnitude of $\vec{S}_n$, $|\vec{S}_n|$, is the obvious scaling factor choice for $|\delta \vec{S}_n|$. A proper
scaling factor for $F_c(\vec{S}_n)$ is more subtle.

Referencing Equations 3.5 and 3.6 used in the formation of the cost function at
each step, \( n \), the cosine of the angle of arrival at the midpoint of each sensor pair is multiplied by 
\[
R_{1k,n} = \left| \vec{S}_n - \vec{M}_{1k} \right|. 
\]
This factor avoids division by 0, when the source is near the midpoint of two sensors, and still allows its contribution to the cost function metric. By choosing the maximum of 
\[
R_{1k,n}, \quad R_{\text{max},n} = \max_k (R_{1k,n}), \quad \forall k = 2, 3, \ldots, N_S,
\]
as the scaling factor, both terms of the scaled Equation 3.8, \( f_{1k}(\vec{S}_0)/R_{\text{max},n} \), are no greater than 1. The scaled exit criteria then becomes:

\[
\frac{\left| \delta \vec{S}_n \right|}{\vec{S}_n} < \epsilon,
\]

\[
\frac{F_c(\vec{S}_n)}{R_{\text{max},n}^2} < \epsilon^2,
\]
where

\[
R_{\text{max},n} = \max_k (R_{1k,n}) = \max_k \left| \vec{S}_n - \vec{M}_{1k} \right|, \quad \forall k = 2, 3, \ldots, N_S
\]

\[
N_S = 7, \quad \text{and}
\]

\[
\epsilon = \text{a suitable convergence threshold}
\]

### 3.9 Convergence Threshold

The convergence threshold, \( \epsilon \), effects the speed and accuracy of the least squares solution in much the same way as the machine precision, \( \text{eps} \). There is an optimal range for \( \epsilon \), outside of which either reduces the solution accuracy or increases the number of steps required to reach the exit criteria with no accuracy improvement. The optimal convergence threshold, \( \epsilon_{\text{opt}} \), increases with larger errors in the input data.
To find the optimal convergence threshold, a maximum uncertainty is assumed for the sensor position and TDOA input data. Simulations are run with $\epsilon = (10^k \ast \text{eps})$, where $k \in \{0, 1, 2, \ldots\}$. The maximum least squares solution errors are monitored with each increment in $k$ until the geolocation errors begin to rise at the source distances of interest. The largest value of $k$ achieving the same maximum solution error as for $k = 0$ is chosen as the optimal convergence threshold, $\epsilon_{opt}$. This choice of $\epsilon_{opt}$ generally reduces the maximum number of iterations to reach the least squares solution with all other parameters remaining the same.

### 3.10 Machine Epsilon

The optimal convergence threshold is also an indicator of the maximum machine epsilon, $\text{eps}_{max}$, allowed to achieve the desired geolocation accuracy. The correlation between $\epsilon_{opt}$ and $\text{eps}_{max}$ is somewhat loosely coupled. Executing equations in an optimal order can minimize accumulated round off errors and increase $\text{eps}_{max}$ without effecting $\epsilon_{opt}$.

The maximum machine epsilon is estimated to be an order of magnitude smaller than the optimal convergence threshold, $\Rightarrow \text{eps}_{max} = \epsilon_{opt}/10$. This estimate is consistent with the new criteria used for the gradient gain and damping parameter updates and the upper limit imposed on the condition number of matrices at intermediate steps in the LM algorithm. This choice may be conservative and larger values for the maximum machine epsilon may be possible without loss of accuracy.
Using the absolute maximum machine precision is unnecessary for potential improvements and provides no room for margin. Depending on the hardware implementation, $\varepsilon_{max}$ values within an order of magnitude of the absolute maximum can still reduce power and resource consumption without loss of geolocation accuracy at the source distances of interest. Simulations confirming the maximum machine epsilon possible are beyond the scope of this work.
Chapter 4

Simulations

This chapter describes simulations using the models of Chapter 3 and the tools of Chapter 2. For separate sensor and TDOA perturbations, 100 random source directions and 20 random sensor or TDOA perturbations were used. For combined sensor and TDOA perturbations, the number of random sensor and TDOA perturbations were increased to 100. Although these sample sizes are sufficient to illustrate the results, a higher number of random samples are recommended to determine the highest maximum possible error.

4.1 Relative Sensor Perturbations

Simulations for relative sensor perturbations with unperturbed TDOAs use:

- 100 source positions, $\hat{R}_s, s = 1, \ldots, 100$, described in Section 3.2;
- 7 original unperturbed sensor positions, $\hat{P}_k, k = 1, \ldots, 7$, as shown in Figure 3.2 and described in Section 3.3;
− 20 random rotation angles, \( \eta_{kj}, j = 1, \ldots, 20 \), for each perturbation angle, \( \xi \), and for each of the 6 non-reference frame sensor positions, \( \hat{P}_k, k = 2, \ldots, 7 \), described in Subsection 3.3.1;
− 6 unperturbed TDOA computations, \( \Delta t_{ik}, k = 2, \ldots, 7 \), described in Section 3.4;
− 6 sensor pairs, \( (\hat{P}_1, \hat{P}_k), k = 2, \ldots, 7 \), in the least squares gradient updates, described in Section 3.6;
− the initial starting point, \( \hat{S}_0 \), described in Section 3.7.2;
− the scaling factors, described in Section 3.8;
− the convergence threshold, \( \epsilon = 1e0^*\text{eps} \), described in Section 3.9; and,
− the modified Levenberg-Marquardt algorithm, described in Section 2.8.

