# Trilateration:

## The Mathematics Behind a Local Positioning System


**Willy Hereman**

Department of Mathematical and Computer Sciences
Colorado School of Mines
Golden, Colorado, U.S.A.
whereman@mines.edu
http://inside.mines.edu/~whereman/

Department of Computer Engineering
Turgut Özal University, Keçiören, Ankara, Turkey



Tuesday, June 21, 2011, 11:00

## Acknowledgements

William Murphy (M.S. Dissertation, CSM, 1992)

William Navidi (Department of MCS, CSM)

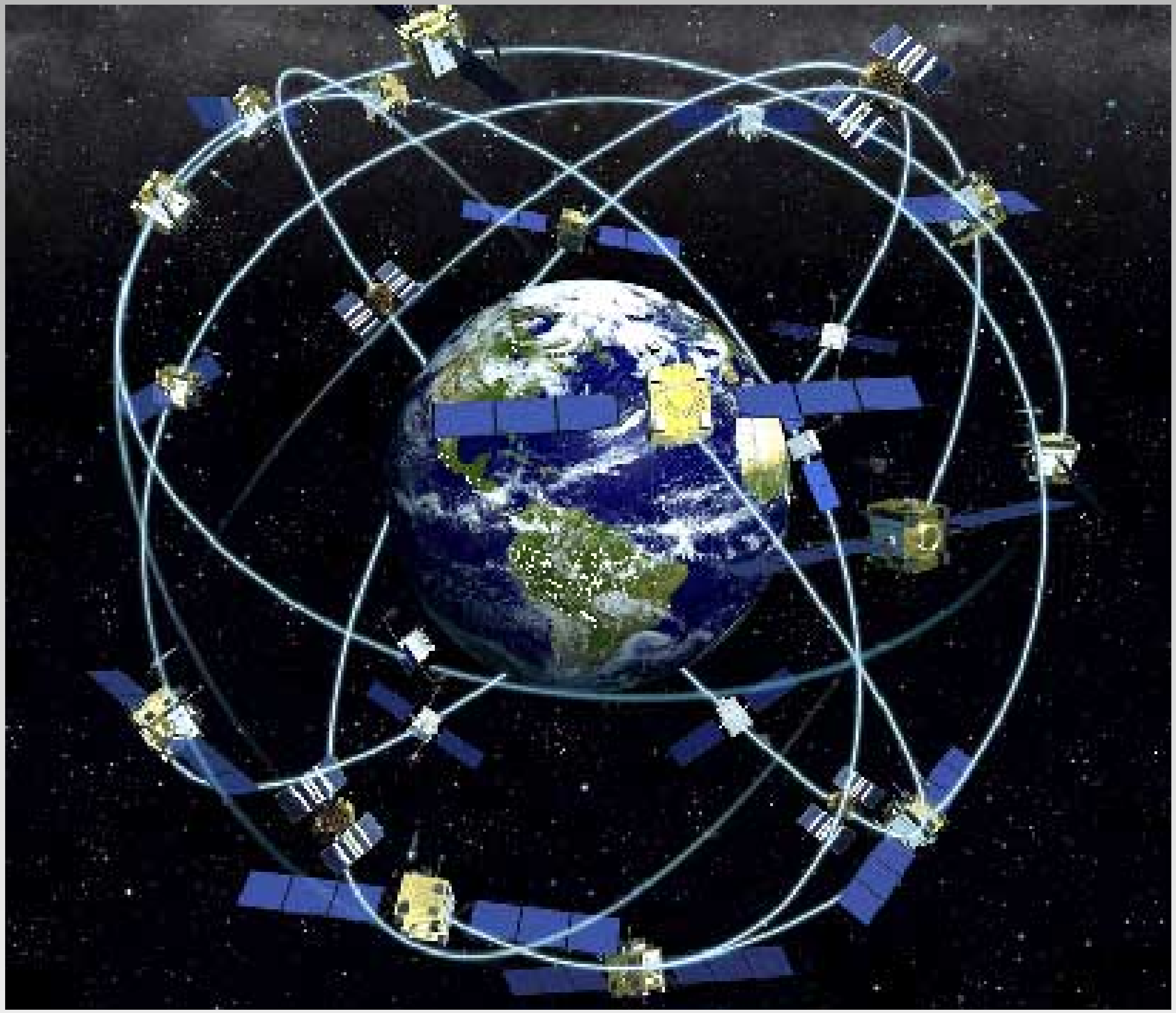
Atlantic Richfield Company (ARCO)

Thunder Basin Coal Company, Wright, Wyoming

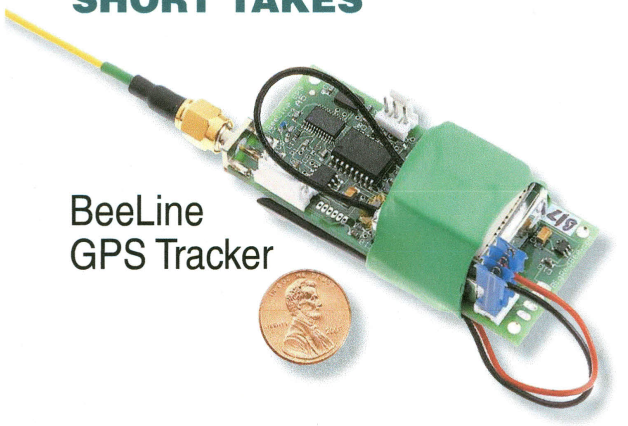
This presentation was made in `TeXpower`

