Comparative Analysis of Multilateration Methods for Signal Emitter Positioning

André Christoffer Andersen
UCSD/NTNU
andre@andersen.im

June 8, 2011

Abstract

In this project paper we explore methods for spatial location of a signal emitter in Euclidian $\mathbb{R}$ˆ3 space by means of time difference of arrival (TDOA) at multiple omnidirectional sensors. First we explore two greedy solvers, using Newton's Method and Iterative Cone Alignment. These work on four sensors. Then, for the five-sensor case we reduce the problem to a linear programming problem. Finally convergence speed and signal error sensitivity is experimentally measured on the Newton's methods approach and the linear programming approach.

1 Introduction

Multilateration is a technique that uses multiple omnidirectional sensors in order to isolate the unknown position of a signal emitter in two- or three-dimensional Euclidian space. Two examples of sensors and signals could be microphones listening for sharp noises, or radio receivers listening for radio signals. In any case, these sensors are located at unique known positions where they listen for what is called a signal event. Such events are timestamped based on a synchronized or centralized clock common to all sensors. The signal from an emitter is registered by all sensors only once as the signal wave expands spherically in all directions with constant propagation speed. The time difference between when two sensors register the signal event is called the time difference of arrival (TDOA). Based on the TDOA and the location of each registration, i.e., sensor positions, we can deduce the location of the signal emitter.

1.1 Model definitions

Let sensors $i = 1, \ldots, k$ at positions p_i be sensors that output a timestamp t_i for when they register a signal wave with propagation speed v from a signal emitter at unknown position x, signal start time t and distance $v\,(t_i - t) = \|x - p_i\|$

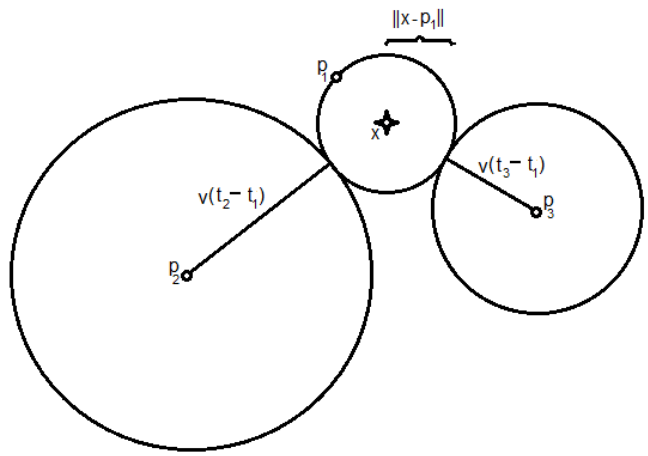

Figure 1: Distance relationships

[1] away. If we define sensor c to be the first sensor to register a signal event, then the signal wave has traveled for $(t_c - t)$ time units and a distance of $v(t_c - t) = \|x - p_c\|$ before reaching the first sensor. Furthermore, the TDOA between each sensor's timestamp and the initial sensor's timestamp is $(t_i - t_c)$ yielding the the distance $v\Delta t_i = v(t_i - t_c)$ traveled by the signal wave since initially being registered and finally reaching the ith sensor. From the definitions above we have the following important relationships.

$$\|x - p_i\| = v\,(t_i - t) \tag{1}$$

and the equivalent formulations

$$\|x - p_i\| = v(t_i - t_c) + \|x - p_c\| \tag{2}$$

$$\|x - p_i\| = v(t_i - t_c) + v(t_c - t) \tag{3}$$

Equation 2 is illustrated in figure 1.1 for a three-sensor problem where $i = 1, 2$ and $c = 1$.

1.2 Geometric Interpretation

In general the positions p_i, p_c and TDOA $t_i - t_c$ of a pair of sensors limits the signal emitter's position to lay on one sheet of a circular two-sheeted hyperboloid with foci at p_i and p_c. Given that c is the first of the two sensors to register a signal event, then only the sheet at focus p_c is materialized. The associated directrix plane can be found by a known position vector and normal vector.

[1] If not stated otherwise, all norms are Euclidian norms.

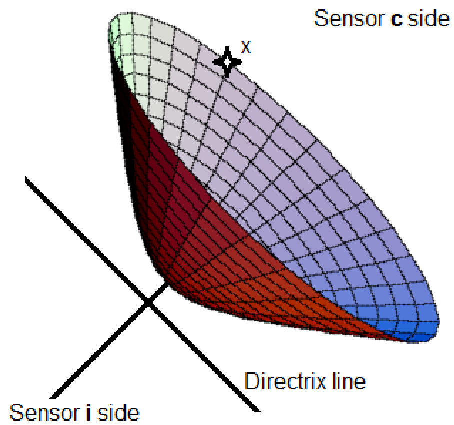

Figure 2: Given sensor pair i and c the feasible region of emittor position x is constrained to one sheet of a two-sheeted circular hyperboloid.

