L-3 Communications Clinic

Sensor Location Estimation

L-3 Communications Clinic

Sensor Location Estimation

Shawn Solberg
Paul Pryor
Joseph Petersen
Jonathan Gorzitze
Luan Nguyen

Acknowledgements

Dr. Neal Patwari
Piyush Agrawal
University of Utah Faculty
L-3 Communications

Motivation and Overview for Using Angle of Arrival in Wireless Sensor Networks

Shawn Solberg
3/27/08

Existing Networks

- GPS
  - Accurate up to 5 feet
  - 2 Dimensional
  - Chip cost upwards of $100
  - Only works outdoors

Existing Networks

- Wireless Sensor Network
  - Works on any plane (3D)
  - Chip cost as low as $1
  - Low Range of Operation
  - Works Indoors
  - Prone to Errors

MICA2 Mote

Wireless Sensor Network

Applications

- Localization
  - People
    Firefighters/Servicemen/S.W.A.T.
  - Inventory
  - Tools/Equipment

MICA2 Mote

MICA2

- Features include
  - Accelerometer
  - Microphone
  - Motion/Light Detector
  - Speaker
  - LEDs
  - etc

MICA2 Mote

How it Works

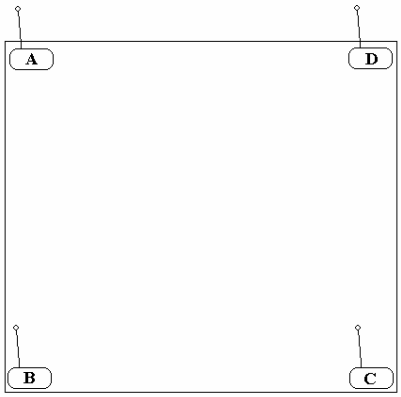
- MICA2 Motes use Received Signal Strength to determine Distance between Motes.
  - Signal Strength Decays Proportional to distance (d) and Path-Loss Exponent (n_p) Typically Between 2 and 4.[1]

$$P(d) = P_0 - 10n_p \log(d/d_0)$$

Where P_0 is the Transmitted Power in dBm at a Reference Distance d_0.

For Example:

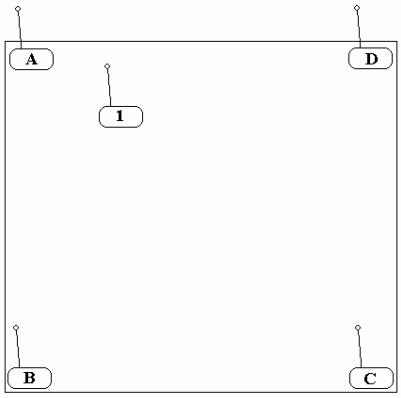
Consider a predefined area: The Locations of Nodes A, B, C, and D are known.



For Example:

Consider a predefined area:
The Locations of Nodes
A, B, C, and D are known.

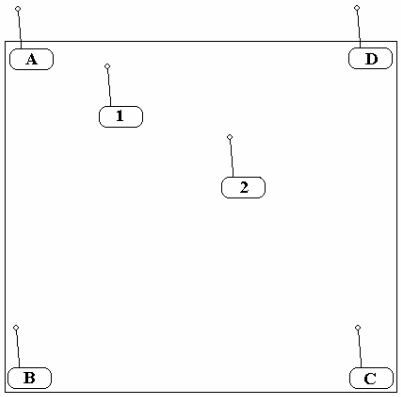
We can then locate nodes
that are randomly placed
within this area
(nodes 1,



For Example:

Consider a predefined area:
The Locations of Nodes
A, B, C, and D are known.

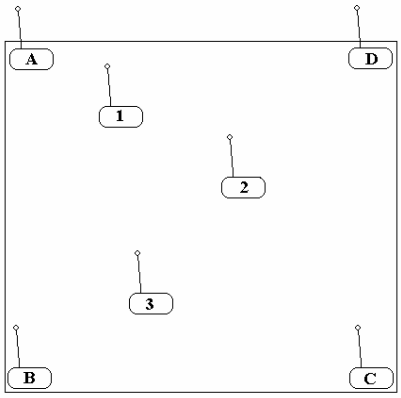
We can then locate nodes
that are randomly placed
within this area
(nodes 1, 2,



For Example:

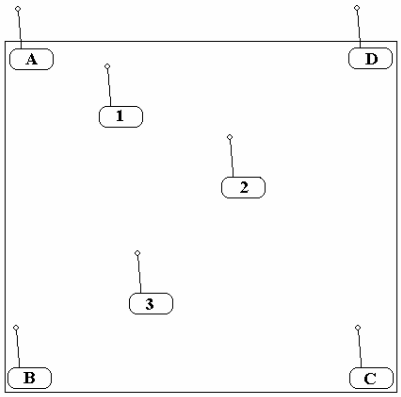
Consider a predefined area: The Locations of Nodes A, B, C, and D are known.

We can then locate nodes that are randomly placed within this area (nodes 1, 2, and 3).



The nodes do two things:

First, they transmit a signal at a constant power (or dB) level.

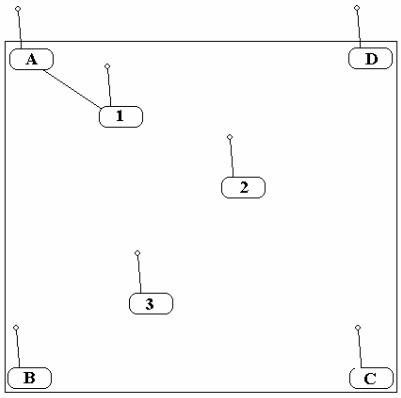


The nodes do two things:

First, they transmit a signal at a constant power (or dB) level.

Second, they use the RSS value to calculate distance.

In this way, ‘A’ knows how far away ‘1’ is,

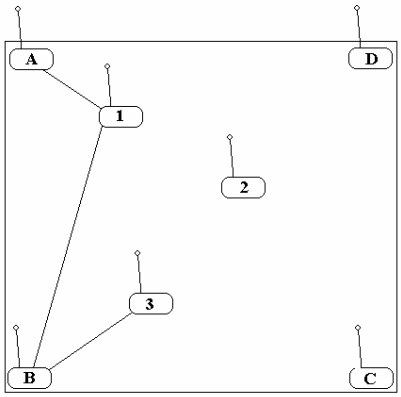


The nodes do two things:

First, they transmit a signal at a constant power (or dB) level.

Second, they use the RSS value to calculate distance.

In this way, ‘A’ knows how far away ‘1’ is, ‘B’ knows how far away ‘1’ and ‘3’ are,

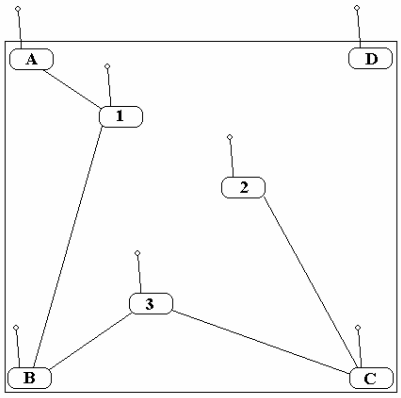


The nodes do two things:

First, they transmit a signal at a constant power (or dB) level.

Second, they use the RSS value to calculate distance.

In this way, ‘A’ knows how far away ‘1’ is, ‘B’ knows how far away ‘1’ and ‘3’ are, ‘C’ knows how far ‘2’ and ‘3’ are,

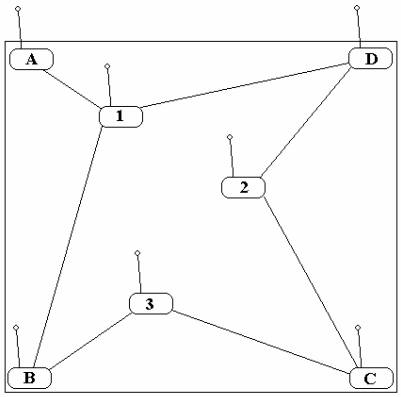


The nodes do two things:

First, they transmit a signal at a constant power (or dB) level.

Second, they use the RSS value to calculate distance.

In this way, ‘A’ knows how far away ‘1’ is, ‘B’ knows how far away ‘1’ and ‘3’ are, ‘C’ knows how far ‘2’ and ‘3’ are, and ‘D’ knows how far ‘1’ and ‘2’ are.



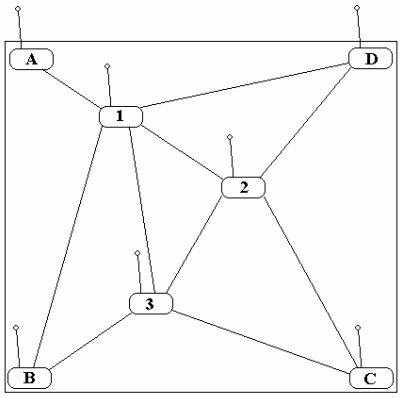
The nodes do two things:

First, they transmit a signal at a constant power (or dB) level.

Second, they use the RSS value to calculate distance.

In this way, ‘A’ knows how far away ‘1’ is, ‘B’ knows how far away ‘1’ and ‘3’ are, ‘C’ knows how far ‘2’ and ‘3’ are, and ‘D’ knows how far ‘1’ and ‘2’ are.

Nodes 1, 2 and 3 also know their distance.



How it Works

- This builds a network and ideally each node uses simple trigonometry to find its location.

Does it Work?

