GPS Free Node Localization in Mobile Wireless Sensor Networks

5th ACM international workshop on Data engineering for wireless and mobile access
Year of Publication: 2006

Presented by

Pavneet Singh Gill

Group No. 1

Background

Purpose of Localization Schemes

- Optimized or maintained connectivity
- Coordination for node tasks

Localization can require GPS

- GPS signal not always available
- Indoors, GPS unreliable

Fixed Point reference nodes

- Use of beacons
- Static network

Paper Contributions

"The main contribution of this work is that it presents a solution to the problem of directional localization in GPS free sensor networks with mobile nodes."

- Provides directional neighbor localization in a network-wide coordinate system
- Works under fairly large motion and distance measurement errors
- Unaffected by the speed of nodes
- Works for any network size
- Supports a stable network in mobility problems

Localization Algorithm

Assumptions

- Each node has a compass pointing North (or any other common reference direction)
- Nodes can measure the distance to their neighbors using a well known range measurement method (e.g. Time of Arrival (TOA))
- Motion actuators allow each node to move a specific distance in a specific direction.
- Actuator, compass and distance measurements are subject to errors caused by various real world disturbances
- Other than the above, no additional positioning equipment or infrastructure is required.

Localization Algorithm

The core localization algorithm works on well defined rounds. Each round essentially consists of three steps:

1. Measure distances between neighbors
2. Move nodes
3. Exchange distance values for that round as validation for movement

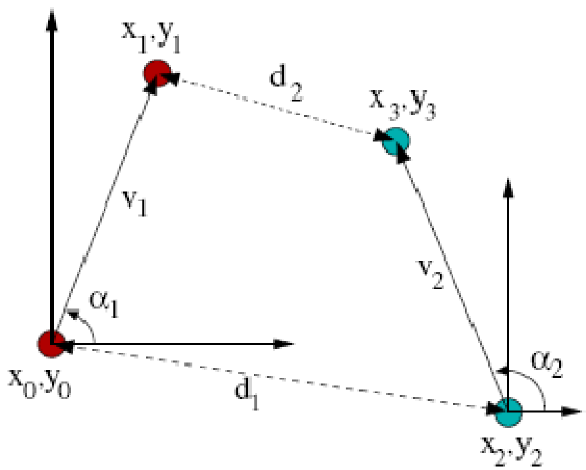
Localization Algorithm

Core Localization Algorithm

CoreLocalization(n_1, n_2, v_1, α_1)

1: $d_1 \leftarrow$ inter-distance(n_1, n_2)
2: Move node n_1 by v_1 and α_1
3: $d_2 \leftarrow$ inter-distance(n_1, n_2)
4: Retrieve v_2 and α_2 from n_2
5: Calculate positions of n_2 using equations (4),(5) and (6)

Localization Algorithm



$$x_1 = v_1 \cos\alpha_1, \qquad y_1 = v_1 \sin\alpha_1,$$

$$x_3 = x_2 + v_2 \cos\alpha_2, \qquad y_3 = y_2 + v_2 \sin\alpha_2,$$

$$(x_3 - x_1)^2 + (y_3 - y_1)^2 = d_2^2, \qquad x_2^2 + y_2^2 = d_1^2.$$

$$x_2 A + y_2 B = C, \tag{4}$$

$$x_2^2 D - 2x_2 E + F = 0, \qquad y_2^2 D - 2y_2 G + H = 0, \tag{5}$$

$$x_2 = \frac{E \pm \sqrt{E^2 - DF}}{D}, \qquad y_2 = \frac{G \pm \sqrt{G^2 - DH}}{D} \tag{6}$$

$$A = v_2 \cos\alpha_2 - v_1 \cos\alpha_1, \qquad B = v_2 \sin\alpha_2 - v_1 \sin\alpha_1,$$