# Outline

- Problem statement: What is trilateration?
- How did we get into this?
- Applications of our algorithm
- An exact linearization
- Linear least squares method
- Nonlinear least squares method
- *Mathematica* demonstration
- Simulation – Results of experiments
- Conclusions



Global Positioning System (GPS)



# BeeLine GPS Tracker

You have to admire any company that is comfortable with calling itself "BigRedBee." Not BeeTrex. Not BeeCom. Just *BigRedBee*.

This is a small business with a small, clever product: the BeeLine GPS Tracker. The BeeLine is a tiny (1¼ × 3 inch) module that contains a GPS receiver and GPS patch antenna, a Lithium-Poly battery and a 70-cm FM transmitter. The whole package weighs about 2 ounces.

The BeeLine is designed to be a go-anywhere APRS tracker. In case you're unfamiliar with the term, APRS stands for the Automatic Position Reporting System. An APRS tracker takes position information supplied by a Global Positioning System (GPS) receiver, reformats it as packet radio data, converts the data to a modulated audio signal and passes the signal to a transmitter (typically a VHF FM radio). At the receiving end, a packet radio Terminal Node Controller (TNC) decodes the transmission and feeds the information to a computer running APRS software. The result is a computer-generated map that displays the location of the tracker (and the object being tracked).

Unlike some bulky APRS tracking setups comprised of separate GPS receivers, TNCs and radios, the BeeLine integrates everything, including the battery, into a single compact unit. The only downside is that the BeeLine operates on 70 cm, whereas most APRS activity takes place on 2 meters (144.39 MHz). For specialized applications where you're not concerned with making the position information available to the traditional APRS network, this probably isn't an issue.

## The BeeLine Package

For this review I purchased the complete BeeLine GPS package, which includes a battery charger, serial adapter (to communicate with your computer) and a 70-cm antenna.

The battery charger is an imported device originally intended to charge cell phone batteries. The BigRedBee Web site suggests modifying the charger to make it easier to interconnect with the BeeLine module. That's the approach I took, modifying the charger by adding a cable with a small three-terminal connector (DigiKey part number WM4201-ND) so that I could easily plug in the BeeLine for recharging.

The serial adapter is something you'll use only occasionally to program the BeeLine with your call sign and other parameters. The *BeeLine Communicator* software for *Windows* is downloadable from the BigRedBee Web site. You simply plug the BeeLine into the serial adapter, plug the serial adapter into a convenient COM port on your computer and then read and write your settings to the BeeLine. It is interesting to note that you can also set the transmit frequency and output power in this fashion. The BeeLine will transmit anywhere in the 70-cm band. I set my unit on 433.920 MHz with full output power (about 16 mW).

The antenna is a quarter wavelength flexible wire terminated in an SMA connector.

## Kitty Tracker!

My first impulse was to launch the BeeLine in a model rocket, but the odds of it finding a new home in a treetop placed that notion well outside my comfort zone. So, I grabbed the nearest moving object at hand—my cat. I attached the BeeLine to her harness and turned her loose for a neighborhood patrol. I set the BeeLine to transmit a position beacon once every 60 seconds.

The BeeLine's GPS receiver quickly acquired enough satellite signals to determine her position and apparently maintained GPS lock throughout most of her journey. Back at home, I had no difficulty receiving the BeeLine's reports. You can see the result in Figure 1.

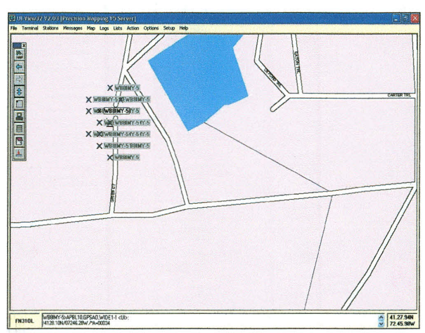


**Figure 1—Herding cats may be impossible, but tracking one with a BeeLine GPS certainly works! These position reports were received on 433.920 MHz using a Kenwood TS-2000 transceiver (with its built-in packet radio TNC) and displayed with *UI-View* APRS software.**

## Serious Applications

The minimal size and weight of the BeeLine makes it ideal for a variety of tracking applications such as model rockets, high-altitude Amateur Radio balloons, radio-controlled airplanes, search and rescue, etc. The BeeLine also features onboard memory that will record about 10 minutes worth of position data. This is particularly useful for model rocket and R/C airplane activity.