Since the directrix is orthogonal to the vector pointing from sensor i to sensor c, we know that $p_c - p_i$ itself or its unit vector

$$\frac{p_c - p_i}{\|p_c - p_i\|}$$

can serve as a normal vector. Next we need a position vector with endpoint on the directrix plane. One such vector can be defined by the point at which the directrix intersects the line segment between p_i and p_c. This point lays a length $v(t_i - t_c)$ away from p_i towards p_c, that is, it lays at the position vector

$$p_i + v(t_i - t_c)\frac{p_c - p_i}{\|p_c - p_i\|}$$

Figure 1.2 illustrates one such surface in $\mathbb{R}^3$ while figure 1.2 illustrates multiple hyperboles $\mathbb{R}^2$ all intersecting at the emiter position x.

1.3 Optimization Problem

Simply put, our goal is to find a solution to equation 1, 2 or 3. Equation 1 can be captured as a scalar function $f_i : \mathbb{R}^4 \to \mathbb{R}$ such that

$$f_i(x, t) = \|x - p_i\| - v(t_i - t) = 0 \tag{4}$$

One natural way to formulate the optimization task of satisfying $f_i(x, t) = 0$ would be to try to minimize the sum of $f_i(x, t)$over all i constrained individually not go below zero.

$$\min \sum_{i=1}^{k} \|x - p_i\| - v(t_i - t)$$
$$\text{s.t.} \|x - p_i\| - v(t_i - t) > 0 \quad \forall i$$

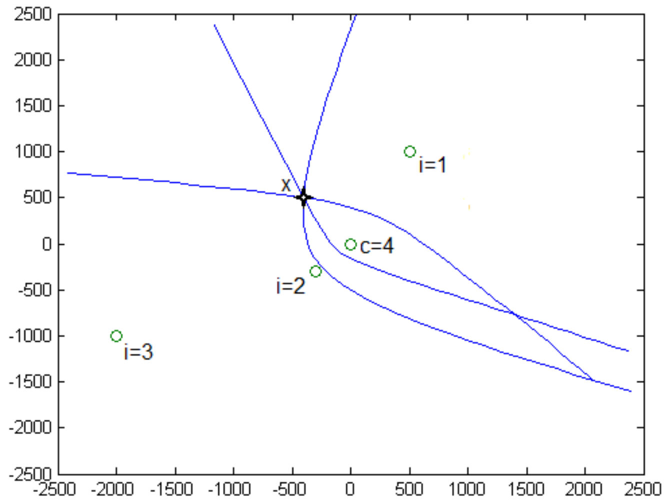

Figure 3: Given pairings between tre sensors $i = 1, 2, 3$ with sensor $c = 4$ we can generate three hyperboles which all intersect at emittor possition x.

which we can try to formulate as a second-order cone programming problem

$$\begin{aligned} &\min \gamma \\ &\text{s.t.} 0 < \|x - p_i\| - v\,(t_i - t) < \gamma \quad \forall i \end{aligned} \tag{5}$$

Alas, the lower bound constraint makes the problem non-convex. However, when that is said it is possible to make the problem linear with enough sensors registering signals. This will be explored in section 2.3.

2 Problem Solving Methods

2.1 Newton-Raphson Method

Using Newton-Raphson Method for solving multilateration problems [B&P] is a greedy, but simple, approach to iteratively isolate a feasible point on the space-time cone (x, t) described by $f_i(x, t)$ in equation 4. Using $y = (x, t) = \begin{bmatrix} x \\ t \end{bmatrix} \in \mathbb{R}^4$ we can extend $f_i(y)$ by describing the entire system as a vector function $f : \mathbb{R}^4 \to \mathbb{R}^4$ such that

$$f(y) = \begin{bmatrix} f_i(y) & f_2(y) & f_3(y) & f_4(y) \end{bmatrix}^T$$

Now, writing the initial guesses as $y_0 = (x_0, t_0) \in \mathbb{R}^4$ we can calculating the individual hessians and formulating the jacobian.

$$\nabla f_i(y) = \begin{bmatrix} \|x - p_i\|^{-1}(x - p_i) \\ v \end{bmatrix} \in \mathbb{R}^4$$

$$J(y) = \begin{bmatrix} \nabla f_i(y) & \nabla f_2(y) & \nabla f_3(y) & \nabla f_4(y) \end{bmatrix}^T \in \mathbb{R}^{4\times 4}$$

This gives us a way to approximate the vector function by

$$f(y) \approx f(y_0) + J(y)\Delta y = 0$$

which yields the following update step

$$\Delta y = -J(y)^{-1} f(y_0)$$

and update rule

$$y \leftarrow \hat{y} := \hat{y} + \Delta y = \hat{y} - J(y)^{-1} f(y_0) \tag{6}$$

2.2 Iterative Cone Alignment

As with the Newton-Raphson method, Iterative Cone Alignment [S&W] uses a greedy algorithm to converge to a solution. However, in contrast with the Newton-Raphson method, we don't converge based on the problem derivatives. Instead we pose the problem as an iterative process of error and correction.

