Localization in Sensor Networks

Jie Gao
Computer Science Department
Stony Brook University
Find where the sensor is…

- Location information is important.
  1. Devices need to know where they are.
     - Sensor tasking: turn on the sensor near the window…
  2. We want to know where the data is about.
     - A sensor reading is too hot – where?
  3. It helps infrastructure establishment.
     - geographical routing
     - sensor coverage.
GPS is not always good

- Requires clear sky, doesn’t work indoor.
- Too expensive.
  - A $1 sensor with a $100 GPS?

Localization algorithm:

- (optional) Some nodes (anchors or beacons) know their locations (e.g., through GPS).
- Nodes make local measurements;
  - Distances or angles between two neighbors.
- Communicate between each other;
- Infer location information from these measurements.
Localization problem

- **Output**: nodes’ location.
  - Global location, e.g., what GPS gives.
  - Relative location.

- **Input**:
  - Connectivity, hop count (under Unit Disk Graph model).
    - Nodes with k hops away are within Euclidean distance k.
    - Nodes without a link must be at least distance 1 away.
  - Distance measurement of an incoming link.
  - Angle measurement of an incoming link.
  - Combinations of the above.
Distance Measurements

- Received Signal Strength Indicator (RSSI)
  - The further away, the weaker the received signal.
  - Mainly used for Radio Frequency (RF) signals.
- Time of Arrival (ToA) or Time Difference of Arrival (TDoA)
  - Signal propagation time translates to distance.
  - RF, acoustic, infrared and ultrasound.
Time of Arrival (ToA)

- Used in GPS.
- Triangulation.
- Need synchronization.
- Synchronization can be relaxed if round-trip time is used.
Time Difference of Arrival (TDoA)

- Anchor B1 and B2 send signal to A simultaneously. The time difference of arrival is recorded.
- A stays on the hyperbola:
  \[ \sqrt{(x - x_1)^2 + (y - y_1)^2} - \sqrt{(x - x_2)^2 + (y - y_2)^2} = \delta_d \]
- Do this for B2 and B3.
- A stays at the intersection of the two hyperbolas.
- If the two hyperbolas have 2 intersections, one more measurement is needed.
Angle Measurements

- **Angle of Arrival (AoA)**
  - Determining the direction of propagation of a radio-frequency wave incident on an antenna array.

- **Directional Antenna**

- **Special hardware**, e.g., laser transmitter and receivers.
Angle of Arrival (AoA)

- A measures the direction of an incoming link by radio array.
- By using 2 anchors, A can determine its position.
Localization algorithms for a network

- **Anchor-based**
  - Some nodes know their locations, either by a GPS or as pre-specified.

- **Anchor-free**
  - Relative location only.
  - A harder problem, need to solve the global structure. Nowhere to start.

- **Range-based**
  - Use range information (distance estimation).

- **Range-free**
  - No distance estimation, use connectivity information such as hop count.
Required Papers


Multilateration: use plane geometry
Triangulation, trilateration

- Anchors advertise their coordinates & transmit a reference signal.
- Other nodes use the reference signal to estimate distances to anchor nodes.
Triangulation, trilateration

- Problem: distance measurements are noisy!
- Solve an optimization problem: minimize the mean square error.

![Diagram showing triangulation and trilateration concepts with points a, b, c, and d.]
Indoor localization systems

• RADAR: with RF signals
  – Offline phase: acquire a detailed map of the signal strength from 3 fixed base station inside a building – fingerprinting.
  – Online phase: match the received signal strength with the readings in the offline phase.
  – Significant overhead for fingerprinting

• Cricket: with ultrasound signals
  – Fixed anchor nodes covering the building.
  – Higher granularity.
  – Use trilateration.
Maximal likelihood estimation

- k beacons at positions \((x_i, y_i)\)
- Assume node to be localized has position \((x_0, y_0)\)
- Distance measurement between node 0 and beacon \(i\) is \(r_i\)
- Error:
\[
  f_i = r_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}
\]
Linearization and Min Mean Square Estimate

• Ideally, we would like the error to be 0

\[ f_i = r_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} = 0 \]

• Re-arrange:

\[ (x_0^2 + y_0^2) + x_0(-2x_i) + y_0(-2y_i) - r_i^2 = -x_i^2 - y_i^2 \]

• Subtract the last equation from the previous ones to get rid of quadratic terms.

\[ 2x_0(x_k - x_i) + 2y_0(y_k - y_i) = r_i^2 - r_k^2 - x_i^2 - y_i^2 + x_k^2 + y_k^2 \]