- The RSS-based distance values differ from one half of the actual distance to double the actual distance.
- These errors are caused by multi-path fading and still exist after implementing Frequency Hopping (900-916MHz)

Solution

- Find the Signals Angle of-Arrival
  - Phased-Array Antennas
  - Directional Antennas

Getting Started

- Break Up Responsibilities
  1. Research Directional Antennas
     - Develop a Function that calculates AoA
  2. Improve RSS Algorithm to Include AoA
     - Learn TinyOS and NesC
       - Tutorials on MICA2

Next Up

- Paul Pryor
  - Discussion of Antenna Design and Characterization for a 3-D Array
- Joseph Peterson
  - Development of Self-Localization Algorithm Based on Angle of Arrival
- Jonathan Gorzitze
  - Angle of Arrival Sensor Network: An Extension to a Received Signal Strength Sensor Network
- Luan Nguyen
  - MICA2/TinyOS: Implementation and Applications

References

- "Locating the Nodes" Neal Patwari,Joshua N. Ash,Spyros Kyperountas, Alfred O. Hero III, Randolph L. Moses, and Neiyer S. Correal

DISCUSSION OF ANTENNA DESIGN AND CHARACTERIZATION FOR 3D-SENSOR ARRAY

Paul Pryor

3/27/08

Table of Contents

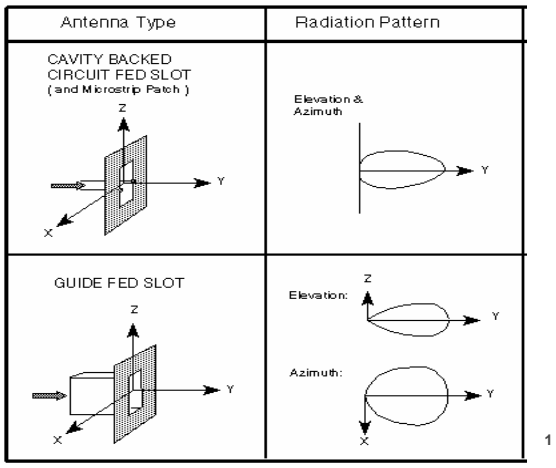
Antenna Types Considered

- Patch
  Yagi
  Log-Periodic (Special Yagi)

Patch Antenna

- Advantages
  - Small
  - Less Expensive
- Disadvantages
  - Not Off-shelf

Patch Antenna Pattern

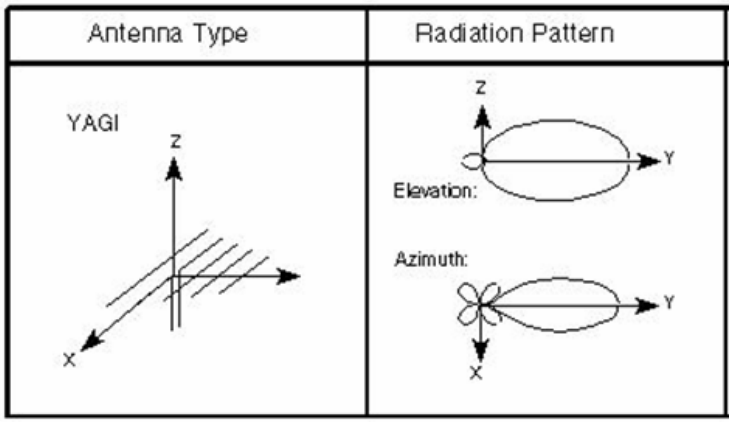


1

Yagi Antenna

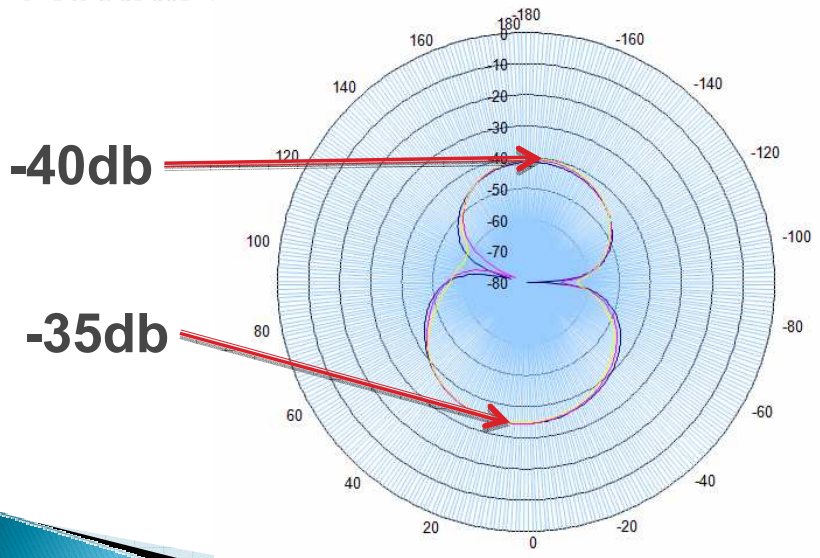
- Advantages
  - Directional
  - In Expensive
  - Small
  - Available
- Disadvantages
  - Gain Low

Yagi Antenna



1

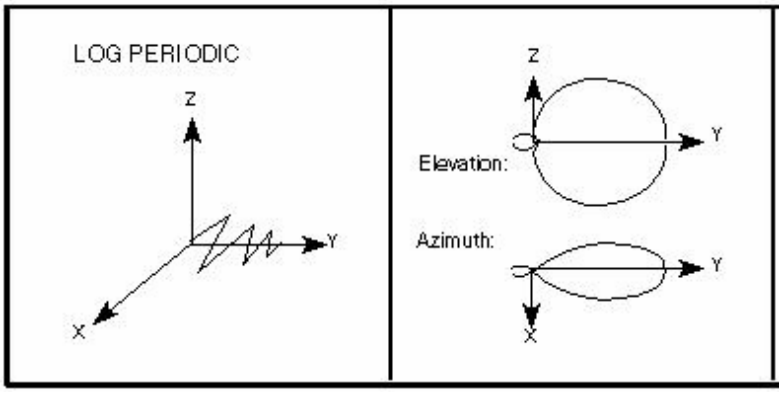
Actual Pattern



Log-Periodic

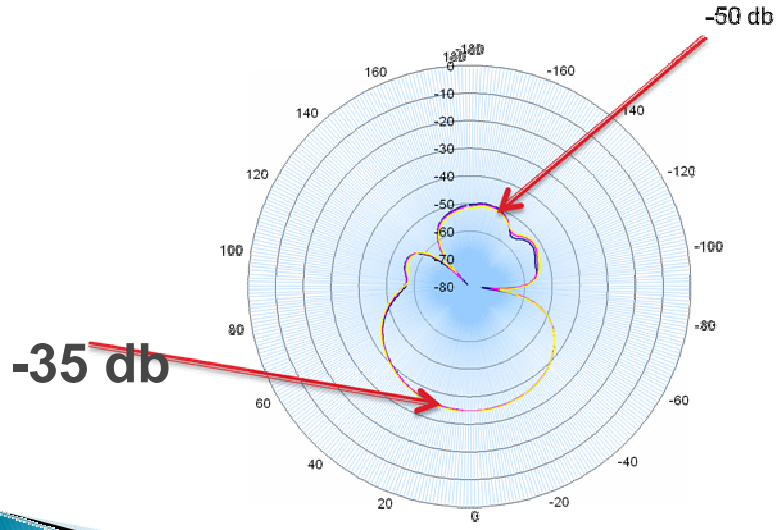
- Advantages
  - Directional
  - High Gain
- Disadvantages
  - Larger
  - Expensive

Log-Periodic



Actual Pattern

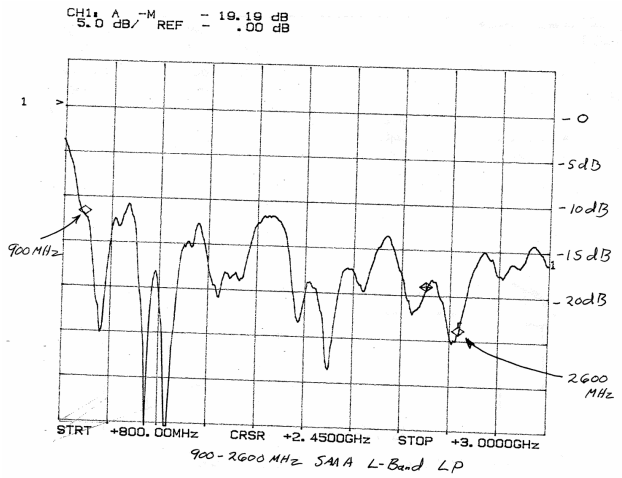
File: yagi_large.dat
Channel: Mag _B2_A1