A separate plot of maximum least squares solution position errors (LSSPEs) for relative sensor perturbations with unperturbed TDOAs is generated for each sensor distance parameter, \( B \). For each value of \( B \in \{1e0, 1e1, \ldots, 1e8\} \) meters, the maximum LSSPE, \( \max_{j,s} \left[ \log_{10} \| \hat{S}_{LS}(\eta_{kj}) - \hat{R}_s \| \right] \), is plotted on the same graph for each relative sensor perturbation angle, \( \xi \in \{0, 1e-15, 1e-14, \ldots, 1e-1\} \) radians, and for each source distance parameter, \( R \in \{1e-2, 1e-1, \ldots, S_{max}\} \) meters.

\( S_{max} \) is the maximum value of \( R \) that produces a stable solution. \( S_{max} \) depends on the values \( B \) and \( \xi \) and ranges in value from 1e1 to 1e21 meters. Simulation results for the maximum LSSPEs for relative sensor perturbations with unperturbed TDOAs are shown Appendix A. In the graph ordinate labels, \( P_s^* \) is the location pointed to by the vector \( \hat{S}_{LS}(\eta_{kj}) \), and \( P_s \) is the location pointed to by the vector \( \hat{R}_s \).
4.2 General Sensor Perturbations

Simulations for general sensor perturbations with unperturbed TDOAs use:

- 100 source positions, $\hat{R}_s, s = 1, \ldots, 100$, described in Section 3.2;
- 7 original unperturbed sensor positions, $\hat{P}_k, k = 1, \ldots, 7$, as shown in Figure 3.2 and described in Section 3.3;
- 20 random directional cosine vectors, $\hat{\varphi}_{kj}, j = 1, \ldots, 20$, for each perturbation angle, $\xi$, and for each of the 7 sensor positions, $\hat{P}_k, k = 1, \ldots, 7$, described in Subsection 3.3.2;
- 21 unperturbed TDOA computations, $\Delta t_{ik}, i = 1, \ldots, 6, k = i, \ldots, 7$, described in Section 3.4;
- 21 sensor pairs, $(\hat{P}_i, \hat{P}_k), i = 1, \ldots, 6, k = 2, \ldots, 7$, in the least squares gradient updates described in Section 3.6;
- the initial starting point, $\hat{S}_0$, described in Section 3.7.2;
- the scaling factors, described in Section 3.8;
- the convergence threshold, $\epsilon = 1e0^*\epsilon$, described in Section 3.9; and,
- the modified Levenberg-Marquardt algorithm, described in Section 2.8.

A separate plot of maximum least squares solution position errors (LSSPEs) for general sensor perturbations with unperturbed TDOAs is generated for each sensor distance parameter, $B$, in the same manner as for the relative sensor perturbations. For each value of $B \in \{1e0, 1e1, \ldots, 1e8\}$ meters, the maximum LSSPE,
max \left[ \log_{10} \left| \tilde{S}_{LS}(\varphi_{kj}) - \tilde{R}_s \right| \right], is plotted on the same graph for each relative sensor perturbation angle, \( \xi \in \{0, 1e-15, 1e-14, \ldots, 1e-1 \} \) radians, and for each source distance parameter, \( R \in \{1e-2, 1e-1, \ldots, S_{\text{max}} \} \) meters.

Simulation results for the maximum LSSPEs for general sensor perturbations with unperturbed TDOAs are shown in Appendix B. In the graph ordinate labels, \( P_s^* \) is the location pointed to by the vector \( \tilde{S}_{LS}(\varphi_{kj}) \), and \( P_s \) is the location pointed to by the vector \( \tilde{R}_s \).

### 4.3 TDOA Perturbations

Simulations for TDOA perturbations with unperturbed sensors use:

- 100 source positions, \( \tilde{R}_s, s = 1, \ldots, 100 \), described in Section 3.2;
- 7 original unperturbed sensor positions, \( \tilde{P}_k, k = 1, \ldots, 7 \), as shown in Figure 3.2 and described in Section 3.3;
- 6 unperturbed TDOA computations, \( \Delta t_{1k}, k = 2, \ldots, 7 \), described in Section 3.4;
- 20 randomly perturbed TDOAs, \( \Delta \tilde{t}_{1k,j}, j = 1, \ldots, 20 \), for each of the 6 unperturbed TDOAs, \( \Delta t_{1k}, k = 2, \ldots, 7 \), described in Section 3.5;
- 6 sensor pairs, \((\tilde{P}_1, \tilde{P}_k), k = 2, \ldots, 7\), in the least squares gradient updates, described in Section 3.6;
- the initial starting point, \( \tilde{S}_0 \), described in Section 3.7.2;
- the scaling factors, described in Section 3.8;
- the convergence threshold, \( \epsilon = 1e0^*\text{\texttt{eps}} \), described in Section 3.9; and,
the modified Levenberg-Marquardt algorithm, described in Section 2.8.