*Manufacturer: BigRedBee, 5752 Bay Point Dr, Lake Oswego, OR 97035;* **www.bigredbee.com**. *$299*

Steve Ford, WB8IMY ♦ *QST* Editor ♦ sford@arrl.org

# How did we get into this?

# Accidents Happen

# Bulldozer with Beacons

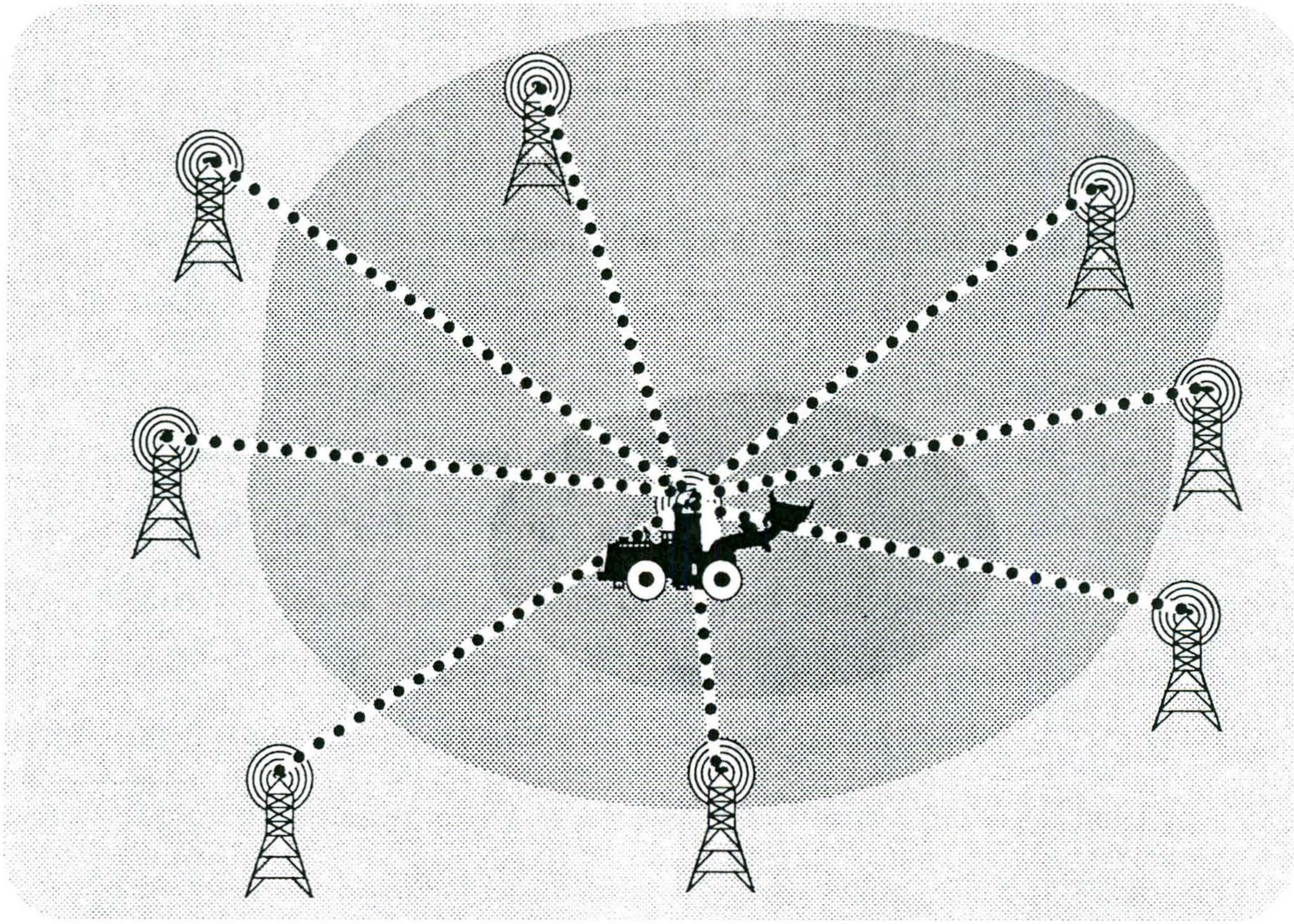


**Figure 1.1** Illustration of the Positioning System

## Applications of our Algorithm

- “Thunder Basin Coal Mine” – locating bulldozers
- Surveying without triangulation (Mining)
- Mobile computing – sensor networks
- Geosensing networks (SmartGeo)
- Precision manufacturing
- Positioning systems for medical applications (Electrical Engineering)
- “Ignite” program – blasting rockets

# Program introduces youngsters to careers in science

By Neesha Hosein

Rocket science is not such an out-of-this-world concept for some high school students, thanks to the SystemsGo Aeroscience Program.

SystemsGo is a product of Ignite, a nonprofit organization whose mission is to enhance education for better workforce development and to ignite tomorrow's innovators.

Brett Williams, an instructor at Fredericksburg High School in Fredericksburg, Texas, is the founder and director of SystemsGo, which is centered around teaching students to design and launch rockets.

"It is a progressive, innovative public education program, developed to promote project-based learning (and) problem-solving. Its intent is to support workforce development in the field of engineering," Williams said.