PCB Antennas

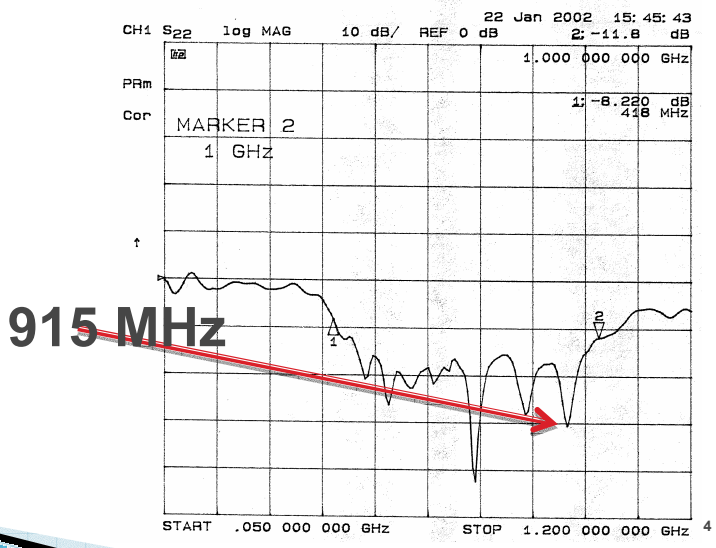
3

900 MHz to 2.4 GHz



4

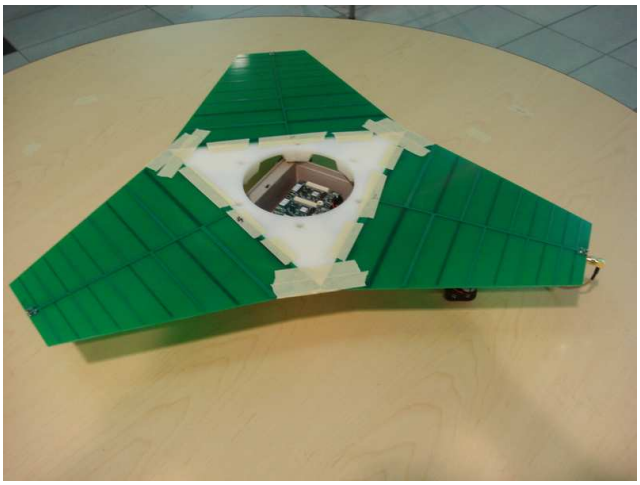
400 MHz – 1000 MHz



Module Designs

- Vertical
  Horizontal

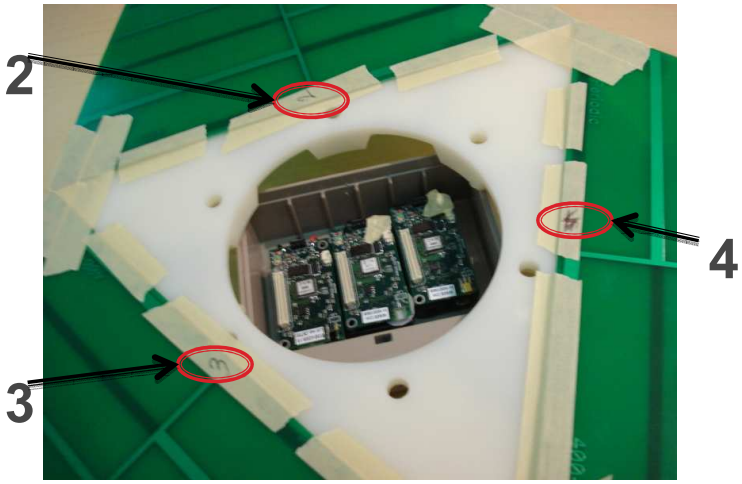
Actual Module



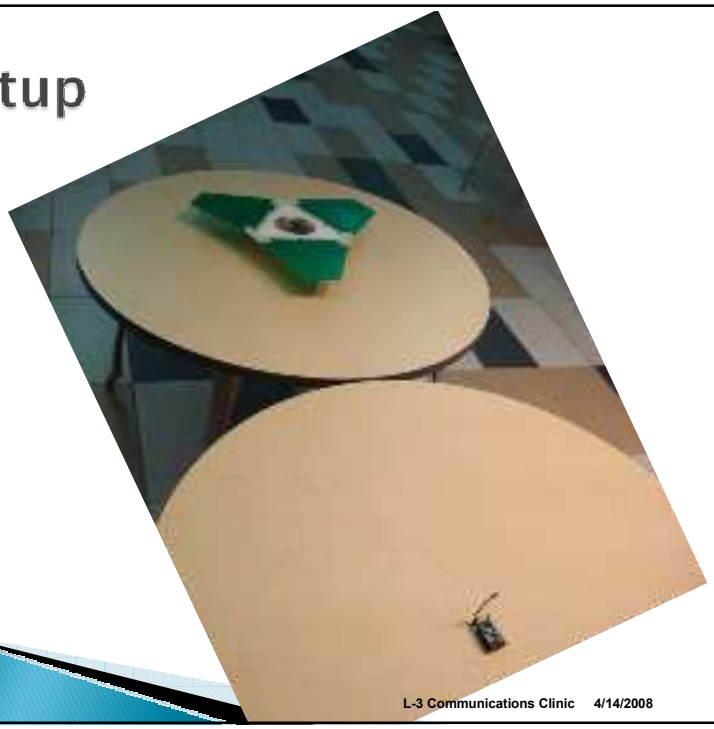
Experiment Results

- Setup
- Results

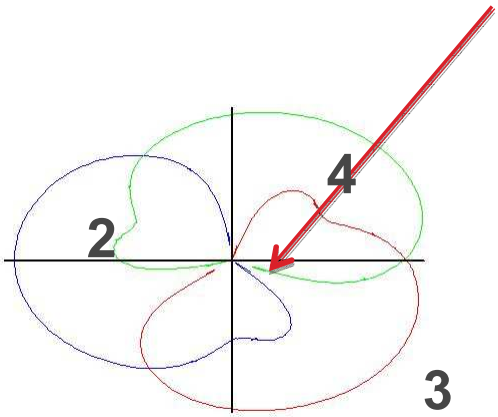
Antenna Numbers



Setup



Radiation Pattern

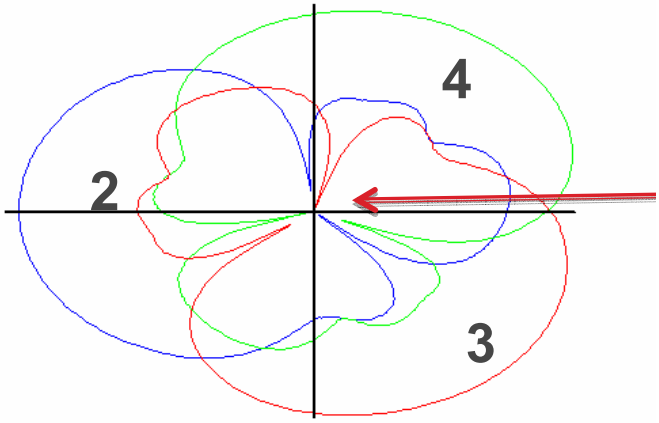


Received Path-Loss For 90°

4	3	2
21	25	26
21	24	26
21	24	26

Values are in db

30° Test

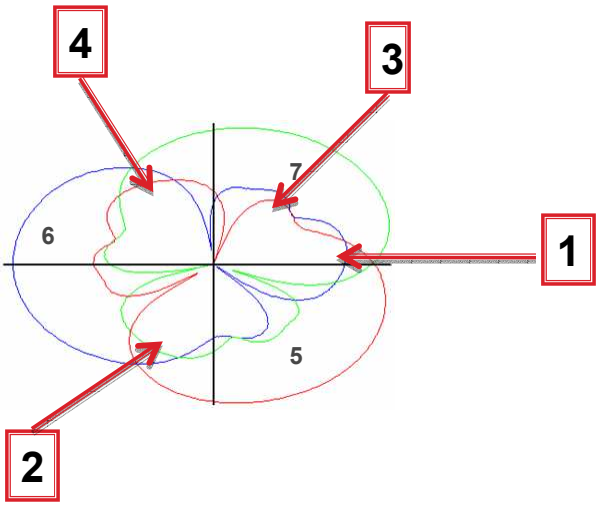


Received Path-loss For 30°

4	3	2
20	25	29
20	24	28
18	25	29

Values are in db

Final Test Setup



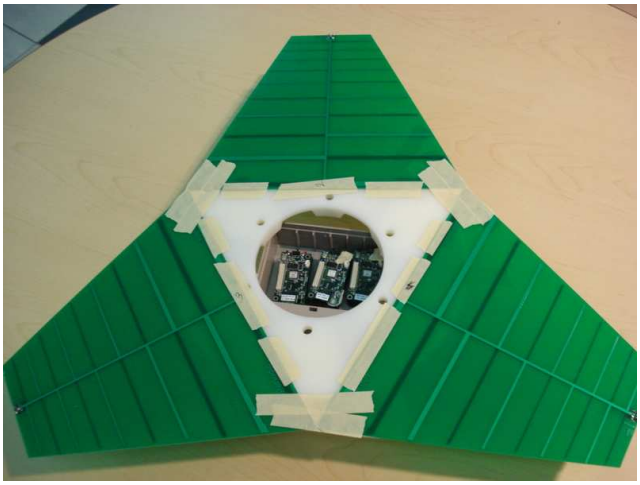
Received Path-Loss

Mote #	4			2			1		
	5	6	7	5	6	7	5	6	7
Path-Loss (db)	28	27	41	27	21	20	31	33	44

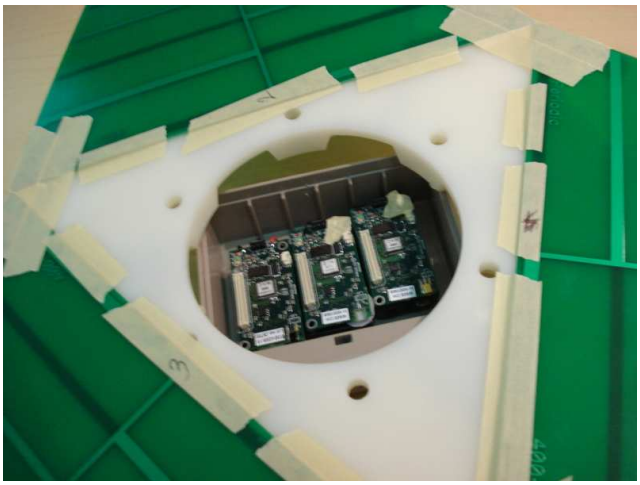
Future Considerations

- 2.4 GHz
  One IC
  All in one Package
  Patch Antenna

Antennas Too Large



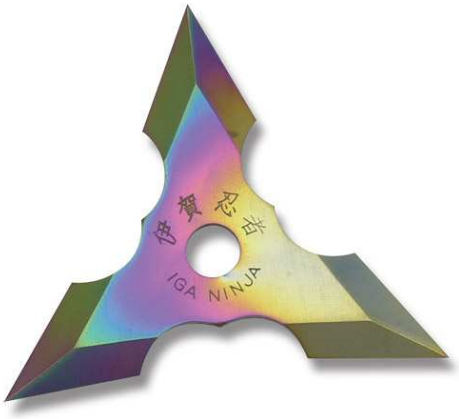
One Chip



Patch Antenna

- Advantages
  - Small
  - Less Expensive
- Disadvantages
  - Not Off-shelf

The Future Module

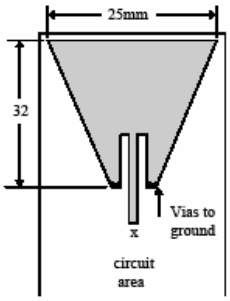


5

Patch Antenna

Trapazoidal Patch:

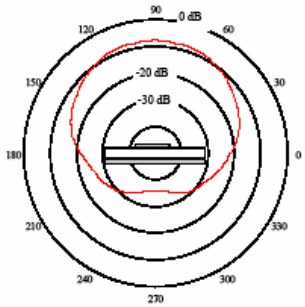
916.5 MHz



AN36
AN36A-070898 Page 13

wireless world ag Im Gree 79, CH-8566 Ellighausen, Fon +41(71)698 6480, Fax +41(71)698 6481, e-mail: info@awirelessworld.ch, www.wirelessworldag.com 2

Patch Antenna Pattern



Trapazoidal Patch over a
Large Ground Plane
(916.5 MHz)

AN36
AN36A-070898 Page 14

Im Gree 79, CH-8566 Ellighausen, Fon +41(71)698 6480, Fax +41(71)698 6481, e-mail: info@awirelessworld.ch, www.wirelessworldag.com 2

Conclusion

- Project Review
- Antenna and Module
- Future Design

References

1. http://www.kyes.com/antenna/navy/rpatterns/antena11.gif
2. http://www.numatechnologies.com/pdf/foilantennas.pdf
3. http://www.ramseyelectronics.com/cgi-bin/commerce.exe?preadd=action&key=LPY2
4. http://www.wa5vjb.com/products1.html
5. http://eknifeworks.com/webapp/eCommerce/prodlist.jsp?Mode=Cat&Cat=121

Questions?

Design of Self-Localization Algorithm & MATLAB Platform for Testing

Joseph Peterson
3/27/08

Table of Contents

Self Localization
Phase1: Data Reception

Each Node obtains

- Reported Coordinates
- Distance/Angle Measurements
  For Other Nodes in Group

Self Localization
Phase2: Optimization

Algorithm Compares

- Reported Locations with
  Predicted Locations
  - Based on Measurements
  - & Node's Assumed Position
- Node Re-Estimates Own Coordinates

Self Localization
Phase3: Transmission

Node Broadcasts
New, Better Estimate
of Its Own Coordinates

Self Localization
Phase4: Convergence

All Nodes Simultaneously
Revise Own Coordinates
& Broadcast.

After Many Iterations,
All Nodes in Network Located.