A separate plot of maximum least squares solution position errors (LSSPEs) for TDOA perturbations with unperturbed sensors is generated for each sensor distance parameter, \( B \). For each value of \( B \in \{1e0, 1e1, \ldots, 1e8\} \) meters, the maximum LSSPE, \( \max_{j,s} \left[ \log_{10} \left| \vec{S}_{LS}(\Delta\tilde{t}_{1k,j}) - \vec{R}_s \right| \right] \), is plotted on the same graph for each max TDOA perturbation error, \( \Delta t_{err} \in \{0, 1e-24, 1e-23, \ldots, \Delta t_{max}\} \) seconds, and for each source distance parameter, \( R \in \{1e-2, 1e-1, \ldots, S_{max}\} \) meters.

\( \Delta t_{max} \) depends on \( B \) and ranges in value from 1e-10 to 1e-2 seconds. \( S_{max} \) is the maximum value of \( R \) that produces a stable solution. \( S_{max} \) depends on the values \( B \) and \( \Delta t \) and ranges in value from 1e1 to 1e21 meters. Simulation results for the maximum LSSPEs for TDOA perturbations with unperturbed sensors are shown in Appendix C. In the graph ordinate labels, \( P^*_s \) is the location pointed to by the vector \( \vec{S}_{LS}(\Delta\tilde{t}_{1k,j}) \), and \( P_s \) is the location pointed to by the vector \( \vec{R}_s \).

### 4.4 Requirements Specification Examples

A couple of system engineering requirements specification examples illustrate applications of the simulation results for either a feasibility assessment or a practical implementation.
4.4.1 Example 1

This is a relative sensor perturbation requirements specification example, in which the reference frame is attached to the middle sensor. Suppose a satellite has six 10-meter booms oriented as shown in Figure 3.2. A sensor is mounted on the satellite at the center of the booms and at the end of each boom of length $B = 10$ meters. It is desired to locate a source relative to the satellite at a distance of $R = 100$ km with a maximum error of 5 km.

In order to meet the desired location error, to the nearest order of magnitude:
1a) What is the maximum relative sensor position angle uncertainty required?
1b) What is the maximum TDOA uncertainty required?
1c) What is the optimal convergence threshold?
1d) What is the approximate maximum machine epsilon allowed?
1e) What is the accuracy improvement in the maximum least squares solution position error over the maximum closed form solution position error?
1f) What is the maximum number of iterations required to locate the source?

From Figure A.2, for sensors oriented as shown in Figure 3.2 and located a distance of $B = 10$ meters apart, the maximum least squares solution position error of a source located at a distance of $R = 100$ km, with a maximum relative sensor perturbation of $\xi = 1e-6$ radians and no TDOA errors, is $10^{3.271} = 1.87$ km. \( \Rightarrow \xi = 1e-6 \) radians is chosen as the initial estimate for the maximum sensor position angle uncertainty.

From Figure C.2, for sensors oriented as shown in Figure 3.2 and located a distance of $B = 10$ meters apart, the maximum least squares solution position error of a source
located at a distance of \( R = 100 \) km, with a maximum TDOA perturbation of \( \Delta t_{\text{err}} = 1\times10^{-14} \) seconds and no sensor perturbations, is \( 10^{3.083} \times 1.21 \) km. \( \Rightarrow \Delta t_{\text{err}} = 1\times10^{-14} \) seconds (10 femtoseconds) is chosen as the initial estimate for the maximum TDOA position uncertainty.

The sum of these two geolocation errors, assuming they are highly correlated, produces the estimate of 3.08 km. TDOA nonlinearities require simulations with both sensor and TDOA errors to confirm the initial estimates meet the desired accuracy.

To estimate the maximum convergence threshold used in the LM algorithm, without loss of geolocation accuracy, simulations were run with convergence thresholds of \( \epsilon = 10^k \times \text{eps} \), for \( k = 0, 1, \ldots, 8 \), for relative sensor perturbations with unperturbed TDOAs and for TDOA perturbations with unperturbed sensors. The results are shown in Table 4.1 and Table 4.2.

Table 4.1 highlights show the maximum least squares solution position errors (LSSPEs) for sensor perturbations with no TDOA errors is constant through \( \epsilon = 1\times10^7 \times \text{eps} \) and degrades at \( \epsilon = 1\times10^8 \times \text{eps} \). Table 4.2 highlights show the maximum LSSPEs for TDOA perturbations with no sensor position errors degrades at \( \epsilon = 1\times10^6 \times \text{eps} \). For separate sensor and TDOA perturbations, the maximum convergence threshold should be no greater than about \( \epsilon = 1\times10^5 \times \text{eps} \). When combining errors, a larger convergence threshold may be possible.
Table 4.1: Max Log$_{10}$ of LSSPEs for Example 1 Relative Sensor Perturbations
Convergence Threshold ($\epsilon$) vs Source Distance (R)
($\xi = 1e-6$ radians, $\Delta t_{err} = 0$, $B = 1e1$ meters)

<table>
<thead>
<tr>
<th>Convergence Threshold ($\epsilon$)</th>
<th>Source Distance (R) in meters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1e0</td>
</tr>
<tr>
<td>1e0 * eps</td>
<td>-5.88</td>
</tr>
<tr>
<td>1e1 * eps</td>
<td>-5.88</td>
</tr>
<tr>
<td>1e2 * eps</td>
<td>-5.88</td>
</tr>
<tr>
<td>1e3 * eps</td>
<td>-5.88</td>
</tr>
<tr>
<td>1e4 * eps</td>
<td>-5.88</td>
</tr>
<tr>
<td>1e5 * eps</td>
<td>-5.88</td>
</tr>
<tr>
<td>1e6 * eps</td>
<td>-5.88</td>
</tr>
<tr>
<td>1e7 * eps</td>
<td>-5.83</td>
</tr>
<tr>
<td>1e8 * eps</td>
<td>-5.55</td>
</tr>
</tbody>
</table>

