Systems Design Project: Indoor Location of Wireless Devices

Prepared By:

Brian Murphy
Senior Systems Science and Engineering
Washington University in St. Louis
Phone: (805) 698-5295
Email: bcm1@cec.wustl.edu

Supervised By:

Patricio La Rosa and Paul Min
Department of Electrical and Systems Engineering
Washington University in St. Louis
Abstract:

Location Based Services (LBS) deliver the user positioning data for the purpose of navigation in unknown environments. Within the coming years, the industry is expected to experience significant growth in revenue and market presence. The more prominent technology within the LBS industry is the Global Positioning System (GPS). GPS is highly effective for navigating outdoor areas but lacks the precision necessary for effectively mapping and negotiating indoor environments.

Indoor mapping technology generally utilizes wireless indoor devices that emit reference signals to the referent devices (sometimes stationary and sometimes mobile) which, in turn, map the surrounding area. Over the past few years, there has been growing demand for precise and relatively cheap indoor mapping technology. This is largely due to the variety of its commercial applications such as helping handicapped persons navigate within their household, giving household robotic devices the ability to map indoor environments, tracking products in a warehouse, and managing retail inventory. In designing an indoor mapping system, the main goals are precision and accuracy. An ancillary goal would be to design a simplistic system which would prove less costly.

This project takes aim at both the aforementioned primary and ancillary goals. We seek to develop a precise method for indoor positioning and navigation by first selecting an appropriate communication protocol and then selecting an efficient tracking algorithm. After surveying available methods, we determined that the Wireless Local Area Network (WLAN) is the appropriate protocol due to its simplicity and low costs. Before choosing a tracking
algorithm for our moving source, we needed to be able to accurately estimate the position of a static source both with and without measurement error. Without measurement error, we could conveniently calculate a Linear and Nonlinear solution for our source position using the three WLAN APs and the trilateration method. With measurement error, we could still use trilateration and the three APs but did not have a perfect three-way intersection that could be ascertained mathematically. Thus, we had to devise our own method of estimating a source position which simply involved taking the average of three closest intersection points. Once we established methods for estimating the position of a static source, we decided to use a Kalman Filter as our tracking algorithm for a mobile source due to its precision and relatively simple implementation.

1.0 Description of Projects and Accomplishments:

1.1 Communication Protocol

The first phase of the project involved selecting an efficient communication protocol between our signal emitters (APs) and signal receiver (source terminal). At the onset of this project the two candidates were the WLAN and Bluetooth protocols. After surveying both methods, we ultimately determined that WLAN protocol was more viable because it is easier to extract pertinent data such as signal strength and round trip time (RTT).

RTT in particular would be of great importance to this project. Under the IEEE 802.11 WLAN Standard, RTT is measured as the time elapsed between the Request-to-Send (RTS) frame and the Clear-to-Send (CTS) frame. RTS and CTS represent a frame sent by the transmitter (source) and a response frame sent by the receiver (AP). Therefore, RTT can be
estimated as the time between when the source asks the AP for a signal and when the AP responds. Using the simple formula “distance = rate x time” where ‘time’ is represented by RTT and ‘rate’ is represented by the speed of light, we can calculate ‘distance’ which represents the signal radius of an AP. When we know the signal radius of each point, we can mathematically calculate the intersection of all three signals and this intersection provides us with a reasonable estimation of our source terminal.

1.2 Trilateration without Measurement Error or Noise

The experiment involves three WLAN access points (APs) and a mobile terminal that receives the signals from each of three APs. Each AP emits a unique signal which forms a unique circle with a unique radius. Using the method of trilateration, we find the intersection of these three circles mathematically and the result is our position estimation. We can use a fourth AP to measure the height of the mobile terminal but, for this project we assume a flat surface and thus only three APs are needed to measure the terminal’s location in two dimensions. This assumption is made for the purpose of simplifying calculations later on. The trilateration process is visualized in the figure below:
For this phase of our calculations, we assume there is no measurement error or noise and thus all three of our AP signals can conveniently intersect at one point. This assumption allows us to mathematically derive an estimate for the intersection point first through the Linear Least Squares method and then a refined estimate through the Nonlinear Least Squares method. The first step in solving for this intersection point is identifying the pertinent system variables. For the system visualized in Figure 1.2.1, we define the following parameters:

\[(x, y): \text{ source position}\]

\[(x_i, y_i), r_i: \text{ center and radius of APs' signals for } i=1,2,3\]