Self Localization Assumptions

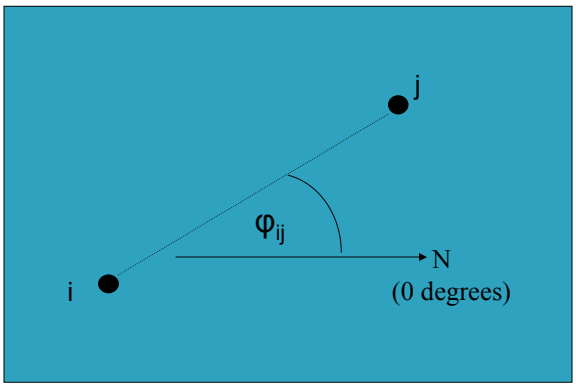
- 2-Dimensional
- Stationary
- Some Nodes Pre-Localized (Programmed w/ Own Coordinates)

Strategy

Cost Function = Σ (Reported-Predicted)^2

Solution: Find Coordinates to Minimize Cost Function

Angle Between Two Nodes Must Assume a Reference "North"

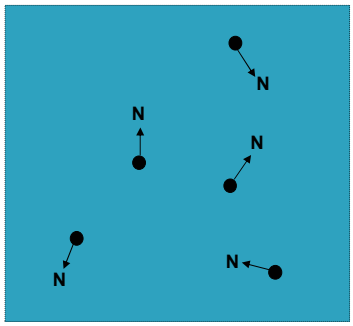


Each Node has its Own Reference "North"

Disadvantage:

Conflicting Assumptions

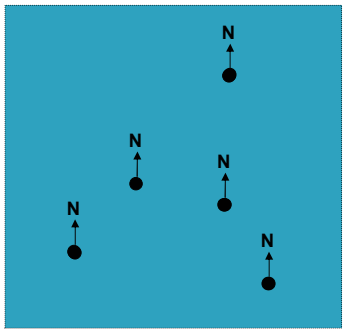
= Conflicting Estimates



All Nodes Standardize to the Same Reference "North"

Disadvantages:

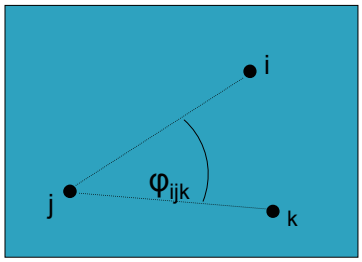
1. Extra Hardware
2. New Source of Error



Triplets: Angle Between 3 Nodes do Not Need Reference "North"

Advantages:

1. Simpler Hardware
2. No Orientation Error



Triplets:
First Disadvantage-
Combinatorial Complexity

N = # nodes in group

Pairs: N*(N-1)

Triplets: N*(N-1)*(N-2)

Triplets:
Second Disadvantage-
Intricate Computation

Algorithm Developed

But Dead End

Not Implemented in MATLAB

First Try: Triplets

Local Cost Function:

$$S_q = \sum_{t=1}^{T} \left(\begin{array}{l} \sum_{\substack{i=1 \\ i \neq q}}^{N+M-1} \sum_{\substack{k=i+1 \\ k \neq q}}^{N+M} w_{iqk}{}^{t} \left(\cos \varphi_{iqk}{}^{t} - \cos \theta_{iqk} \right)^2 \\ + \sum_{j=N+1}^{N+M} \sum_{\substack{k=q+1 \\ k \neq q}}^{N+M} w_{qjk}{}^{t} \left(\cos \varphi_{qjk}{}^{t} - \cos \theta_{qjk} \right)^2 \end{array} \right)$$

First Try: Triplets

Introduce Some Short-hand:

$$\cos \theta_{ijk} = \frac{\left(\vec{x}_i - \vec{x}_j \right) \cdot \left(\vec{x}_k - \vec{x}_j \right)}{\delta_{ij}{}^{t} \delta_{kj}{}^{t}} = \frac{L_{ijk}}{\delta_{ij}{}^{t} \delta_{kj}{}^{t}}$$

$$G_{ijk} = \beta_{ijk}{}^{2} = \sum_{t=1}^{T} \frac{w_{ijk}{}^{t}}{\left(\delta_{ij}{}^{t} \delta_{kj}{}^{t} \right)^2}$$

$$H_{ijk} = \alpha_{jk} \beta_{ijk} = \sum_{t=1}^{T} \frac{w_{iqk}{}^{t} \cos \varphi_{iqk}{}^{t}}{\delta_{iq}{}^{t} \delta_{kq}{}^{t}}$$

First Try: Triplets

Differentiate to Minimize:

$$\frac{\partial S_q}{\partial x_q} = 2\sum_{j=N+1}^{N+M}\sum_{\substack{k=q+1\\k\neq q}}^{N+M}\begin{pmatrix} -H_{qjk}\left[x_k - x_j\right] \\ +G_{qjk}\left[\left(x_k - x_j\right)^2 x_q - x_j\left(x_k - x_j\right)^2\right]\end{pmatrix}$$

$$+2\sum_{\substack{i=1\\i\neq q}}^{N+M-1}\sum_{\substack{k=i+1\\k\neq q}}^{N+M}\begin{pmatrix} -H_{iqk}\left[x_q^{\ 2} - (x_i + x_k)x_q + x_i x_k\right] \\ +G_{iqk}\begin{bmatrix} 2x_q^{\ 3} - 3(x_i + x_k)x_q^{\ 2} + \\ (x_i^{\ 2} + 4x_i x_k + x_k^{\ 2})x_q - x_i x_k (x_i + x_k)\end{bmatrix}\end{pmatrix}$$

First Try: Triplets

Derivative is a Cubic Polynomial:

$$\frac{1}{2}\frac{\partial S_q}{\partial x_q} = Ax_q^{\ 3} + Bx_q^{\ 2} + Cx_q + D$$

A = 2 r1
B = -3 r2
C = r3 - 2 r4 + r5
D = - r6 - r7 + r8 - r9

$$r_1 = \sum_{\substack{i=1\\i\neq q}}^{N+M-1}\sum_{\substack{k=i+1\\k\neq q}}^{N+M} G_{iqk}$$

$$r_2 = \sum_{\substack{i=1\\i\neq q}}^{N+M-1}\sum_{\substack{k=i+1\\k\neq q}}^{N+M} G_{iqk}\left(x_i + x_k\right)$$

$$r_3 = \sum_{\substack{i=1\\i\neq q}}^{N+M-1}\sum_{\substack{k=i+1\\k\neq q}}^{N+M} G_{iqk}\left(x_i^{\ 2} + 4x_i x_k + x_k^{\ 2}\right)$$

$$r_4 = \sum_{\substack{i=1\\i\neq q}}^{N+M-1}\sum_{\substack{k=i+1\\k\neq q}}^{N+M} H_{iqk}$$

$$r_5 = \sum_{j=N+1}^{N+M}\sum_{\substack{k=q+1\\k\neq q}}^{N+M} G_{qjk}\left(x_k - x_j\right)$$

$$r_6 = \sum_{j=N+1}^{N+M}\sum_{\substack{k=q+1\\k\neq q}}^{N+M} H_{qjk}\left(x_k - x_j\right)^2$$

$$r_7 = \sum_{j=N+1}^{N+M}\sum_{\substack{k=q+1\\k\neq q}}^{N+M} G_{qjk}x_j\left(x_k - x_j\right)$$

$$r_8 = \sum_{\substack{i=1\\i\neq q}}^{N+M-1}\sum_{\substack{k=i+1\\k\neq q}}^{N+M} H_{iqk}\left(x_i + x_k\right)$$

$$r_9 = \sum_{\substack{i=1\\i\neq q}}^{N+M-1}\sum_{\substack{k=i+1\\k\neq q}}^{N+M} G_{iqk}x_i x_k\left(x_i + x_k\right)$$

First Try: Triplets

Solve Using Lagrange Resolvents:

Discriminant of Polynomial:

Δ = 4B3D – B2C2 + 4AC3 – 18ABCD + 27A2D2

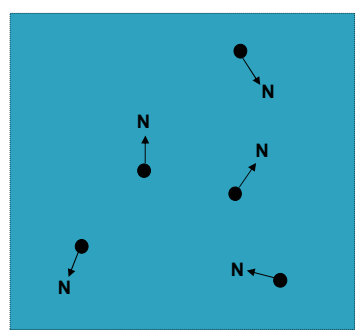
Let $$z_1, z_2 = \frac{9}{2}\frac{BC}{A^2} - \left(\frac{B}{A}\right)^3 - \frac{27}{2}\frac{D}{A} \pm \frac{3}{2}\sqrt{3\Delta}$$

Root: $$p_1 = \frac{1}{3}\left(-\frac{B}{A} + z_1^{1/3} + z_2^{1/3}\right)$$

Back to Square One… Solution:

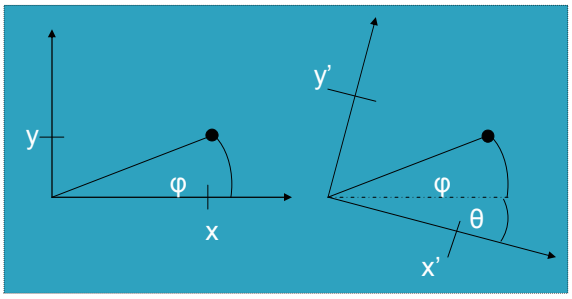
Each Node Guesses

1. Its Own Coordinates
2. *Its Own Orientation*



Second Try: Pairs

Clockwise Rotation of Axes



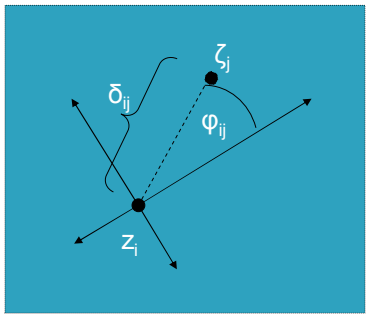
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = Q \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Second Try: Pairs

φ_{ij}	Measured Angle Between "i" and "j"
δ_{ij}	Measured Distance Between "i" and "j"
z_j	Reported Coordinates for Node "j"
z_i	Last Estimate for Node "i" Itself
$Z=[z_1\ z_2\ \ldots\ z_n]$	All Reported Coordinates (z_i Not Included)
θ_i	Orientation Bias for Node "i"
r_j	Location of Node "j" Relative to Node "i"
$R=[r_1\ r_2\ \ldots\ r_n]$	All Locations Relative to Node "i"
ζ_j	Predicted Coordinate for Node "j"
Q	Rotation Matrix for Clockwise θ_i
h	Scaling Factor (Accounts for Distortion)
S_i	Cost Function for Node "i"

Second Try: Pairs

Node "i" Takes Measurements On Node "j"



$$r_j = \begin{bmatrix} \delta_{ij} \cos\varphi_{ij} \\ \delta_{ij} \sin\varphi_{ij} \end{bmatrix}$$

$$\varsigma_j = z_i + h \cdot Q \cdot r_j$$

Second Try: Pairs

Cost Function: $S_i = \sum_j \left| z_j - \varsigma_j \right|^2$

Matrix Form:

$$S_i = \left\| Z - h \cdot Q \cdot R - z_i \cdot [1\ 1\ 1 \ldots 1\ 1] \right\|_F^2$$