Table 4.2: Max Log$_{10}$ of LSSPEs for Example 1 TDOA Perturbations
Convergence Threshold ($\epsilon$) vs Source Distance (R)
($\xi = 0$ radians, $\Delta t_{err} = 1e-14$ seconds, $B = 1e1$ meters)

<table>
<thead>
<tr>
<th>Convergence Threshold ($\epsilon$)</th>
<th>Source Distance (R) in meters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1e0</td>
</tr>
<tr>
<td>1e0 * eps</td>
<td>-5.241</td>
</tr>
<tr>
<td>1e1 * eps</td>
<td>-5.241</td>
</tr>
<tr>
<td>1e2 * eps</td>
<td>-5.241</td>
</tr>
<tr>
<td>1e3 * eps</td>
<td>-5.241</td>
</tr>
<tr>
<td>1e4 * eps</td>
<td>-5.241</td>
</tr>
<tr>
<td>1e5 * eps</td>
<td>-5.241</td>
</tr>
<tr>
<td>1e6 * eps</td>
<td>-5.241</td>
</tr>
<tr>
<td>1e7 * eps</td>
<td>-5.242</td>
</tr>
<tr>
<td>1e8 * eps</td>
<td>-5.195</td>
</tr>
</tbody>
</table>

Table 4.3 shows the maximum least squares solution position error results for
convergence thresholds of $\epsilon = 10^k \times \text{eps}$, for $k = 4, 5, 6 \& 7$ versus distance of the
source $R$ from the satellite. Highlights in this table show minor loss of accuracy at distances of 10 and 100 meters from the satellite beginning at $\epsilon = 1e6*\text{eps}$. Assuming distances less than 1 km from the satellite are not of interest, the optimal convergence threshold is $\epsilon_{opt} = 1e6*\text{eps}$.

Table 4.3: Max $\log_{10}$ of LSSPEs for Example 1 Sensor & TDOA Perturbations

<table>
<thead>
<tr>
<th>Convergence Threshold ($\epsilon$)</th>
<th>Source Distance (R) in meters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1e0</td>
</tr>
<tr>
<td>1e4 * \text{eps}</td>
<td>-5.16739</td>
</tr>
<tr>
<td>1e5 * \text{eps}</td>
<td>-5.16739</td>
</tr>
<tr>
<td>1e6 * \text{eps}</td>
<td>-5.16739</td>
</tr>
<tr>
<td>1e7 * \text{eps}</td>
<td>-5.16738</td>
</tr>
</tbody>
</table>

Simulation results for a convergence threshold of $\epsilon = 1e6*\text{eps}$, a maximum relative sensor uncertainty angle of $\xi = 1e-6$ radians, and a maximum TDOA error of $\Delta t_{err} = 1e-14$ seconds are shown in Figures D.1, D.2 and D.3. The number of random perturbation errors were increased from 20 to 100 for the combined simulations in order to obtain a better estimate of the maximum position error.

Figure D.1 plots the maximum of the $\log_{10}$ of the least squares solution position errors vs the source distance from the satellite. Figure D.2 plots the $\log_{10}$ of the ratio of the maximum closed form solution position error over the maximum least squares solution position error vs the source distance from the satellite. Figure D.3 plots the maximum number of least squares iterations it took to reach the solution vs the source distance from the satellite.
From Figure D.1, the maximum geolocation error for a source that is $R = 100$ km from the satellite is $10^{3.458} = 2.87$ km. Per Section 3.9, the maximum machine epsilon is estimated to be $\epsilon_{max} = \epsilon_{opt}/10 = 1e5*\epsilon_{opt}$, where $\epsilon_{opt} = 2^{-52} \approx 2.22e-16$. 

$\Rightarrow \epsilon_{max} = 2.22e-11 \approx 2^{-35}$. This estimate is conservative and a larger machine epsilon may be possible.

From Figure D.2, the expected least squares improvement in the maximum error over the closed form solution is a factor of $10^{0.3409} = 2.19$. From Figure D.3, the expected maximum number of iterations required to geolocate a source 100 km from the satellite is 212.

In summary and answer to the original questions, the systems engineering requirements to satisfy the desired geolocation accuracy given in Example 1 are:

1a) The maximum relative sensor position angle uncertainty required is $\xi = 1e^{-6}$ radians.
1b) The maximum TDOA uncertainty required is $\Delta t_{err} = 1e^{-14}$ seconds.
1c) The optimal convergence threshold is $\epsilon_{opt} = 2.22e-10$.
1d) The maximum machine epsilon allowed is $\epsilon_{max} = 2^{-35}$.
1e) The accuracy improvement in maximum position error using the least squares solution over the closed form solution is 2.19.
1f) The maximum number of least squares iterations expected is 212.

The maximum closed form solution geolocation error is $2.19 \times 2.87 = 6.29$ km. Using the least squares solution, the maximum relative sensor position angle uncertainty and the maximum TDOA uncertainty can be increased slightly and still meet
the maximum 5 km geolocation error. The maximum machine epsilon uses 17 fewer mantissa bits than the double precision floating point standard.