• Note that this is linear.
Linearization and Min Mean Square Estimate

- In general, we have an over-constrained linear system

\[ Ax = b \]

\[
\begin{bmatrix}
    r_1^2 - r_k^2 - x_1^2 - y_1^2 + x_k^2 + y_k^2 \\
    r_2^2 - r_k^2 - x_2^2 - y_2^2 + x_k^2 + y_k^2 \\
    \vdots \\
    r_{k-1}^2 - r_k^2 - x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 
\end{bmatrix}
\]

\[
\begin{bmatrix}
    2(x_k - x_1) & 2(y_k - y_1) \\
    2(x_k - x_2) & 2(y_k - y_2) \\
    \vdots & \vdots \\
    2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) 
\end{bmatrix}
\]

\[
\begin{bmatrix}
    x_0 \\
    y_0 
\end{bmatrix}
\]

\[
\begin{array}{c}
\text{A} \\
\text{x} \\
\text{=} \\
\text{b}
\end{array}
\]
The linearized equations in matrix form become

$$Ax = b$$

Now we can use the least squares equation to compute an estimation.

$$x = (A^T A)^{-1} A^T b$$
How to solve it in a sensor network?

• Check conditions
  – Beacon nodes must not lie on the same line

• For ToA, TDoA, if we use acoustic signals, how to solve for the speed of sound?
Acoustic case: Also solve for the speed of sound

With at least 4 beacons,

\[ f_i = st_i - \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \]

This can be linearized to the form

\[ Ax = b \]

where

\[ b = \begin{bmatrix} -x_1^2 - y_1^2 + x_k^2 + y_k^2 \\ -x_2^2 - y_2^2 + x_k^2 + y_k^2 \\ \vdots \\ -x_{k-1}^2 - y_{k-1}^2 + x_k^2 + y_k^2 \end{bmatrix} \]

\[ A = \begin{bmatrix} 2(x_k - x_1) & 2(y_k - y_1) & t_{k0}^2 - t_{10}^2 \\ 2(x_k - x_2) & 2(y_k - y_2) & t_{k0}^2 - t_{20}^2 \\ \vdots & \vdots & \vdots \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) & t_{k0}^2 - t_{(k-1)0}^2 \end{bmatrix} \]

\[ x = (A^T A)^{-1} A^T b \]
The Node Localization Problem

• Localize nodes in an ad-hoc multihop network
• Based on a set of inter-node distance measurements
Iterative multilateration

- A node with at least 3 neighboring beacons estimates its position and becomes a beacon.
- Iterate until all nodes with 3 beacons are localized.

Connectivity matters! Each node needs at least 3 neighbors.
Iterative multilateration: how many beacons?

- $n$ nodes deployed randomly in a square of side $L$,
- $P(d) = \Pr\{a \text{ node } x \text{ has degree } d\} = \, ?$

Probability that one node falls inside the transmission range of $x$?

Transmit range has radius $R$

$$p = \frac{\pi R^2}{L^2}$$

Binomial distribution

$$P(d) = p^d \cdot (1 - p)^{n-d-1} \cdot \binom{n-1}{d}$$
Iterative multilateration: how many beacons?

- When \( n \) tends to infinity, the binomial distribution converges to a Poisson distribution.

\[
\lambda = n \cdot p
\]

Transmission range has radius \( R \)

Probability that one node falls inside the transmission range of \( x \)?

\[
P(d) = \frac{\lambda^d}{d!} \cdot e^{-\lambda}
\]
Iterative multilateration: how many beacons?

\[ P(d) = \frac{\lambda^d}{d!} \cdot e^{-\lambda} \quad P(\geq d) = 1 - \sum_{i=1}^{n-1} P(i) \]

100 by 100 field
Sensor range: 10

Probability of a node with 0, 1, 2, \( \geq 3 \) neighbors.

With 200 nodes, \( P(\geq 3) \) is about 95%.
Iterative multilateration: how many beacons?

With 200 nodes, \( P(\geq 3) \) is about 95%.

With 200 nodes, we need about 50~60 beacons to localize about 90% of the nodes. That’s \( \frac{1}{4} \) of the total number of nodes.
Problems of iterative multilateration

Problems

1. Requires a large fraction of beacons.
2. Error accumulates.
3. It gets stuck --- not all nodes with 3 or more neighbors can be solved.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{diagram.png}
\caption{Diagram illustrating the problems of iterative multilateration.}
\end{figure}
Problems of iterative Multilateration

Problems

1. Requires a large fraction of beacons.
3. It gets stuck --- not all nodes with 3 or more neighbors can be located. Global optimization (to be discussed next class)
However, optimization does not solve:

**Ambiguity in localization**
Ambiguity in localization

- Same distances, different realization.

(a) Ground truth    (b) Alternate realization

\[ \sigma_{err} = 0.37 \]

Error of the measured distances from the calculated distances

\[ \sigma_{err} = 0.34 \]
Continuous deformation

- Nodes move continuously without violating the distance constraints.
Flip

- No continuous deformation, but the solution is subject to global flipping.
Discontinuous flex ambiguity

- Remove AD, flip ABD up, insert AD.
- No continuous deformation in between.
- But both are valid realization of the distances.