The program is supported by NASA and is a two-year, junior-senior program. The first year is dedicated to the design and development of remotely operated and unmanned vehicles, which are used for research or industrial applications. The second year is when students design and fabricate rockets for testing, reaching elevations of 80,000 to 100,000 feet.

Aside from NASA, SystemsGo is certified by The Space Foundation, Johnson Space Center, the U.S. Army, as well as numerous government and corporate partners.

Williams made a visit last December to Houston to meet with NASA Shuttle Program's Flight Dynamics Division and also with some local school officials interested in replicating the program.

With Governor Perry's approval of funding, William's said the program could extend beyond Texas to Tennessee, Indiana, Iowa, Maryland, Virginia and New York.

Michelle Woods, program coordinator, said that SystemsGo promotes teamwork.

"It is remarkable to see the kids divide into teams and delegate responsibilities, while at the same time, working together," Woods said.

Ignite President Carson Dickie said that "after two years of outreach, Ignite has established SystemsGo at 26 high schools across Texas."

"We presently have five schools in Houston," Williams said. Williams hopes the program, in partnership with NASA, will continue to grow to other areas and draw more schools to participate.

PHOTO/US ARMY

*The 500-pound Red Bird-11 rocket launched on May 31, 2008, from the White Sands, N.M., army base. It was the research project of six graduate students out of Stanford University.*


FIGHTIN TEXAS
AGGIES

# Problem Statement and Setup

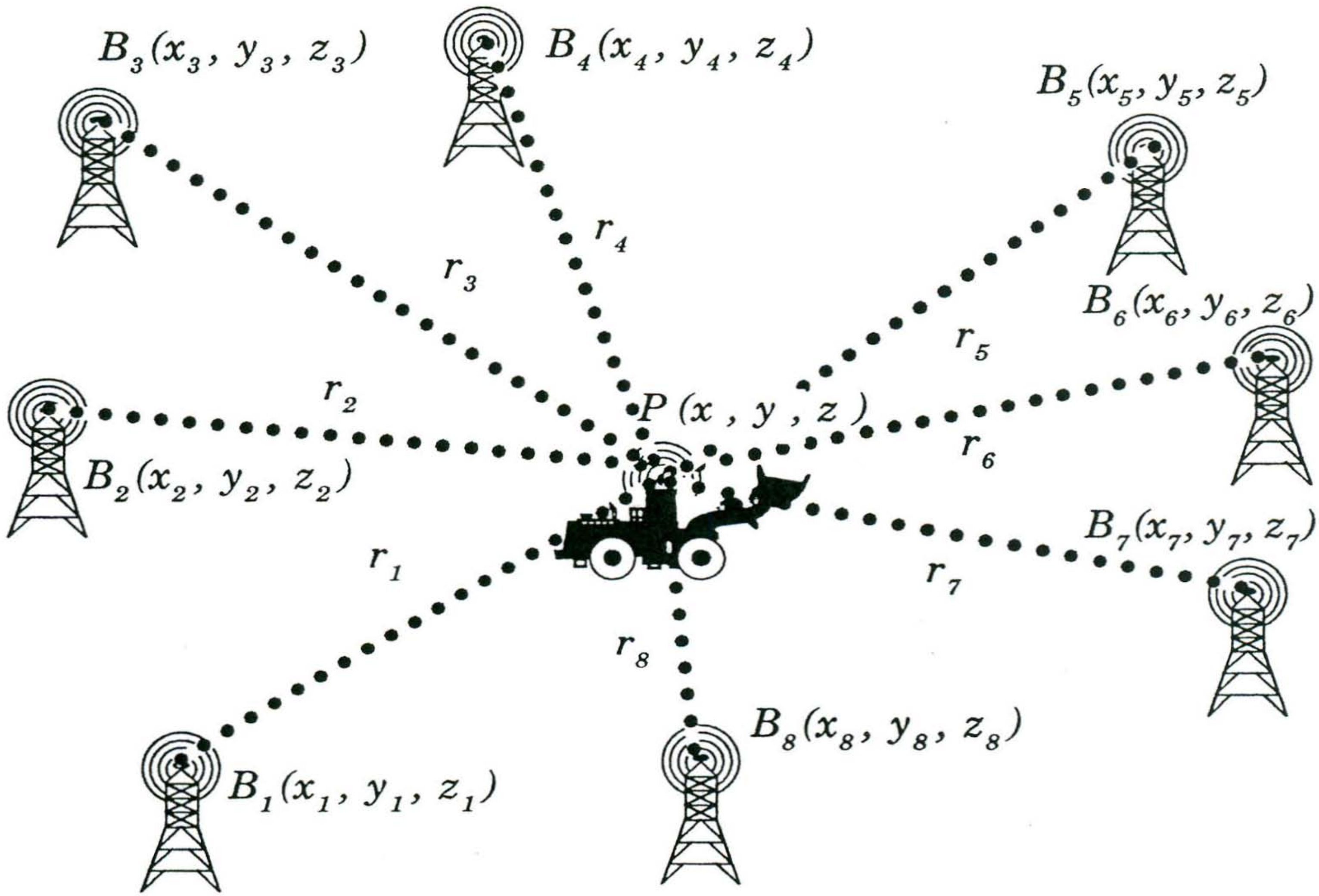



**Figure 1.2** Illustration of Mathematical Symbols

# Notations

$\theta = (x, y, z)$ : spatial coordinates of target point $\theta$.