Comparing Figure D.1 with Figure A.2, the TDOA errors swamp out the errors for relative sensor perturbations for a source distance less than $B$. Comparing Figure B.2 with Figure A.2, the maximum error in the floating reference frame is $10^{3.435}/10^{3.271} = 1.46$ times larger than the maximum error in the fixed reference frame for a source distance of 100 km. This increase is expected for the ratio of absolute error over relative error.

### 4.4.2 Example 2

This is a general sensor perturbation requirements specification example, in which the reference frame is not attached to any sensor. Suppose 7 satellites are oriented as shown in Figure 3.2. The distance between the unperturbed center satellite and the other unperturbed sensors is $B = 100$ km. It is desired to locate a source in a reference frame, that is not attached to any satellite position, at a distance of $R = 1,000$ km from the unperturbed center satellite position with a maximum error of 50 meters.

In order to meet the desired location error, to the nearest order of magnitude:

2a) What is the maximum relative sensor position angle uncertainty required?
2b) What is the maximum TDOA uncertainty required?
2c) What is the optimal convergence threshold?
2d) What is the approximate maximum machine epsilon allowed?
2e) What is the accuracy improvement in the maximum least squares solution position error over the maximum closed form solution position error?

2f) What is the maximum number of iterations required to locate the source?

From Figure B.6, for sensors oriented as shown in Figure 3.2 and located a distance of $B = 100$ km apart, the maximum least squares solution position error of a source located at a distance of $R = 1,000$ km, with a maximum relative sensor perturbation of $\xi = 1e-6$ radians and no TDOA errors, is $10^{1.397} = 25.0$ m. $\xi = 1e-6$ radians is chosen as the initial estimate for the maximum relative sensor perturbation angle. Per Subsection 3.3.2, $\xi = 1e-6$ radians corresponds to a maximum positional error of $\varrho = B \times \xi / 2 = 1e5 \times 1e-6 / 2 = 0.05$ meters = 5 cm for all sensors.

From Figure C.6, for sensors oriented as shown in Figure 3.2 and located a distance of $B = 100$ km apart, the maximum least squares solution position error of a source located at a distance of 1,000 km, with a maximum TDOA perturbation of $\Delta t_{err} = 1e-10$ seconds and no sensor perturbations, is $10^{1.068} = 11.7$ m. $\Delta t_{err} = 1e-10$ seconds (100 picoseconds) is chosen as the initial estimate for the maximum TDOA position uncertainty.

The sum of these two geolocation errors, assuming they are highly correlated, produces the estimate of 36.7 m. TDOA nonlinearities require simulations with both sensor and TDOA errors to confirm the initial estimates meet the desired accuracy.

To estimate the maximum convergence threshold used in the LM algorithm, without loss of geolocation accuracy, simulations were run with convergence thresholds of
\[ \epsilon = 10^k \epsilon, \text{ for } k = 0, 1, \ldots, 9, \text{ for general sensor perturbations with unperturbed TDOAs and for } k = 0, 1, \ldots, 7, \text{ for TDOA perturbations with unperturbed sensors.} \]

The results are shown in Table 4.4 and Table 4.5.

Table 4.4 highlights show maximum least squares solution position errors for sensor perturbations with no TDOA errors degrade for short distances from the reference frame at \( \epsilon = 1e6\epsilon \), but remain constant at the larger distances of interest until \( \epsilon = 1e8\epsilon \). Table 4.5 highlights show maximum least squares solution position errors for TDOA perturbations with no sensor position errors degrade at \( \epsilon = 1e6\epsilon \) for the distance of interest. For separate sensor and TDOA perturbations, the maximum convergence threshold should be about \( \epsilon = 1e5\epsilon \). When combining errors, a larger threshold may be possible.

Table 4.4: Max Log_{10} of LSSPEs for Example 2 General Sensor Perturbations

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{Convergence Threshold (\( \epsilon \))} & \text{Source Distance (R) in meters} \\
\hline
& 1e0 & 1e1 & 1e2 & 1e3 & 1e4 & 1e5 & 1e6 \\
\hline
1e0 \epsilon & -1.07 & -1.07 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e1 \epsilon & -1.07 & -1.07 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e2 \epsilon & -1.07 & -1.07 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e3 \epsilon & -1.07 & -1.07 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e4 \epsilon & -1.07 & -1.07 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e5 \epsilon & -1.07 & -1.07 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e6 \epsilon & -1.041 & -1.07 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e7 \epsilon & -1.04 & -1.04 & -1.07 & -1.07 & -1.106 & -0.73 & 1.397 \\
1e8 \epsilon & -1.04 & -1.04 & -0.963 & -1.07 & -1.106 & -0.729 & 1.398 \\
1e9 \epsilon & -1.041 & -1.04 & -0.963 & -0.914 & -1.086 & -0.653 & 1.405 \\
\hline
\end{array}
\]
Table 4.5: Max $\log_{10}$ of LSSPEs for Example 2 TDOA Perturbations

**Convergence Threshold ($\epsilon$) vs Source Distance ($R$)**

$(\xi = 0 \text{ radians}, \Delta t_{err} = 1e-10 \text{ seconds}, B = 1e5 \text{ meters})$