Differentiate & Use Averaging to Extract:

$$z_i \cdot [1\ 1\ 1 \ldots 1\ 1] = \left(Z - \overline{z_j} \right) - h \cdot Q \cdot \left(R - \overline{r_j} \right)$$

Second Try: Pairs

Singular Value Decomposition:

$$U \cdot K \cdot V' = \left(R - \overline{r_j}\right) \cdot \left(Z - \overline{z_j}\right)'$$

Find Rotation Matrix & Scaling Factor:

$$Q = V \cdot U' \qquad h = \frac{trace(K)}{trace(R \cdot R')}$$

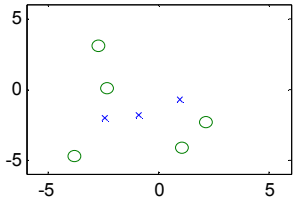
Can Now Compute New zi

MATLAB Platform

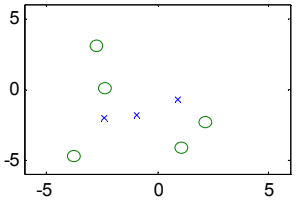
Reasons to Use Matlab

- No Bugs from TinyOS Code
- No Errors from Antennas or Other Hardware
- Verify Algorithm Converges Correctly
- Automatically Store Data
- Investigate Modifications to Algorithm
- Perform Controlled Statistical Testing

actual coordinates

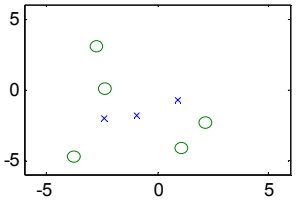


Old Algorithm



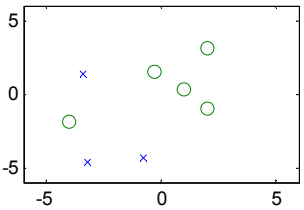
49 iterations

New Algorithm

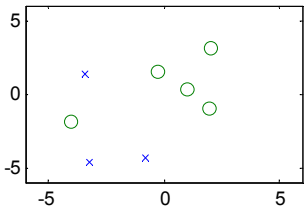


16 iterations

actual coordinates

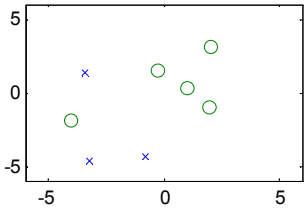


Old Algorithm



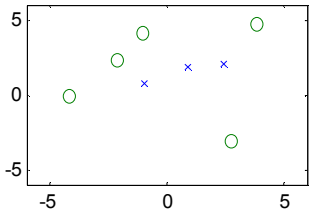
59 iterations

New Algorithm

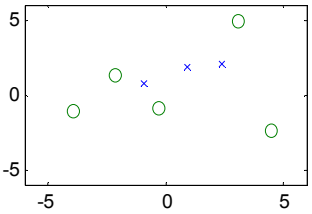


11 iterations

actual coordinates

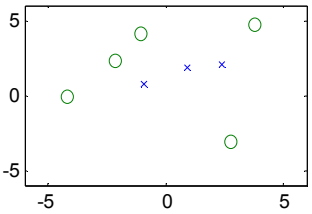


Old Algorithm



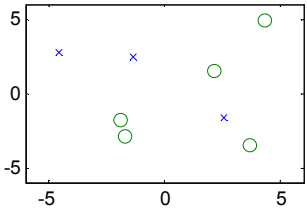
1000 iterations

New Algorithm

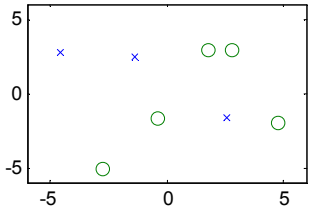


21 iterations

actual coordinates

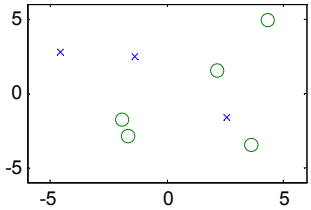


Old Algorithm

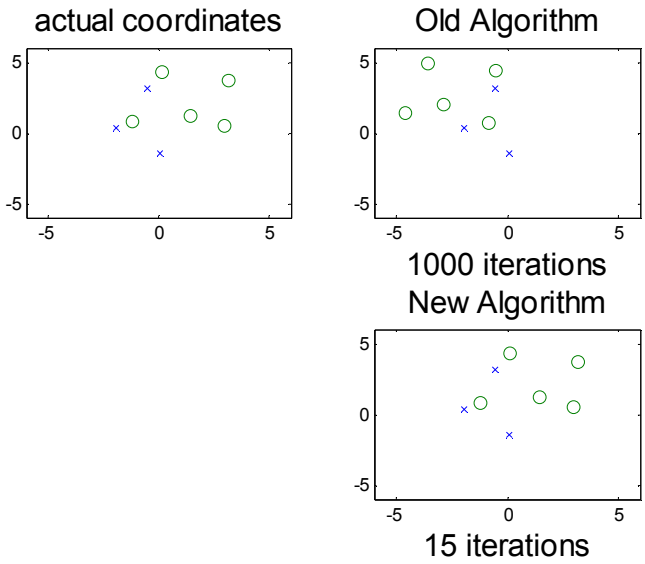


1000 iterations

New Algorithm



17 iterations



Further Research

- Fixed Patterns for Pre-Localized Nodes
- How to Choose Neighbors
- Weighting Coefficients
- Interface w/ Nodes that Use Old Algorithm
- How to Isolate a Malfunctioning Node

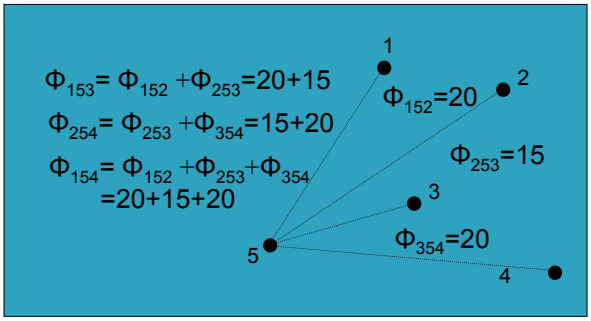
Further Research: Reinvestigate Triplets

Why?

- More Angles Available = More Flexibility
- Use Redundant Triplets To Correct Error
- Isolate Malfunctioning Node

Further Research: Reinvestigate Triplets

Use Redundant Triplets To Correct Error



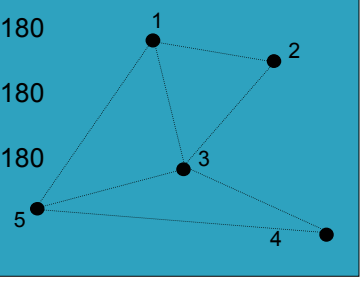
Further Research: Reinvestigate Triplets

Use Redundant Triplets To Correct Error

$\Phi_{513}+\Phi_{135}+\Phi_{351}=180$

$\Phi_{312}+\Phi_{123}+\Phi_{231}=180$

$\Phi_{534}+\Phi_{345}+\Phi_{453}=180$



References

- *Locating the Nodes: Cooperative Localization in Wireless Sensor Networks* by Patwari, Ash, Kyperountas, Hero, Moses, Correal. Printed in *IEEE Signal Processing Magazine*, July 2005, pp. 54-69.

ANGLE OF ARRIVAL ANALYSIS AND EXTENSION TO RECEIVED SIGNAL STRENGTH SENSOR NETWORK

Jonathan Gorzitze
3/27/08

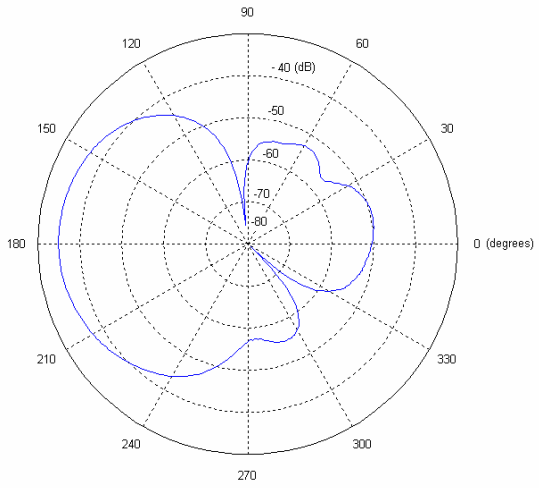
Outline:

- Angle of Arrival Analysis
- nesC Implementation for Angle Calculation
- Simulations
  - Distance Only Measurements
  - Distance and Angle Measurements
- Conclusion