$B_i = (x_i, y_i, z_i)$ : exact location of beacon $B_i$.
$i = 1, 2, \ldots, n$ with $n \geq 4$.

$d_i(\theta) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}$ : true distance between beacon $B_i$ and target $\theta$.

$(x_r, y_r, z_r)$ : exact coordinates of a reference point.

$d_{ir} = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$ : true distance between reference point and beacon $B_i$.

$d_r(\theta) = \sqrt{(x - x_r)^2 + (y - y_r)^2 + (z - z_r)^2}$ : true distance between the reference point and the target $\theta$.

# Derivation of an Exact Linear Model

Apply a simple trick (the cosine rule!)

$$
\begin{aligned}
d_i(\theta)^2 &= (x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 \\
&= (x - x_r + x_r - x_i)^2 + (y - y_r + y_r - y_i)^2 \\
&\quad + (z - z_r + z_r - z_i)^2 \\
&= (x - x_r)^2 + 2\,(x_r - x_i)(x - x_r) + (x_r - x_i)^2 \\
&\quad + (y - y_r)^2 + 2\,(y_r - y_i)(y - y_r) + (y_r - y_i)^2 \\
&\quad + (z - z_r)^2 + 2\,(z_r - z_i)(z - z_r) + (z_r - z_i)^2
\end{aligned}
$$

Keep the double product terms on the left hand side.

$$
\begin{aligned}
& 2\Big((x_i - x_r)(x - x_r) + (y_i - y_r)(y - y_r) + (z_i - z_r)(z - z_r)\Big) \\
= \;& (x - x_r)^2 + (y - y_r)^2 + (z - z_r)^2 \\
& +(x_r - x_i)^2 + (y_r - y_i)^2 + (z_r - z_i)^2 - d_i(\theta)^2 \\
= \;& d_r(\theta)^2 + {d_{ir}}^2 - d_i(\theta)^2
\end{aligned}
$$

where $i = 1, 2, \ldots, n$ with $n \geq 4$.

Use any beacon (say, $B_1$) as reference point.

Replace exact distances by measured distances.

$$(x_2 - x_1)(x - x_1) + (y_2 - y_1)(y - y_1) + (z_2 - z_1)(z - z_1)$$
$$\approx \frac{1}{2}\left[{r_1}^2 - {r_2}^2 + d_{21}^2\right] := b_{21}$$
$$(x_3 - x_1)(x - x_1) + (y_3 - y_1)(y - y_1) + (z_3 - z_1)(z - z_1)$$
$$\approx \frac{1}{2}\left[{r_1}^2 - {r_3}^2 + d_{31}^2\right] := b_{31}$$
$$\vdots$$
$$(x_n - x_1)(x - x_1) + (y_n - y_1)(y - y_1) + (z_n - z_1)(z - z_1)$$
$$\approx \frac{1}{2}\left[{r_1}^2 - {r_n}^2 + d_{n1}^2\right] := b_{n1}$$

Linear system of $(n-1)$ equations in $3$ unknowns.

# Linear Least Squares (LSQ) Model

Write the linear system in matrix form: $\boxed{\mathbf{A}\mathbf{x} \approx \mathbf{b}}$

with

$$\mathbf{A} = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ \vdots & \vdots & \vdots \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{pmatrix}, \ \mathbf{x} = \begin{pmatrix} x - x_1 \\ y - y_1 \\ z - z_1 \end{pmatrix}, \ \mathbf{b} = \begin{pmatrix} b_{21} \\ b_{31} \\ \vdots \\ b_{n1} \end{pmatrix}$$

Minimizing the sum of the squares of the residuals

$$S = (\mathbf{b} - \mathbf{A}\mathbf{x})^{\mathrm{T}}(\mathbf{b} - \mathbf{A}\mathbf{x})$$

requires solving the normal equation

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

Solution method depends on the condition number of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$.

If $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is non-singular and well-conditioned then

$$\boxed{\mathbf{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}}$$

If $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is nearly-singular (poorly conditioned):

- $\star$ Compute $\boxed{\mathbf{A} = \mathbf{QR}}$

  $\mathbf{Q}$ is orthonormal matrix,

  $\mathbf{R}$ is upper-triangular matrix.

- $\star$ Solve $\boxed{\mathbf{Rx} = \mathbf{Q}^{\mathrm{T}}\vec{b}}$

  by back substitution when $\mathbf{A}$ is full rank.

The target $\theta$ is then $\boldsymbol{\theta} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{x} + \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$.