<table>
<thead>
<tr>
<th>Convergence Threshold ($\epsilon$)</th>
<th>Source Distance ($R$) in meters</th>
<th>1e0</th>
<th>1e1</th>
<th>1e2</th>
<th>1e3</th>
<th>1e4</th>
<th>1e5</th>
<th>1e6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1e0 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.242</td>
<td>-1.242</td>
<td>-1.241</td>
<td>-1.027</td>
<td>0.9518</td>
</tr>
<tr>
<td>1e1 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.242</td>
<td>-1.242</td>
<td>-1.241</td>
<td>-1.027</td>
<td>0.9518</td>
</tr>
<tr>
<td>1e2 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.242</td>
<td>-1.242</td>
<td>-1.241</td>
<td>-1.027</td>
<td>0.9518</td>
</tr>
<tr>
<td>1e3 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.242</td>
<td>-1.242</td>
<td>-1.241</td>
<td>-1.027</td>
<td>0.9518</td>
</tr>
<tr>
<td>1e4 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.242</td>
<td>-1.242</td>
<td>-1.241</td>
<td>-1.027</td>
<td>0.9518</td>
</tr>
<tr>
<td>1e5 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.242</td>
<td>-1.242</td>
<td>-1.241</td>
<td>-1.027</td>
<td>0.9518</td>
</tr>
<tr>
<td>1e6 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.219</td>
<td>-1.242</td>
<td>-1.241</td>
<td>-1.028</td>
<td>0.9520</td>
</tr>
<tr>
<td>1e7 * $\epsilon$</td>
<td></td>
<td>-1.066</td>
<td>-1.066</td>
<td>-1.065</td>
<td>-1.218</td>
<td>-1.241</td>
<td>-1.034</td>
<td>0.9531</td>
</tr>
</tbody>
</table>

Table 4.6 shows the maximum least squares solution position error results for convergence thresholds of $\epsilon = 10^k * \epsilon$, for $k = 4, 5, \ldots, 10$, versus $R$ for combined perturbations. The highlights in this table show minor loss of accuracy at distances of 1 and 10 meters from the satellite beginning at $\epsilon = 1e6*\epsilon$. Loss of precision extends to distances of 100 meters from the satellite at $\epsilon = 1e7*\epsilon$, and to distances of 1 km at $\epsilon = 1e8*\epsilon$. Assuming distances less than 10 km from the satellite are not locations for a source of interest, the optimal convergence threshold is $\epsilon_{opt} = 1e8*\epsilon$. 

90
Simulation results for a convergence threshold of $\epsilon = 1e8*\text{eps}$, a maximum relative sensor uncertainty angle of $\xi = 1e-6$ radians, and a maximum TDOA error of $\Delta t_{err} = 1e-10$ seconds are shown in Figures D.4, D.5 and D.6. The number of random perturbation errors were increased from 20 to 100 for the combined simulations in order to obtain a better estimate of the maximum position error.

Figure D.4 plots the maximum of the log$_{10}$ of the least squares solution position errors vs the source distance from the satellite. Figure D.5 plots the log$_{10}$ of the ratio of the maximum closed form solution position error over the maximum least squares solution position error vs the source distance from the satellite. Figure D.6 plots the maximum number of least squares iterations it took to reach the solution vs the source distance from the satellite.

From Figure D.4, the maximum geolocation error for a source that is $R = 1,000$ km from the satellite is $10^{1.594} = 39.3$ meters. Per Section 3.9, the maximum machine
epsilon is estimated to be \( \text{eps}_{\text{max}} = \epsilon_{\text{opt}}/10 = 1e7^*\text{eps} \), where \( \text{eps} = 2^{-52} \approx 2.22e-16 \).

\[ \Rightarrow \text{eps}_{\text{max}} = 2.22e-9 \approx 2^{-29}. \] 
This estimate is conservative and a larger machine epsilon may be possible.

From Figure D.5, the expected least squares improvement in the maximum position error over the closed form solution is a factor of \( 10^{0.1403} = 1.38 \). From Figure D.6, the expected maximum number of iterations required to geolocate a source 1,000 km from the satellite is 42.

In summary and answer to the original questions, the systems engineering requirements to satisfy the desired geolocation accuracy given in Example 2 are:

2a) The maximum sensor position angle uncertainty required is \( \xi = 1e-6 \) radians.
2b) The maximum TDOA uncertainty required is \( \Delta t_{\text{err}} = 1e-10 \) seconds.
2c) The optimal convergence threshold is \( \epsilon_{\text{opt}} = 2.22e-8 \).
2d) The maximum machine epsilon allowed is \( \text{eps}_{\text{max}} = 2^{-29} \).
2e) The accuracy improvement in maximum position error using the least squares solution over the closed form solution is 1.38.
2f) The maximum number of least squares iterations expected is 42.

The maximum closed form solution geolocation error is \( 1.38*39.3 = 54.2 \) m. Using the least squares solution, the maximum relative sensor position angle uncertainty and the maximum TDOA uncertainty can be increased slightly and still meet the maximum 50 m geolocation error. The maximum machine epsilon uses 23 fewer mantissa bits than the double precision floating point standard.
Chapter 5

Summary

This chapter summarizes results of the present work and offers suggestions for possible future research pertaining to the geolocation of a signal source using time differences of arrival.