*Figure 1.2.2: Trilateration System Parameters*

### 1.2.1 Linear Least Squares Method

With system defined system parameters from *Figure 1.2.2*, our system can be visualized as such:

*Figure 1.2.1.1: Trilateration Visualized with System Parameters*
Knowing that the distance between each individual AP and the source is the AP’s signal radius, we can derive the following system of equations using the distance formula:

\[
\begin{align*}
(1) \quad (x_1 - x)^2 + (y_1 - y)^2 &= r_1^2 \\
(2) \quad (x_2 - x)^2 + (y_2 - y)^2 &= r_2^2 \\
(3) \quad (x_3 - x)^2 + (y_3 - y)^2 &= r_3^2
\end{align*}
\]

3 equations, 2 unknowns and \((x_i, y_i), r_i\) for \(i=1,2,3\) are given

In order to obtain a solution to the aforementioned system of equations, we apply the Linear Least Squares (LLS) method. Though the method is not the most accurate, it provides a decent means for estimating the source, an estimate we can later improve via the Nonlinear Least Squares method.

To linearize the system, we need to remove one constraint and so we arbitrarily choose AP\(_1\) which gives us a system of two equations and two unknowns. The source estimation is set up as follows:

1. Calculate the distance from AP\(_1\) to the other APs via the following formula:
   \[
d_{ij} = \sqrt{(x_i - x)^2 + (y_i - y)^2}, \quad (i=2,3 \text{ and } j=1)
   \]

2. Calculate the system constraints \((b_{ij})\) given by:
   \[
b_{ij} = \frac{1}{2}(r_j^2 - r_i^2 + d_{ij}^2), \quad (i=2,3 \text{ and } j=1)
   \]
Using Matlab, we implemented the LLS method above and effectively estimated the initial static source position with the radii and location of the three AP signals as our inputs.

### 1.2.2 Nonlinear Least Square Method

As mentioned before, the LLS method is not the most accurate method of estimation but, using the result from the LLS method as an initial estimate, we can improve that estimate by minimizing the sum of the squared error for each distance using the Nonlinear Least Squares (NLS) method. NLS is an iterative algorithm that runs until the difference between the current and previous iterations meets some modifiable and pre-specified threshold. The NLS is set up as follows:

3. Simplify the system into matrix form:

\[ Ax = b \]

where

\[
A = \begin{pmatrix}
x_2-x_1 & y_2-y_1 \\
x_3-x_1 & y_3-y_1
\end{pmatrix},
\]

\[
x = \begin{pmatrix}
x-x_1 \\
y-y_1
\end{pmatrix},
\]

\[
b = \begin{pmatrix}
b_{21} \\
b_{31}
\end{pmatrix}
\]

4. Since the radii, \( r \), are only approximate, the problem requires the determination of \( x \) such that \( Ax = b \):

\[ A^T Ax = A^T b \]

5. Assuming the APs are not placed in a straight line, we know that \( A^T A \) is non-singular and can solve for \( x \):

\[ x = (A^T A)^{-1} A^T b \]
The above algorithm can easily be programmed in Matlab given the following equations:

\[
(1) \quad R_{k+1} = R_k - (J_k^T J_k)^{-1} J_k^T f_k
\]

*\(R_k\) denotes the \(k^{th}\) approximate solution \((x, y)^T\), \(J_k\) represents the Jacobian matrix, and \(f_k\) represents error between the given radius and the measured distance at the \(k^{th}\) iteration for each access point.

Figure 1.2.2.1: NLS Algorithm

The above algorithm can easily be programmed in Matlab given the following equations:

\[
J^T J = \begin{pmatrix}
\sum_{i=1}^n (x-x_i)^2 & \sum_{i=1}^n (x-x_i)(y-y_i) \\
\sum_{i=1}^n (x-x_i)(y-y_i) & \sum_{i=1}^n (y-y_i)^2
\end{pmatrix}
\]

\[
J^T f = \begin{pmatrix}
\sum_{i=1}^n (x-x_i)^2 f_i \\
\sum_{i=1}^n (y-y_i)^2 f_i
\end{pmatrix}
\]

*measurement error \(f_i = ((x-x_i)^2 + (y-y_i)^2)^{1/2} - r_i\) for \(i=1,2,3\)

Figure 1.2.2.2: NLS Algorithm Simplified

//Still to do: explain and define our NLS method, show graphical results from Matlab functions

//Explain our ‘cluster’ method for source estimation with measurement noise