Given a proposed point $(\hat{x}, \hat{t})$ in conic space-time while keeping $\hat{t}$ constant we can calculate the feasibility error of $\hat{x}$ with respect to some sensor i. This error can be found by direct application of equation 1, thus prompting us to reuse function f_i of equation 4 from the Newton-Raphson approach.

$$f_i(\hat{x}, \hat{t}) = \|\hat{x} - p_i\| - v(t_i - \hat{t}) = 0$$

Of course, optimizing this error function by only changing position isn't enough; we also need to change time in a reasonable way. The approach Schindelhauer and Wendeberg suggests is to use the normal vector of the cone, described by the error function, at point $(\hat{x}, \hat{t})$ to find the shortest direction to a feasible point $(\hat{x}, \hat{t})$ with regard to sensor i. When $\hat{t} > t_j + \|\hat{x} - p_i\|/v$ this normal vector is defined by

$$n = \left(\frac{\hat{x} - p_i}{\|\hat{x} - p_i\|}, \frac{1}{v}\right)$$

else we force a reduction of only $\hat{t}$ by using

$$n = \left(\vec{0}, -1\right)$$

Now that we know in which direction we want to go, we need to find out how far to go. To simplify notation we can use $\hat{y} = (\hat{x}, \hat{t})$ and $q_i = (p_i, t_i)$. Finding a the distance along the normal to the lateral surface of the cone can be accomplished by noting that there exists a scalar d such that

$$f(\hat{y} - q_i + dn) = 0$$

which when solved yields[S&W]

$$d(\hat{y} - q_i) = \frac{f(\hat{y} - q_i + n)}{f(\hat{y} - q_i) + f(\hat{y} - q_i + n)}$$

2.3 Reduction to Linear Form

The third method [B&M] for determining the location of a signal emitter uses three unique pairs i, j of sensors in addtion to the first discovery sensor c to make the problem a linear system of three equations. This requires at minimum five sensor. To derive the system we start by expanding equation 2

$$\begin{aligned}
\|x - p_i\| &= v(t_i - t_c) + \|x - p_c\| \\
\|x - p_i\|^2 &= (v(t_i - t_c) + \|x - p_c\|)^2 \\
\|x - p_i\|^2 &= v^2(t_i - t_c)^2 + 2v(t_i - t_c)\|x - p_c\| + \|x - p_c\|^2
\end{aligned}$$

Introducing another sensor j we can eliminate the $\|x - p_c\|$ term.

$$\begin{aligned}
-2\|x - p_c\| &= v(t_i - t_c) + \frac{\|x - p_c\|^2 - \|x - p_i\|^2}{v(t_i - t_c)} \\
-2\|x - p_c\| &= v(t_j - t_c) + \frac{\|x - p_c\|^2 - \|x - p_j\|^2}{v(t_j - t_c)} \\
v(t_j - t_c) + \frac{\|x - p_c\|^2 - \|x - p_j\|^2}{v(t_j - t_c)} &= v(t_i - t_c) + \frac{\|x - p_c\|^2 - \|x - p_i\|^2}{v(t_i - t_c)}
\end{aligned} \tag{7}$$

Now, expanding the definitions of the squared distances from each of the sensors and the emittor we get the following.

$$\begin{aligned}
\|x - p_i\|^2 &= (x - p_i)^T (x - p_i) = x^T x - 2p_i^T x + p_i^T p_i \\
\|x - p_j\|^2 &= x^T x - 2p_j^T x + p_j^T p_j \\
\|x - p_c\|^2 &= x^T x - 2p_c^T x + p_c^T p_c
\end{aligned} \tag{8}$$

With results from 8 we can rewrite 7 as

$$v(t_j - t_c) + \frac{2\left(p_j^T - p_c^T\right)x + p_c^T p_c - p_j^T p_j}{v(t_j - t_c)} = v(t_i - t_c) + \frac{2\left(p_i^T - p_c^T\right)x + p_c^T p_c - p_i^T p_i}{v(t_i - t_c)} \tag{9}$$