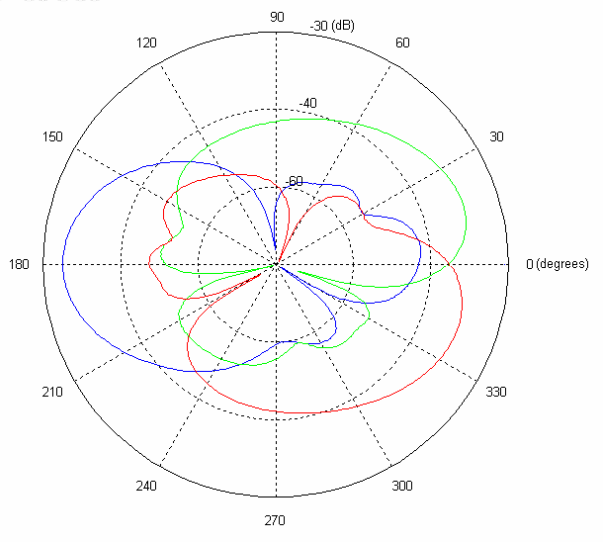
Get Angle Code Analysis

- Get Data
  Create Gain Pattern
  Find Angle

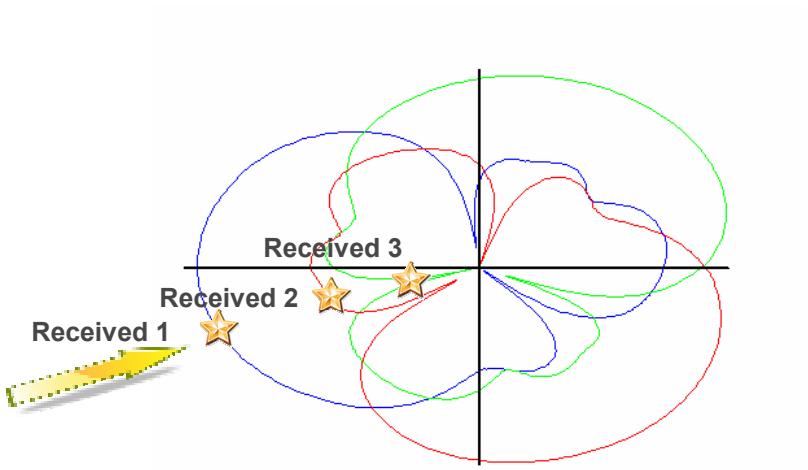
Get Data



Get Data

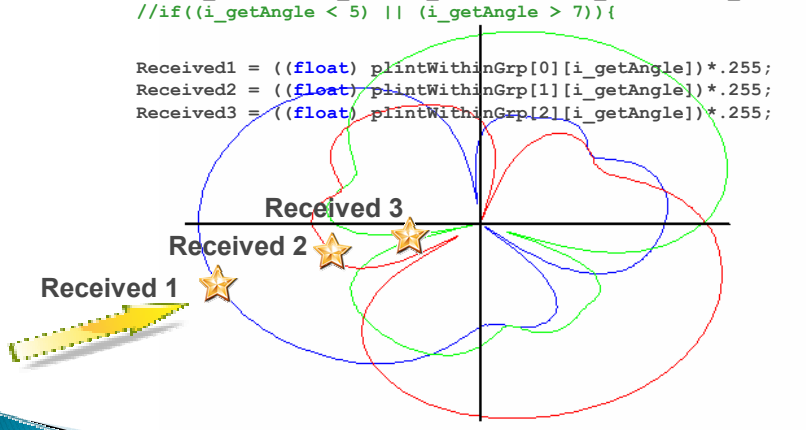


Get Data



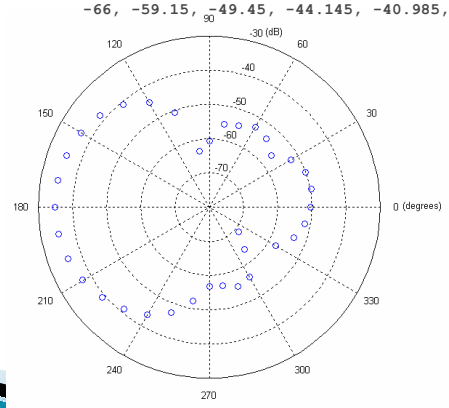
Get Data

```
for(i_getAngle=0;i_getAngle<TOTAL_NODES;i_getAngle++){
        if((i_getAngle < NODES_DIRECTIONAL_ANTENNA_START) || (i_getAngle >
        (NODES_DIRECTIONAL_ANTENNA_START + MAXNUM_DIRECTIONAL_ANTENNA -1)))
        //if((i_getAngle < 5) || (i_getAngle > 7)){

        Received1 = ((float) plintWithinGrp[0][i_getAngle])*.255;
        Received2 = ((float) plintWithinGrp[1][i_getAngle])*.255;
        Received3 = ((float) plintWithinGrp[2][i_getAngle])*.255;
```



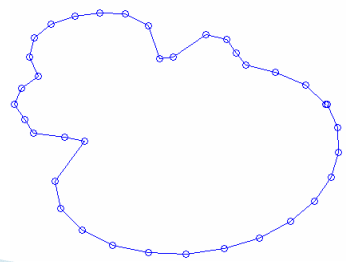
Find Gain of Antenna

```
task void getAngle(){
        uint16_t i_getAngle,k_getAngle,z_getAngle, angle_by_ten, rem_by_ten, low_index;
        int8_t  N = 36;      // number of times to iterate between points in gain arrays
        uint16_t  inc = 2;   // degree incrementation size
        float sqr_error, least_sqr_error = 260000.0;    //largest possible error

        float Gain1_meas[] = {-66, -59.135, -49.465, -44.145, -40.985, -39.95, -36.945,
                              -35.8175, -35.13, -34.88, -35.13, -35.8175, -36.945 ...};
        float Gain2_meas[] = {-36.945, -39.95, -40.985, -44.145, -49.465, -59.135, -66,
                              -58, -57, -56, -55, -54, -53, -52, -51, -50, -51, -52...};
        float Gain3_meas[] = {-53, -52, -51,-50, -51, -52, -53, -54, -55, -56, -57, -58,
                              -66, -59.15, -49.45, -44.145, -40.985, -39.95, -36.945...};
```

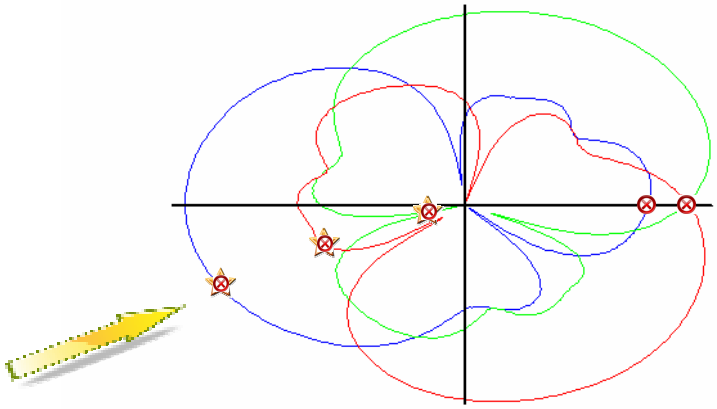


Find Gain of Antenna

```
//create array for each slope point
for (k_getAngle = 0; k_getAngle < N; k_getAngle++){

slope1[k_getAngle] = (Gain1_meas[k_getAngle+1] - Gain1_meas[k_getAngle])/(360/N);
slope2[k_getAngle] = (Gain2_meas[k_getAngle+1] - Gain2_meas[k_getAngle])/(360/N);
slope3[k_getAngle] = (Gain3_meas[k_getAngle+1] - Gain3_meas[k_getAngle])/(360/N);     }

for (z_getAngle = 0; z_getAngle <= 358; z_getAngle+=2){
         angle_by_ten = z_getAngle / 10;
         rem_by_ten   = z_getAngle % 10;    //remainder term

//Calculate the Gain value at every two degrees
Gain1 = Gain1_meas[angle_by_ten] + slope1[angle_by_ten]*rem_by_ten;
Gain2 = Gain2_meas[angle_by_ten] + slope2[angle_by_ten]*rem_by_ten;
Gain3 = Gain3_meas[angle_by_ten] + slope3[angle_by_ten]*rem_by_ten;
```



Find Angle



Received Signal Strength = Gain Pattern(⊗) – Received Signal(☆) ~ 0

Difference = Received Signal Strength(1) – Received Signal Strength(2) ~ 0

Squared Error = Difference(12)^2 + Difference(23)^2 ~ 0

Find Angle

```
RSS1 = -Gain1 + Received1;
RSS2 = -Gain2 + Received2;
RSS3 = -Gain3 + Received3;

diff_12 = RSS1 - RSS2;
diff_23 = RSS2 - RSS3;
diff_31 = RSS3 - RSS1;

sqr_error = (diff_12*diff_12) + (diff_23*diff_23);
        if(sqr_error < least_sqr_error){
                least_sqr_error = sqr_error;
                        low_index = z_getAngle;
                }
        }
        AngleToEachNode[i_getAngle] = (float) low_index*inc;
```

Simulations

Setup
Distance Only Algorithm
Angle and Distance Algorithm

Simulation Setup:

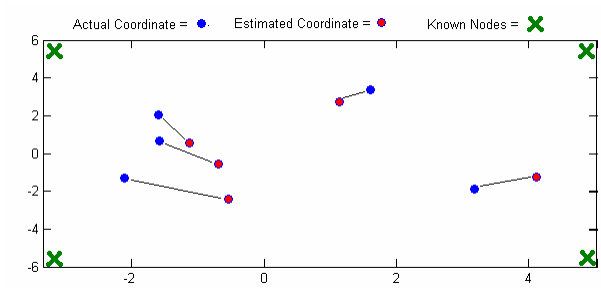
Implemented testing for:

- Accuracy of self-localization
- Time to self-localization

Introduced three error terms:

- Angle Error
- Distance Error
- Noise term (Jitter)

Simulation Setup:



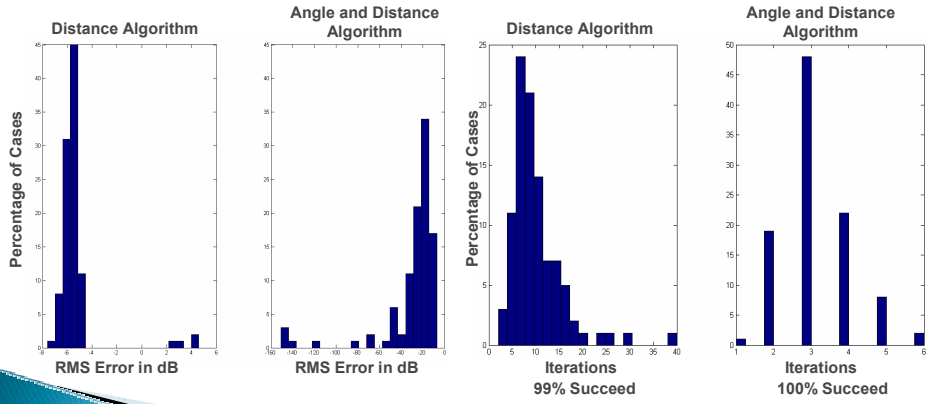
$$\text{Root Mean Squared} = \sqrt{\frac{\sum_{n} (x_{actual} - x_{est.})^2 + (y_{actual} - y_{est.})^2}{total\ motes}}$$

Self-Localization

No Angle Error, No Distance Error, No Noise error

Accuracy

Iterations Until RMS <1m

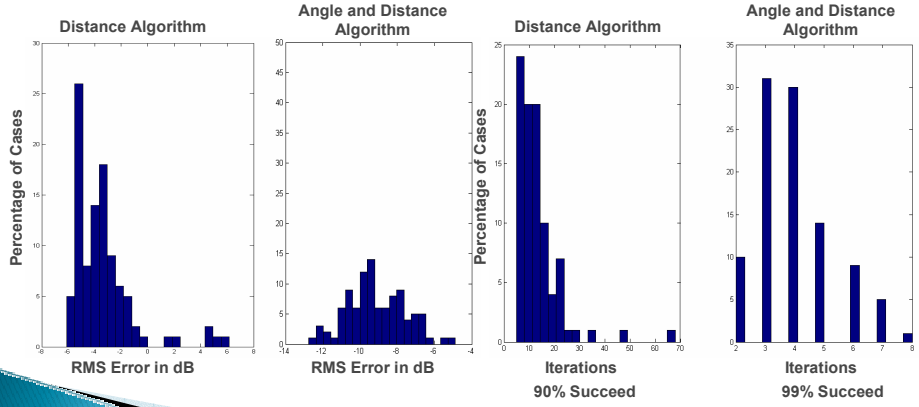


Self-Localization

5% Distance Error, No Angle Error, No Noise error

Accuracy

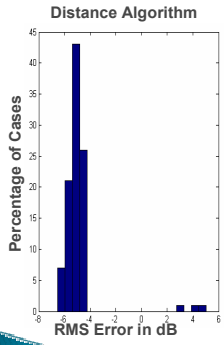
Iterations Until RMS <1m

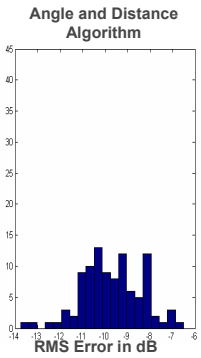


Self-Localization

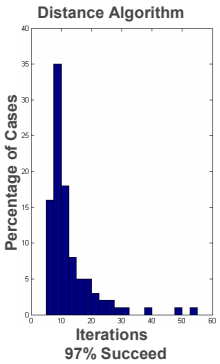
No Angle Error, No Distance Error, 5% Noise error