# Nonlinear Least Squares (NLSQ) Model

Minimize the sum of the squares of the errors on the distances:

$$F(\theta) = F(x, y, z) = \sum_{i=1}^{n} f_i(x, y, z)^2$$

where

$$\begin{aligned} f_i(x, y, z) &= f_i(\theta) := d_i(\theta) - r_i \\ &= \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - r_i. \end{aligned}$$

Recall: $r_i$ are the measured distances between the target $\theta = (x, y, z)$ and beacon $B_i = (x_i, y_i, z_i)$, and $n$ is the number of beacons.

Differentiating $F$ with respect to $x$ yields

$$\frac{\partial F(\theta)}{\partial x} = 2\sum_{i=1}^{n} f_i \frac{\partial f_i(\theta)}{\partial x} = 2\sum_{i=1}^{n} f_i \frac{\partial d_i(\theta)}{\partial x}.$$

The formulae for $\frac{\partial F(\theta)}{\partial y}$ and $\frac{\partial F(\theta)}{\partial z}$ are similar.

Let

$$\mathbf{f}(\theta) = \begin{pmatrix} f_1(\theta) \\ f_2(\theta) \\ \vdots \\ f_n(\theta) \end{pmatrix}, \quad \nabla F(\theta) = \begin{pmatrix} \frac{\partial F(\theta)}{\partial x} \\ \frac{\partial F(\theta)}{\partial y} \\ \frac{\partial F(\theta)}{\partial z} \end{pmatrix}$$

and define the Jacobian as

$$\mathbf{J}(\theta) = \begin{pmatrix} \frac{\partial d_1(\theta)}{\partial x} & \frac{\partial d_1(\theta)}{\partial y} & \frac{\partial d_1(\theta)}{\partial z} \\ \\ \frac{\partial d_2(\theta)}{\partial x} & \frac{\partial d_2(\theta)}{\partial y} & \frac{\partial d_2(\theta)}{\partial z} \\ \\ \vdots & \vdots & \vdots \\ \\ \frac{\partial d_n(\theta)}{\partial x} & \frac{\partial d_n(\theta)}{\partial y} & \frac{\partial d_n(\theta)}{\partial z} \end{pmatrix}$$

We must solve

$$\boxed{\nabla F(\theta) = 2\mathbf{J}(\theta)^{\mathrm{T}}\mathbf{f}(\theta) = \mathbf{0}}$$

where

$$\mathbf{J}(\theta)^{\mathrm{T}}\mathbf{f}(\theta) = \begin{pmatrix} \sum_{i=1}^{n} \frac{(x-x_i)f_i(\theta)}{d_i(\theta)} \\ \\ \sum_{i=1}^{n} \frac{(y-y_i)f_i(\theta)}{d_i(\theta)} \\ \\ \sum_{i=1}^{n} \frac{(z-z_i)f_i(\theta)}{d_i(\theta)} \end{pmatrix}.$$

# Newton-Raphson Method – Iterative Solver

Problem: Solve the scalar problem $f(x) = 0$

Solution: Newton's method:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Problem: Solve the vector problem: $\mathbf{f}(\mathbf{x}) = \mathbf{0}$

simplest case: $n$ equations, $n$ unknowns.

Solution: Newton's method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}(\mathbf{x}_k)]^{-1}\mathbf{f}(\mathbf{x}_k)$$

Apply Newton's method to $\mathbf{g}(\theta) = \mathbf{J}(\theta)^{\mathrm{T}}\mathbf{f}(\theta) = \mathbf{0}$.

Solution:

$$\boxed{\theta_{\{k+1\}} = \theta_{\{k\}} - [\mathbf{J}(\theta_{\{\mathbf{k}\}})^T\mathbf{J}(\theta_{\{\mathbf{k}\}})]^{-1}\mathbf{J}(\theta_{\{\mathbf{k}\}})^T\mathbf{f}(\theta_{\{\mathbf{k}\}})}$$

where $\theta_{\{k\}}$ denotes the $k$th estimate of the target.

A reasonably accurate initial guess, $\theta_{\{1\}}$, could be computed with the LSQ method.

Starting with $\theta_{\{1\}}$, iterate until the change $\|\theta_{\{k+1\}} - \theta_{\{k\}}\|$ is sufficiently small.

The expression for $\mathbf{J}(\theta)^T\mathbf{J}(\theta)$ is

$$\begin{pmatrix} \sum_{i=1}^n \frac{(x-x_i)^2}{d_i(\theta)^2} & \sum_{i=1}^n \frac{(x-x_i)(y-y_i)}{d_i(\theta)^2} & \sum_{i=1}^n \frac{(x-x_i)(z-z_i)}{d_i(\theta)^2} \\ \sum_{i=1}^n \frac{(x-x_i)(y-y_i)}{d_i(\theta)^2} & \sum_{i=1}^n \frac{(y-y_i)^2}{d_i(\theta)^2} & \sum_{i=1}^n \frac{(y-y_i)(z-z_i)}{d_i(\theta)^2} \\ \sum_{i=1}^n \frac{(x-x_i)(z-z_i)}{d_i(\theta)^2} & \sum_{i=1}^n \frac{(y-y_i)(z-z_i)}{d_i(\theta)^2} & \sum_{i=1}^n \frac{(z-z_i)^2}{d_i(\theta)^2} \end{pmatrix}.$$