Notice that the quadratic terms have fallen out and we are left with a very solvable linear system of equations

$$a_{ijc}^T x = b_{ijc}$$

where

$$\begin{aligned} a_{ijc} =& 2\left(v(t_j - t_c)(p_i - p_c) - v(t_i - t_c)(p_j - p_c)\right) \in \mathbb{R}^3 \\ b_{ijc} =& \quad v(t_i - t_c)\left(v^2(t_j - t_c)^2 - p_j^T p_j\right) \\ +& \quad (v(t_i - t_c) - v(t_j - t_c))\, p_c^T p_c \\ +& \quad v(t_j - t_c)\left(p_i^T p_i - v^2(t_i - t_c)^2\right) \in \mathbb{R} \end{aligned}$$

This can be expressed as just

$$Ax = b$$

where each pair i, j creates the matrix A and vector b defined by $b_{ijc} \in b \in \mathbb{R}^3$ and $a_{ijc} \in A \in \mathbb{R}^{3\times 3}$ such that a_{ijc}^T are the rows of A. This can be solved using LP or some other more or less direct method. We will be simply computing the iverse of A such that

$$\hat{x} = A^{-1}b = A\backslash b$$

3 Experiment Design

Implementation Implementation of the selected methods of section 2 is done in MATLAB. All experiments are simulted within the MATLAB framwork. As discussed in section 2.3, the linear problem of that section is only a linear system of equations; this means we can solve it by directly taking the inverse or to use a LP capabilities of the optimization suit CVX [G&B]. The latter alternative is predicted to be much slower. Accuracy and reliability on the other hand is unknown.

Experiment setup and problem state generation The experiments are designed to illustrate the performance of the solution methods based on quantiative measurements. For all experiments we apply the same randomly generated set of problem states on each solution method. The problem state generator places sensors based on a uniform random distribution within a given square area about the origin. The signal emitters are placed the same way where the only difference is its larger area. This will make sure that the tests are exposing the solution methods to problems where the signal emitter isn't always in the spatial convex hull of the sensors. This is done since it might be a source of difference between the solution methods.

3.1 Solution realibility

The two chosen methods get scored based on how many successful convergence results it can generate on a fixed amount of problem states. A point is achieved if the method converges within a specified small distance from the emitter position with ample convergence time.

3.2 Convergence accuracy and speed

Speed tests are conducted to find the qualitatively best converging solver. To do this we give both methods equal time to solve the problem and measure the error when done. The one with lowererror is said to be the fastest.

3.3 Experiment Setup

In order to aid intuition we will generate problem configurations which have real world interpretation. The scenario we depict is passive detection and localization of a submarine (signal emitter) in a four cubic kilometer area. Four or five acoustic sensors are randomly placed in a one cubic kilometer area situated within the center of the four cubic kilometer area. A submarine is successfully detected when a solution method has converged to within one meter of the center of the submarine. Underwater sound propagation is to$v = 1500$ m/s. And to ensure good estimates we will be generating 1000 problem scenarios for each experiment and using the arithmetic mean on these results. The mean position of the sensors is used as a heuristic for selecting an initial guess for the Newton's method approach.

4 Results and Analisis

Since Newton's method is a greedy algorithm it is prone to converge to local minima. This is evident in the test results. The Newton's method solver produced 632 local minimums out of 1000 problem setups, i.e., the solver failed to produce the actual position of the signal emitter 63% of the time.

Linear methods are much more reliable and never failed to converge in our experiments. This is due to it being a convex problem rather than a hilly problem like the formulation used by our Newton's method solver. Given the same amount of time to converge the linear problem is solved to an error of less than 10^{-12}, practically zero, and Newton's goes down to only 0.0056.

There is no doubt; the linear problem formulation is far more supperior.

5 Conclusion

In this project paper we have explored different aprouches to the TDOA problem. We explored different problem definitions and experimentally compaired two prisipally different solution methods.

Future studies could be to do Lagrangian or SDP relaxation of the optimization problem of equation 5 for the four-sensor problem. However, from a preliminary and practical point of view, it is hard to imagine a situation where you are stuck with having to settle with exactly four sensors and not able to introduce a fifth sensor. It means the difference between working with a non-convex problem using greedy minimizes which don't guarantee correct solutions versus having a LP problem which is readily solvable with a multitude of methods.

A limitation and potential future work is looking at what effect errors and signal omissions have on the localization task. Without further studies it is unclear what effect it would have on the presented solution methods. Also there was a period in testing where Newton's method converged only 3% of the time to the wrong minimum. It is unclear if this was a fluke or a favorable configuration.

References

[B&P] Bohnish Banerji and Sidharth Pande; Sound Source Triangulation Game, Retrived from web, June 2011.

[S&W] Christian Schindelhauer and Johannes Wendeberg, Localization Solely based on Ambient Signals, Paper of 2010.

[B&M] R. Bucher and D. Misra, A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System, Paper of 2001.

[G&B] Michael Grant and Stephen Boyd, CVX: Matlab Software for Disciplined Convex Programming, Software package retrived from web June 2011.