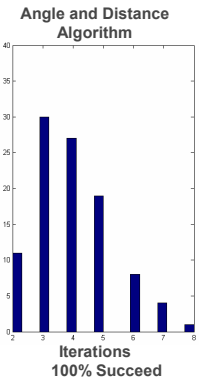
Accuracy





Iterations Until RMS <1m

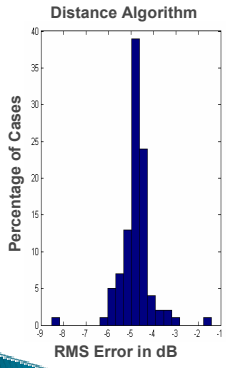


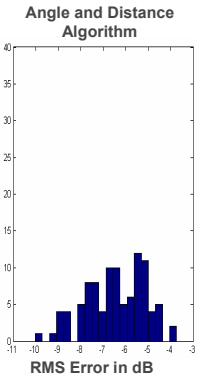


Self-Localization

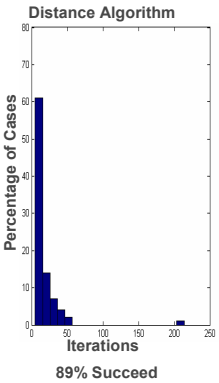
5° Angle Error, 5% Distance Error, 5% Noise error

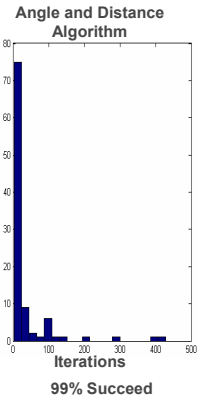
Accuracy





Iterations Until RMS <1m

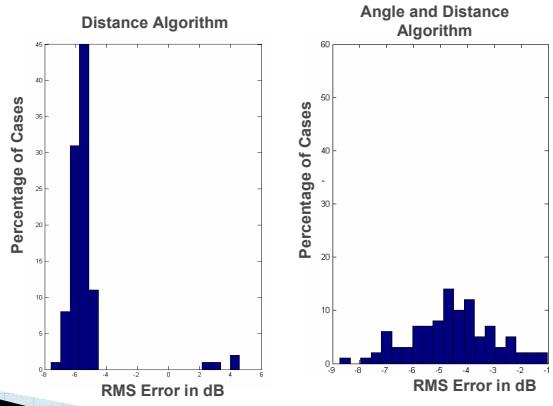




Self-Localization

How much angle error can be tolerated until angle measurements no longer reflect higher accuracy over distance only measurements?

8° Angle Error



Conclusion

- In the absence of noise terms, the Angle and Distance algorithm out performs the Distance Only algorithm by converging approximately five times quicker and approximately one order of magnitude higher accuracy (after 500 iterations).
- When Distance Errors and noise errors are introduced, the Angle and Distance algorithm outperforms the Distance Only algorithm by converging approximately four time quicker and 3-4 times more accurately.
- The Angle and Distance algorithm outperforms the Distance Only algorithm until an angle error of 8° or greater is measured.

Questions:

MICA2 **AND** TinyOS

IMPLEMENTATION **AND** APPLICATION

Luan Nguyen
3/27/08

Table of Contents

Mica2

a. Overview
b. Components
c. MIB510

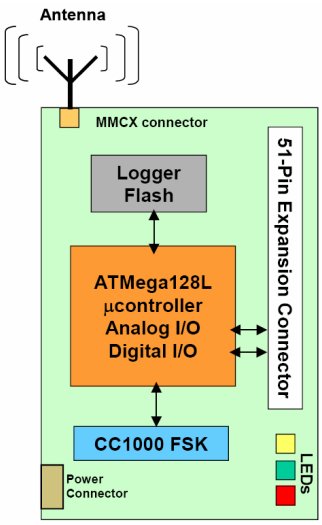
Mica2

- Overview
  - Third generation MPR (Mote Processor Radio)
  - Usage

(1)

Mica2

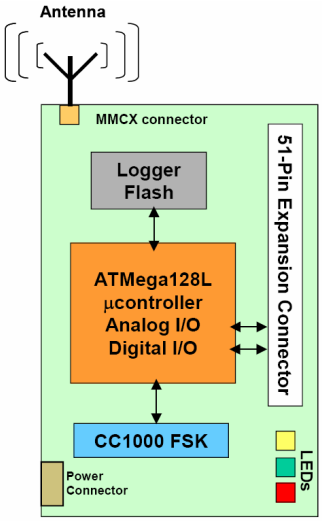
- Design
  - Data Flash Logger
  - Micro Controller
  - Radio
  - Expansion Connector
  - Antenna



(2)

Mica2

- Advantages
  - Small, compact
  - Power efficiency
  - Economical
- Disadvantages
  - Lack of User Interface for diagnosing and debugging
  - Interaction between programming software and motes

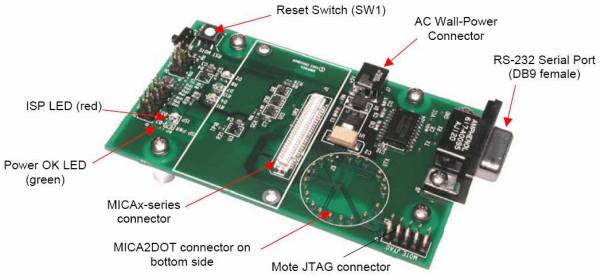


(2)

MIB510

- Mote Interface Board
  - On board ISP (In System Processor)
  - Port RS 232 to USB port
  - Baud rate of 115.2 kbaud
  - 5 leds : power, isp and 3 MPR leds
  - Programming interface for the motes

(2)



TinyOS

Overview
Cygwin

- Overview
  - Embedded Operating System
  - Written in NesC
  - Created by UCBerkley and Intel

- NesC
  - Embedded Operating System
  - Written in NesC
  - Overall similar with C
  - More specified declaration of vairables

NesC	C
Uint16_t	integer
Uint8_t	

  - Limit the memory allocated for each variable

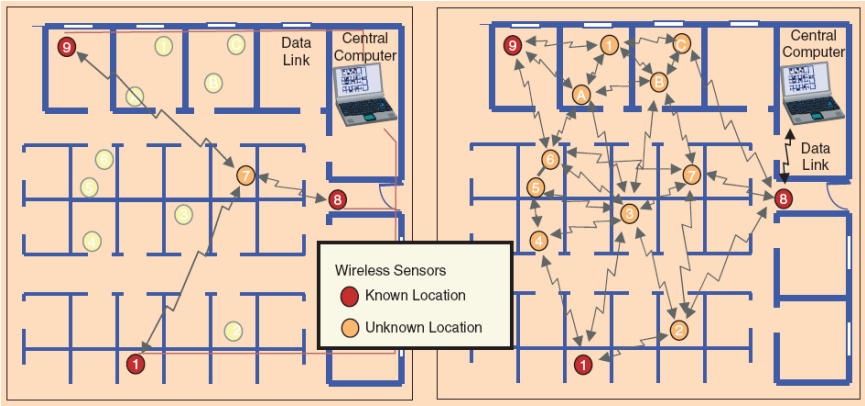
CygWin

- CygWin
  - Provide Linux Inside Windows
  - NesC is compiled inside Linux environment
  - Similar to DOS environment

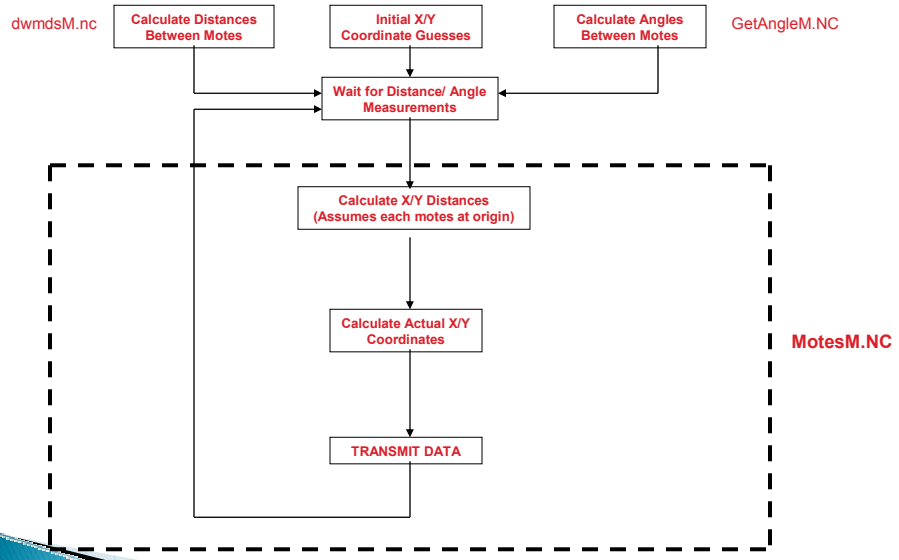
CygWin

```
/opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
avr-objcopy --output-target=ihex build/mica2/main.exe build/mica2/main.ihex

L3 Clinic@L3Clinic /opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
$ make mica2 install.0 MIB510,COM4
mkdir -p build/mica2
    compiling Dwmds3Base to a mica2 binary
ncc -o build/mica2/main.exe -Os -I/opt/tinyos-1.x/contrib/xbow/tos/platform/mica
2 -I/opt/tinyos-1.x/contrib/xbow/tos/apps/dwmds3 -finline-limit=100000 -Wall -Ws
hadow -DDEF_TOS_AM_GROUP=125 -Wnesc-all -target=mica2 -fnesc-cfile=build/mica2/a
pp.c -board=micasb -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_04 -DRADIO_XMIT_POW
ER=0xFF -DCC2420_TXPOWER=TXPOWER_MAX -DCC2420_DEF_CHANNEL=26 -DIDENT_PROGRAM_NAM
E="Dwmds3Bas" -DIDENT_PROGRAM_NAME_BYTES="68,119,109,100,115,51,66,97,115,0" -DI
DENT_USER_HASH=0x462beaefL -DIDENT_UNIX_TIME=0x47e56b47L Dwmds3Base.nc -lm
C:/tinyos/cygwin/opt/tinyos-1.x/tos/system/RealMain.nc: In function `main':
C:/tinyos/cygwin/opt/tinyos-1.x/tos/interfaces/StdControl.nc:63: warning: `resul
t' might be used uninitialized in this function
    compiled Dwmds3Base to build/mica2/main.exe
          18142 bytes in ROM
           1586 bytes in RAM
avr-objcopy --output-target=srec build/mica2/main.exe build/mica2/main.srec
avr-objcopy --output-target=ihex build/mica2/main.exe build/mica2/main.ihex
set-mote-id build/mica2/main.srec build/mica2/main.srec.out-0 0
    installing mica2 binary using mib510
uisp -dprog=mib510 -dserial=COM4 --wr_fuse_h=0xd9 -dpart=ATmega128 --wr_fuse_e=f
f  --erase --upload if=build/mica2/main.srec.out-0
Programmer is not responding.
make: *** [program] Error 2

L3 Clinic@L3Clinic /opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
$ _
```