# Mathematica Demonstration 1

## Computation of Target using the NLSQ Method

## Mathematica's NMinimize Function

## Mathematica Demonstration 2

## Computation of Target using the NLSQ Method

## Newton Iteration

# Mathematica Demonstration 3

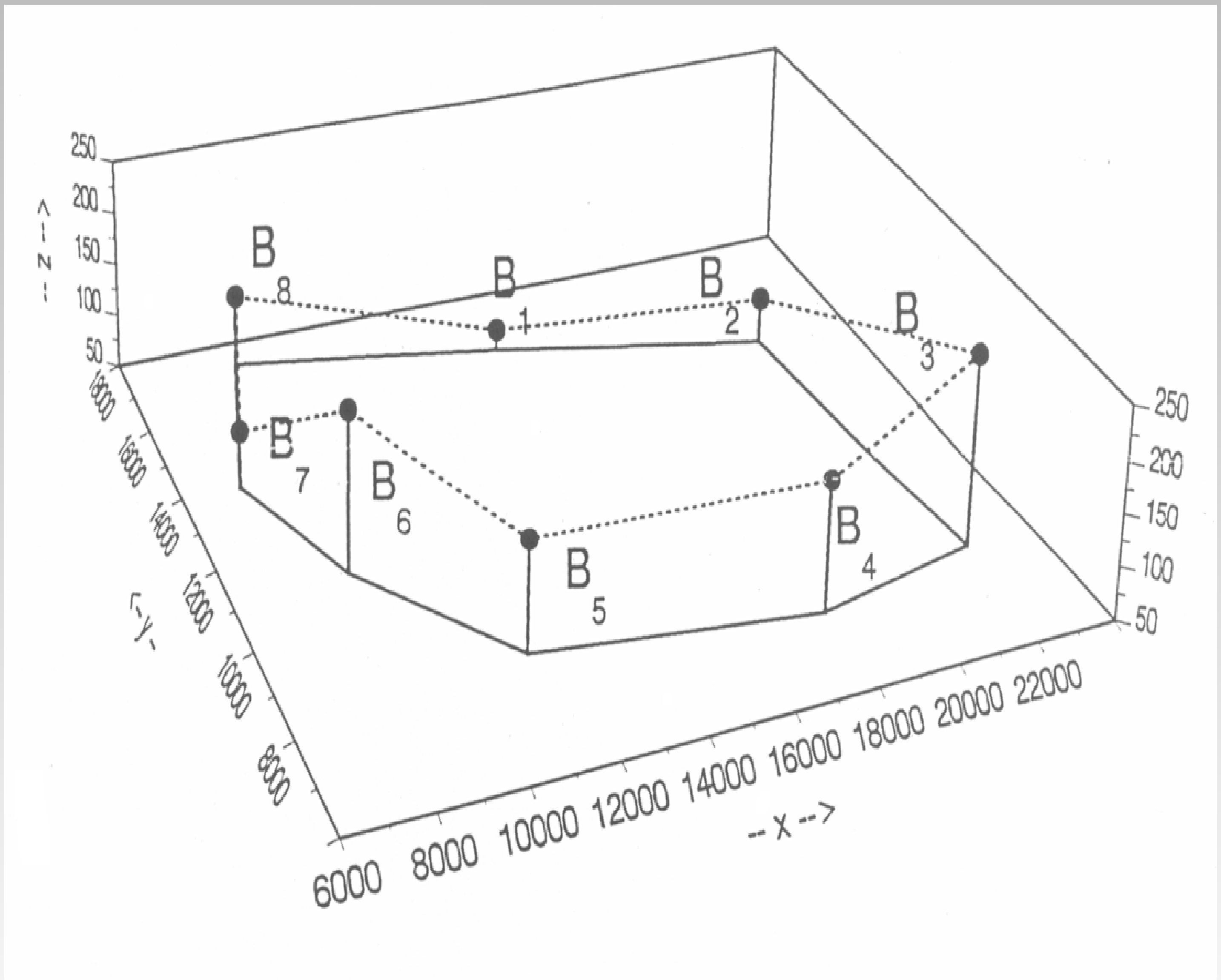
## Computation of Target using the LSQ Method