5.1 Results

The results of this work can be used to determine the absolute maximum error allowed in sensor position errors and in TDOA errors between sensor pairs to achieve a desired geolocation accuracy, knowing only the sensor positions and the TDOAs between sensor pairs. Similarly, given a maximum TDOA error, or a maximum sensor position error, or both, the results can be used to determine the absolute maximum geolocation accuracy possible for a sensor system. This may help to define requirements or assess their feasibility for a TDOA system.
5.1.1 Geometric Approach

This work presents a geometric approach for nonlinear least squares solutions of a signal source location. Nonlinear least squares minimizations locate a signal source position in three dimensions using only the time difference of arrival between multiple sensor pairs. There is no restriction on the maximum number of sensors used. No additional information is assumed, other than known sensor positions and TDOAs between sensor pairs. The source position originates from over $4\pi$ steradians without any direction or range assumption or knowledge.

5.1.2 Sensor Position Anomaly

From this geometric approach, a symmetric sensor position anomaly is revealed that defeats geolocations of a signal source lying within the null space using 4 and 5 sensor closed form solutions. To avoid the symmetric sensor position anomaly and to evaluate least squares improvements with more than 5 sensors, simulations include 7 sensors. A non-anomalous, 5-sensor set is extracted using a dilution of precision metric.

5.1.3 Dilution of Precision Metric

The condition number of Bakhoum’s 5-sensor closed form solution matrix is used as a new TDOA dilution of precision metric for a 5-sensor system. The minimum condition number determines the 5-sensor set from the given 7 sensor positions and
TDOAs between each sensor pair. Bakhoum’s 5-sensor closed form solution for the chosen 5-sensor set is the initial starting point to the least squares algorithm.

### 5.1.4 Modifications to LM Algorithm

Residual geolocation errors reach the numerical limits of machine precision. To reach the limits of machine precision, the Levenberg-Marquardt (LM) algorithm is modified to solve the nonlinear least squares minimization problem. Modifications include new gradient gain adjustments, damping parameter limiters and restrictions on the upper bound of the condition number of matrices at intermediate steps of the algorithm. A complex form of the LM algorithm allows inclusion of relatively large sensor position and TDOA uncertainties for signal sources originating near sensor pair midpoints.

### 5.1.5 TDOA Computation Improvements

TDOAs are typically computed from the difference of square roots. This method suffers in accuracy at larger source distances from the sensors. TDOA computations for larger source distances apply the binomial theorem with fractional exponents to achieve improvements in accuracy by as much as 8 orders of magnitude for double precision floating point numbers. This alternate method for computing TDOAs, along with modifications to the LM algorithm, helps to reach the limits of machine precision.
5.1.6 Optimal Convergence Times

Scaling avoids threshold adjustments with variations in the source distance from the sensors, which are otherwise necessary for reasonable convergence times. Optimal thresholds minimize the number of steps required for algorithm convergence without loss of geolocation accuracy. The optimal convergence threshold is the maximum convergence threshold that does not increase the maximum geolocation error for given upper bounds on input data uncertainties.

5.1.7 Maximum Machine Epsilon

A larger machine epsilon generally reduces consumed power, required resources and geolocation accuracy. For nonzero sensor position and TDOA input data errors, it is possible to increase the machine epsilon without loss of geolocation accuracy. The maximum machine epsilon is estimated to be one-tenth of the optimal convergence threshold, consistent with the new criteria used for the damping parameter and matrix condition number limiters in the modified LM algorithm.

5.1.8 Frames of Reference Models

Least squares solutions are generated for sensor perturbations with no TDOA errors, for TDOA perturbations with no sensor errors, and for perturbations in both sensor positions and TDOAs. Sensor perturbation models are based on two different frames of reference. The first frame of reference, referred to as fixed reference frame or
relative sensor perturbations, is attached to a sensor and uses only sensor pairs that include the sensor attached to the reference frame in the LM algorithm updates. The second frame of reference, referred to as floating reference frame or general sensor perturbations, is not attached to any sensor and uses all sensor pairs in the LM algorithm updates.

5.1.9 Fixed Reference Frame Results

Plots of the maximum least squares solution position errors (LSSPEs) for unperturbed TDOAs and various levels of sensor position uncertainty in a reference frame attached to a sensor are shown in Appendix A. The maximum source geolocation errors increase at a rate that is proportional to the source distance from the reference frame origin that are less than about half of the distance between sensors. The maximum errors increase at a rate that is proportional to the square of the source distance from the reference frame origin for source distances greater than about half of the distance between sensors.

5.1.10 Floating Reference Frame Results

Plots of the maximum least squares solution position errors (LSSPEs) for unperturbed TDOAs and various levels of sensor position uncertainty in a reference frame not attached to a sensor are shown in Appendix B. The maximum source geolocation errors remain about the same for all source distances from the reference frame origin
that are less than about half of the distance between sensors. The maximum errors increase at a rate that is proportional to the square of the source distance from the reference frame origin for source distances greater than about half of the distance between sensors.

5.1.11 TDOA Perturbation Results

Plots of the maximum least squares solution position errors (LSSPEs) for unperturbed sensor positions and various levels of TDOA uncertainty are shown in Appendix C. The maximum source geolocation errors remain about the same for all source distances from the reference frame origin that are less than about half of the distance between sensors. The maximum errors increase at a rate that is proportional to the square of the source distance from the reference frame origin for source distances greater than about half of the distance between sensors.