CREATE A WIRELESS SENSOR NETWORK



- System Flow Chart



Common Problems when working with Mica2 and TinyOS

GetAngleCodeM.NC

```
    angle = low_index*inc;
    // Start a repeating timer that fires every 1000ms
    return call Timer.start(TIMER_REPEAT, 1000);
}
```

```
  event result_t Timer.fired()
  {

    if(angle == 260)

    //if(angle >= 5 && angle <= 6)
    {
        call Leds.redon();
        call Leds.greenon();
        call Leds.yellowon();
    }
    //else if(angle >= 280 && angle <= 299)
    else if(angle == 300)
    {
        call Leds.redon();
        call Leds.greenoff();
        call Leds.yellowoff();
    }
    else if(angle >= 301 && angle <= 320)
    {
        call Leds.redoff();
        call Leds.greenon();
        call Leds.yellowoff();
    }

    return SUCCESS;

  }
}
```


- Memory Issue
  - Create an array for look-up table with estimated angle
  - Use array to calculate and confirm angle
  - Program crash because there is not enough memory
  - Solution: Use for loop to calculate each element of the array

- Not Responding Motes

```
/opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
avr-objcopy --output-target=ihex build/mica2/main.exe build/mica2/main.ihex

L3 Clinic@L3Clinic /opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
$ make mica2 install.0 MIB510,COM4
mkdir -p build/mica2
    compiling Dwmds3Base to a mica2 binary
ncc -o build/mica2/main.exe -Os -I/opt/tinyos-1.x/contrib/xbow/tos/platform/mica
2 -I/opt/tinyos-1.x/contrib/xbow/tos/apps/dwmds3 -finline-limit=1000000 -Wall -Ws
hadow -DDEF_TOS_AM_GROUP=125 -Wnesc-all -target=mica2 -fnesc-cfile=build/mica2/a
pp.c -board=micasb -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_04 -DRADIO_XMIT_POW
ER=0xFF -DCC2420_TXPOWER=TXPOWER_MAX -DCC2420_DEF_CHANNEL=26 -DIDENT_PROGRAM_NAM
E="Dwmds3Bas" -DIDENT_PROGRAM_NAME_BYTES="68,119,109,100,115,51,66,97,115,0" -DI
DENT_USER_HASH=0x462beaefL -DIDENT_UNIX_TIME=0x47e56b47L Dwmds3Base.nc -lm
C:/tinyos/cygwin/opt/tinyos-1.x/tos/system/RealMain.nc: In function `main':
C:/tinyos/cygwin/opt/tinyos-1.x/tos/interfaces/StdControl.nc:63: warning: `resul
t' might be used uninitialized in this function
    compiled Dwmds3Base to build/mica2/main.exe
          18142 bytes in ROM
           1586 bytes in RAM
avr-objcopy --output-target=srec build/mica2/main.exe build/mica2/main.srec
avr-objcopy --output-target=ihex build/mica2/main.exe build/mica2/main.ihex
set-mote-id build/mica2/main.srec build/mica2/main.srec.out-0 0
    installing mica2 binary using mib510
uisp -dprog=mib510 -dserial=COM4 --wr_fuse_h=0xd9 -dpart=ATmega128 --wr_fuse_e=f
f  --erase --upload if=build/mica2/main.srec.out-0
Programmer is not responding.
make: *** [program] Error 2

L3 Clinic@L3Clinic /opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
$ _
```


- Compiled but Where is the data?

```
/opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
avr-objcopy --output-target=ihex build/mica2/main.exe build/mica2/main.ihex
set-mote-id build/mica2/main.srec build/mica2/main.srec.out-0 0
    installing mica2 binary using mib510
uisp -dprog=mib510 -dserial=COM4 --wr_fuse_h=0xd9 -dpart=ATmega128 --wr_fuse_e=f
f  --erase --upload if=build/mica2/main.srec.out-0
Firmware Version: 2.1
Atmel AVR ATmega128 is found.
Uploading: flash

Fuse High Byte set to 0xd9

Fuse Extended Byte set to 0xff
rm -f build/mica2/main.exe.out-0 build/mica2/main.srec.out-0

L3 Clinic@L3Clinic /opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
$ make mica2 install.0 MIB510=/dev/ttyS3
mkdir -p build/mica2
    compiling Dwmds3Base to a mica2 binary
ncc -o build/mica2/main.exe -Os -I/opt/tinyos-1.x/contrib/xbow/tos/platform/mica
2 -I/opt/tinyos-1.x/contrib/xbow/tos/apps/dwmds3 -finline-limit=1000000 -Wall -Ws
hadow -DDEF_TOS_AM_GROUP=125 -Wnesc-all -target=mica2 -fnesc-cfile=build/mica2/a
pp.c -board=micasb -DCC1K_DEFAULT_FREQ=RADIO_916BAND_CHANNEL_04 -DRADIO_XMIT_POW
ER=0xFF -DCC2420_TXPOWER=TXPOWER_MAX -DCC2420_DEF_CHANNEL=26 -DIDENT_PROGRAM_NAM
E="Dwmds3Bas" -DIDENT_PROGRAM_NAME_BYTES="68,119,109,100,115,51,66,97,115,0" -DI
DENT_USER_HASH=0x462beaefL -DIDENT_UNIX_TIME=0x47e56fa5L Dwmds3Base.nc -lm
C:/tinyos/cygwin/opt/tinyos-1.x/tos/system/RealMain.nc: In function `main':
C:/tinyos/cygwin/opt/tinyos-1.x/tos/interfaces/StdControl.nc:63: warning: `resul
t' might be used uninitialized in this function
    compiled Dwmds3Base to build/mica2/main.exe
          18142 bytes in ROM
           1586 bytes in RAM
avr-objcopy --output-target=srec build/mica2/main.exe build/mica2/main.srec
avr-objcopy --output-target=ihex build/mica2/main.exe build/mica2/main.ihex
set-mote-id build/mica2/main.srec build/mica2/main.srec.out-0 0
    installing mica2 binary using mib510
uisp -dprog=mib510 -dserial=/dev/ttyS3 --wr_fuse_h=0xd9 -dpart=ATmega128 --wr_fu
se_e=ff  --erase --upload if=build/mica2/main.srec.out-0
Firmware Version: 2.1
Atmel AVR ATmega128 is found.
Uploading: flash

Fuse High Byte set to 0xd9

Fuse Extended Byte set to 0xff
rm -f build/mica2/main.exe.out-0 build/mica2/main.srec.out-0

L3 Clinic@L3Clinic /opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
$ export MOTECOM=serial@COM4:mica2

L3 Clinic@L3Clinic /opt/tinyos-1.x/contrib/xbow/apps/Dwmds3Base
$ java net.tinyos.tools.Listen | tee listen_test0322.txt
serial@COM4:57600: resynchronising
```

Applications

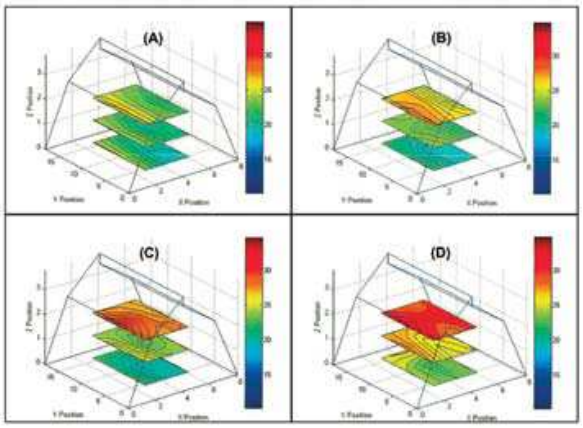
Applications

- Controlled Environment Agriculture Monitor
  - Sense Temperature and Moisture
  - Minimize care and labor
  - Minimize water required (reduced by 50%)
  - Sensor network ties to automation system make CEA systems Economical

(4)

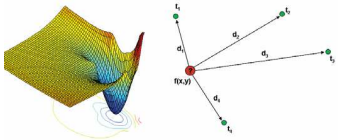
Applications

- Controlled Environment Agriculture Monitor



(4)

Applications



- Military – Snipers Detection
  - PinPtr Application
  - Multi acoustic sensors network measures the muzzle blasts and shock wave
  - Determine the location of the shooters base on speed of sound and sensors location

(3)

Applications

- Military – Snipers Detection

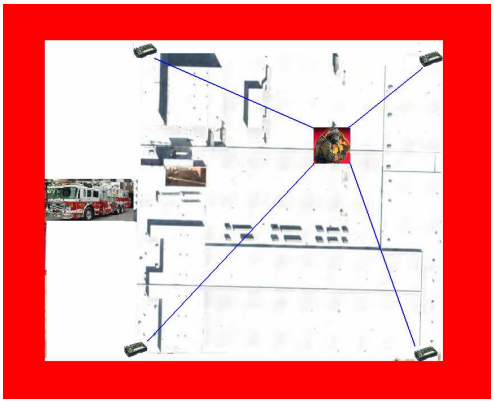
(5)

Applications

- Firefighters
  - Sensors with hear rate monitor
  - Determine the exact location of firefighters even in high rise buildings

Applications

- Firefighters
  - Monitor the condition of firefighters
  - Determine the locations of firefighters not only from a bird view (GPS) but also in 3D environment



Conclusion

- Lessons Learned
  - Experiences from working with new and different technologies
  - Planning and Executing Project effectively
  - Communications in Teamwork

References

1. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf
2. http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Product_Feature_Reference_Chart.pdf
3. http://www.xbow.com/General_info/Info_pdf_files/XbowNewsletter_Q1-05.pdf
4. http://www.xbow.com/General_info/Info_pdf_files/Xbow_Newsletter_Q4_05.pdf
5. http://www.isis.vanderbilt.edu/projects/nest/applications.html

Final Questions