## Simulation – Results of Experiments

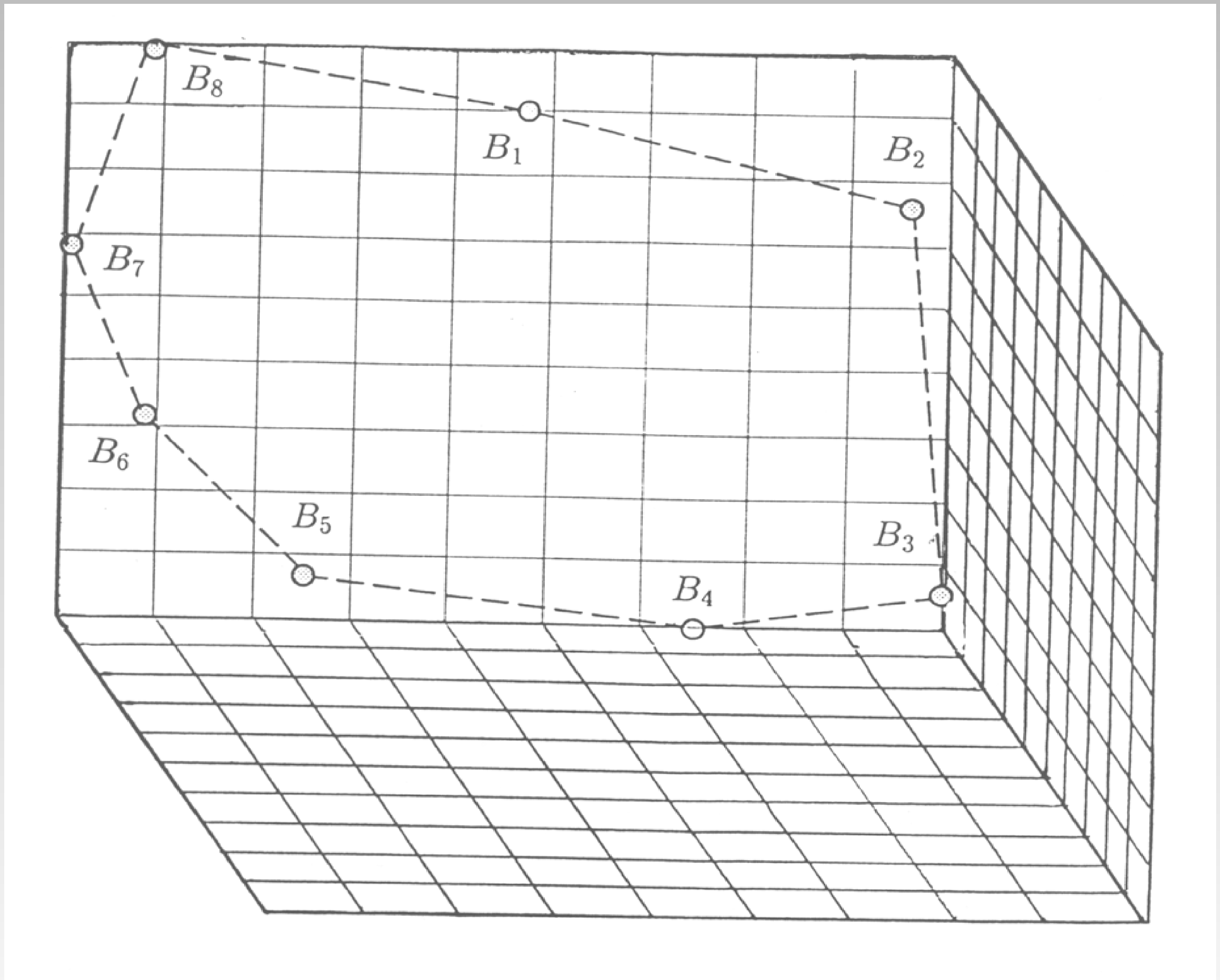
Beacon coordinates (8 beacons were used)

| $X$ | $Y$ | $Z$ |
|---|---|---|
| 920 | 3977.5 | −77.125 |
| 7360 | 2577.5 | −53.125 |
| 8090 | −3892.5 | 83.875 |
| 3910 | −4512.5 | 27.875 |
| −2710 | −3742.5 | 4.875 |
| −5420 | −1082.5 | 55.875 |
| −6740 | 1657.5 | −42.125 |
| −5410 | 5017.5 | −0.125 |

# Location of 8 Beacons



# Test Grid of 1000 Points



- Requirement: determine target within $2$ feet (distances measured within $\frac{1}{2}$ foot).
- One thousand target points on a rectangular grid.
- Top of box is 5 feet below lowest beacon.
- For each target point, $10,000$ data sets were generated.
- Each data set consisted of one measurement from each beacon.
- Each measurement was obtained by adding to the true distance a random error distributed uniformly on $(-0.5,\ 0.5)$.

- Methods were implemented in *Macsyma* and C++
- Horizontal coordinates were accurate ($98\%$ of test points).
- Vertical coordinate (height) was imprecise (off by several feet for $5\%$ of test points).
- Trouble with hardware (AccuTrack, Canada).

## Conclusions

- Exact linearization for nonlinear problem.
- LSQ method is reliable even with small samples.
- NLSQ method gives best performance.
- Methods are easy to implement.
- Good alternative for applications where GPS cannot be used.
- Publications are on the Internet:

  URL: http://inside.mines.edu/∼whereman/

Thank You

# Publications


1. W. Navidi, W. Murphy, Jr., and W. Hereman, Statistical methods in surveying by trilateration, Computational Statistics and Data Analysis, **27**(2), pp. 209-227 (1998).
2. W. Murphy and W. Hereman, Determination of a position in three dimensions using trilateration and approximate distances, Technical Report MCS-95-07, Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado (1995), 19 pages.
3. W. Murphy, Determination of a Position Using Approximate Distances and Trilateration, M.S. Thesis, Colorado School of Mines, May 1992.