5.1.12 Sensor Position & TDOA Perturbation Results

Examples illustrate a procedure to use the numerical results to assess feasibility or specify requirements for a desired geolocation accuracy. Example 1 combines sensor position and TDOA perturbations in a fixed reference frame. Example 2 combines sensor position and TDOA perturbations in a floating reference frame. Plots of the maximum least squares solution position errors (LSSPEs) for the combined perturbations for each example are shown in Appendix D.
5.2 Future Research

An example disclosed the fact that 4 and 5 sensor closed form solutions can be defeated from locating a signal source in 3D, when it lies within the null space of symmetrically oriented sensors. If sensor position symmetry arises, gradient cancellations can be avoided with iterative techniques.

Some sensors can determine a signal’s angle of arrival (AOA). Velocity estimates of source or sensor motion and Doppler shifts at radio frequency (RF) receivers can be used in frequency difference of arrival (FDOA) refinements. Atmospheric effects can alter the signal speed in different signal paths. The time of arrival (TOA) and the received signal strength (RSS) can be computed at each sensor location. Including one or more of these sources of information may improve results in some applications. A similar approach may help define requirements for additional parameters.

Running a simulation using custom machine precisions would determine how close the maximum machine epsilon could approach the optimal convergence threshold and provide a more accurate basis for its estimate. This may also be of use in implementations of general nonlinear least squares minimization problems.

An analysis of the nonlinear geometric error constructed here could confirm the constant, linear and squared relations of the source distance from the origin with least squares solution position errors arising from limitations on the knowledge of sensor positions, TDOA measurements and numerical machine precision in the floating and fixed reference frames.
Other least squares descent techniques may improve the rate of and time to convergence.
Appendices

A  Relative Sensor Perturbations, Unperturbed TDOAs . . . . . . . . . . 102
B  General Sensor Perturbations, Unperturbed TDOAs . . . . . . . . . . 112
C  TDOA Perturbations, Unperturbed Sensors . . . . . . . . . . . . . . . . 122
D  Sensor and TDOA Perturbation Examples . . . . . . . . . . . . . . . . 132
Appendix A

Relative Sensor Perturbations,
Unperturbed TDOAs
Figure A.1: Relative Sensor Perturbation LSSPE, $B=1e0$
Figure A.2: Relative Sensor Perturbation LSSPE, B=1e1
Figure A.3: Relative Sensor Perturbation LSSPE, B=1e2
Figure A.4: Relative Sensor Perturbation LSSPE, B=1e3
Figure A.5: Relative Sensor Perturbation LSSPE, B=1e4
Figure A.6: Relative Sensor Perturbation LSSPE, B=1e5
Figure A.7: Relative Sensor Perturbation LSSPE, B=1e6
Figure A.8: Relative Sensor Perturbation LSSPE, B=1e7
Figure A.9: Relative Sensor Perturbation LSSPE, B=1e8
Appendix B

General Sensor Perturbations,

Unperturbed TDOAs
Figure B.1: General Sensor Perturbation LSSPE, B=1e0
Figure B.2: General Sensor Perturbation LSSPE, B=1e1
Figure B.3: General Sensor Perturbation LSSPE, B=1e2
Figure B.4: General Sensor Perturbation LSSPE, B=1e3

Max Log of Least Squares Solution Position Error vs Log of Source Distance
Sensor Distances from Origin = 1e3 in Meters
Figure B.5: General Sensor Perturbation LSSPE, B=1e4
Figure B.6: General Sensor Perturbation LSSPE, B=1e5
Figure B.7: General Sensor Perturbation LSSPE, B=1e6
Figure B.8: General Sensor Perturbation LSSPE, $B=1e7$
Figure B.9: General Sensor Perturbation LSSPE, B=1e8
Appendix C

TDOA Perturbations,

Unperturbed Sensors
Figure C.1: TDOA Perturbation LSSPE, B=1e0
Figure C.2: TDOA Perturbation LSSPE, B=1e1
Figure C.3: TDOA Perturbation LSSPE, B=1e2
Figure C.4: TDOA Perturbation LSSPE, B=1e3
Figure C.5: TDOA Perturbation LSSPE, B=1e4
Figure C.6: TDOA Perturbation LSSPE, B=1e5
Figure C.7: TDOA Perturbation LSSPE, $B=1e6$
Figure C.8: TDOA Perturbation LSSPE, B=1e7
Figure C.9: TDOA Perturbation LSSPE, $B=1e8$
Appendix D

Sensor and TDOA Perturbation

Examples
Figure D.1: Example 1 Max LSSPE
Relative Sensor & TDOA Perturbation LSSPE, B=1e1
Figure D.2: Example 1 Max LSSPE Improvement
Relative Sensor & TDOA Perturbation LSSPE Improvement, B=1e1
Figure D.3: Example 1 Max Iterations to Convergence
Relative Sensor & TDOA Perturbation Max Iterations to Convergence, B=1e1
Figure D.4: Example 2 Max LSSPE
General Sensor & TDOA Perturbation LSSPE, B=1\times 10^5
Figure D.5: Example 2 Max LSSPE Improvement
General Sensor & TDOA Perturbation LSSPE Improvement, B=1e5
Figure D.6: Example 2 Max Iterations to Convergence
General Sensor & TDOA Perturbation Max Iterations to Convergence, B=1e5
References