![Diagram showing discontinuous flex ambiguity](image)
Rigidity theory

Given a system of rigid bars and hinges in 2D, does it have a continuous deformation? Or multiple realizations?
Rigidity theory

- Given a set of rigid bars connected by hinges, rigidity theory studies whether you can move them continuously.

![Diagram showing a rigid configuration on the left and a non-rigid configuration on the right.](image)
Rigidity and global rigidity

- Not rigid
- Rigid: No continuous deformation
- Globally rigid: unique realization

What we want!
Intuition on rigidity (not global rigidity yet)

How many distance constraints are necessary to limit a framework to only trivial motion?

==

How many edges are necessary for a graph to be rigid?

Total degrees of freedom: $2n$
How many edges are necessary to make a graph of $n$ nodes rigid?

Each edge can remove a single degree of freedom. Rotations and translations will always be possible, so at least $2n-3$ edges are necessary for a graph to be rigid.
Are 2n-3 edges sufficient?

- $n = 3, 2n-3 = 3$: yes
- $n = 4, 2n-3 = 5$: yes
- $n = 5, 2n-3 = 7$: no
Further intuition

• Need at least $2n-3$ “well-distributed” edges.

• If a subgraph has more edges than necessary, some edges are redundant.

• Non-redundant edges are independent, i.e., they remove a degree of freedom each.

• Therefore, $2n-3$ independent edges guarantee rigidity.
Laman condition

**Laman graph:** it has $2n-3$ edges and no subgraph of $k$ vertices has more than $2k-3$ edges.

**Laman condition:** A graph is rigid if it contains a Laman graph.

What does a Laman graph look like?
Henneberg constructions

- **Henneberg constructions** (Tay-Whiteley): inductive, add one vertex at a time:

- Start with an edge. At each step, add a new vertex
  - Type I step: join the vertex to two old vertices via two edges
  - Type II step: join the vertex to three old vertices with at least one edge in between, via three edges. Remove an old edge between the three endpoints.
Henneberg constructions

- Type I step: join the vertex to two old vertices via two edges
- Type II step: join the vertex to three old vertices with at least one edge in between, via three edges. Remove an old edge between the three endpoints.
Laman = Henneberg construction

• A graph constructed by Henneberg construction is Laman.

• Every Laman graph can be constructed by using Henneberg construction.
**Henneberg ➔ Laman**

**Claim:** A graph constructed “Henneberg-ly” is Laman.

**Proof:** By induction. Suppose the current graph G is Laman with n vertices, 2n-3 edges.

**Type I:** Add node x. We have n+1 vertices, and 
\[2n-3+2=2(n+1)-3\] edges.

Similarly, for a subgraph with k nodes, if it does not include x, by the induction hypothesis, there are \(\leq 2k-3\) edges.

If the subgraph includes x, for the other k-1 nodes, there are at most \(2(k-1)-3\) edges between them (induction hypothesis), in total there are \(\leq 2(k-1)-3 + 2 = 2k-3\) edges.
Type II: Add node $x$. We have $n+1$ vertices, and $2n-3+3-1=2(n+1)-3$ edges.

For a subgraph with $k$ nodes, if it does not include $x$, by the induction hypothesis, there are $\leq 2k-3$ edges.

If the subgraph includes $x$, for the other $k-1$ nodes, there are at most

1. $2(k-1)-3$ edges, if not all of $a$, $b$, $c$ are included.
2. $2(k-1)-4$ edges, if $a, b, c$ are all included.

Add $x$, for case 1, there are $\leq 2(k-1)-3 + 2 = 2k-3$ edges.
For case 2, there are $\leq 2(k-1)-4 + 3 = 2k-3$ edges. #
Claim: Each Laman graph has a Henneberg construction.
• If $m=2n-3$, there exists at least one vertex of degree 2 or 3.
• Otherwise, all nodes have degree 4. Thus we have at least $4n/2=2n$ edges. \(\Rightarrow\) contradiction.
Laman $\Rightarrow$ Henneberg

Claim: Each Laman graph has a Henneberg construction.

- If degree 2: remove the vertex and its adjacent edges (Type I step in reverse)
- If degree 3: remove the vertex and the edges to its three neighbors \( \{a, b, c\} \). They can’t span all three edges (else violate \( 2k-3 \) for \( k=4 \), e.g., \( \{a, b, c, x\} \)). Put one edge between them. (Type II step in reverse).
- Argue like before that Laman still holds, so we can continue.
Questions

• How to identify whether a graph is rigid or not?

• If a graph is globally rigid, how to use this information in localization algorithms?