$$C = \frac{1}{2}\left(d_2^2 - d_1^2 - v_1^2 - v_2^2 + 2v_1 v_2 \cos(\alpha_1 - \alpha_2)\right).$$

$$D = A^2 + B^2, \qquad E = AC, \qquad F = C^2 - d_1^2 B^2,$$

$$G = BC, \qquad H = C^2 - d_1^2 A^2.$$

Localization Algorithm

Verification Algorithm

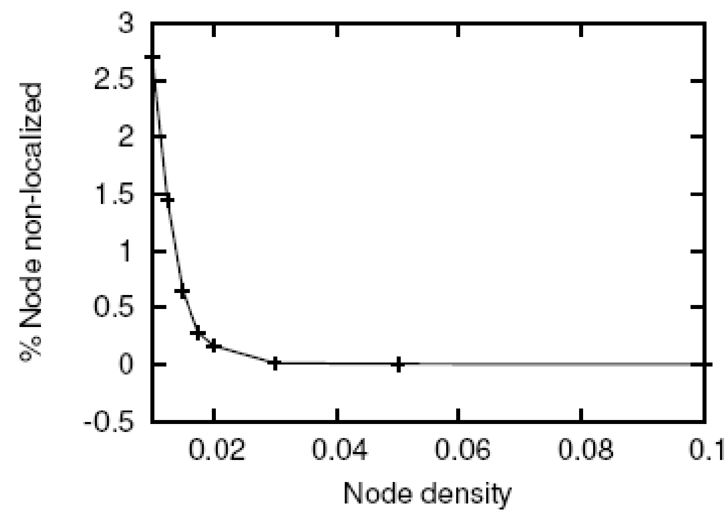
VERIFICATION(NEIGHBORLIST NL)

for each neighbor pair (m, n) in NL **do**
  if m and n are neighbors **then**
    $d_{m,n} \leftarrow$ measured inter-distance(m, n)
    for each position pair $\{m^i, n^j \mid i, j = 1, 2\}$ **do**
      Compute Euclidean distance D between m^i and n^j
      if $D = d_{m,n}$ **then**
        mark m^i and n^j as exact positions

Simulation Results

Ideal Conditions

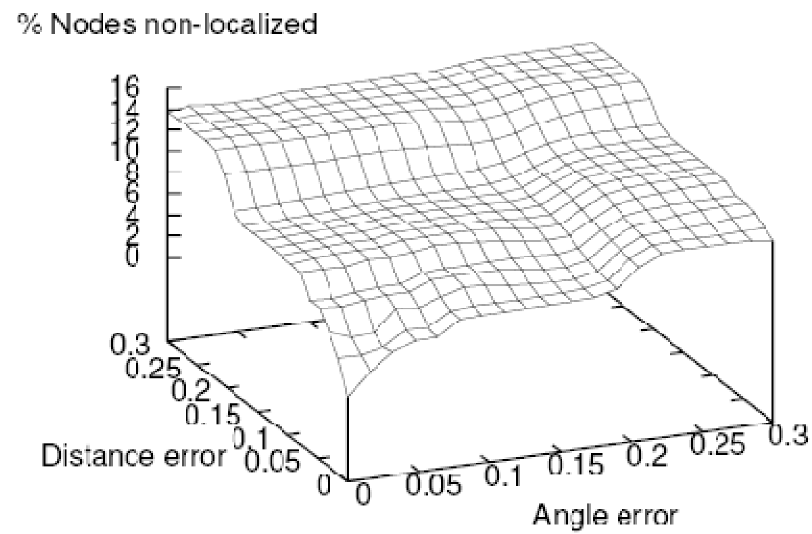
- Placed nodes in a 100x100 area
- Each simulation is run for 100 rounds
- Random walk, random speed
- As small as 3% maximum failure for very sparse networks



Simulation Results

Including Measurement Errors

- Same as previous experiment
- Distance noise as a percent of measured value
- Angle noise as a static % of 2 pi



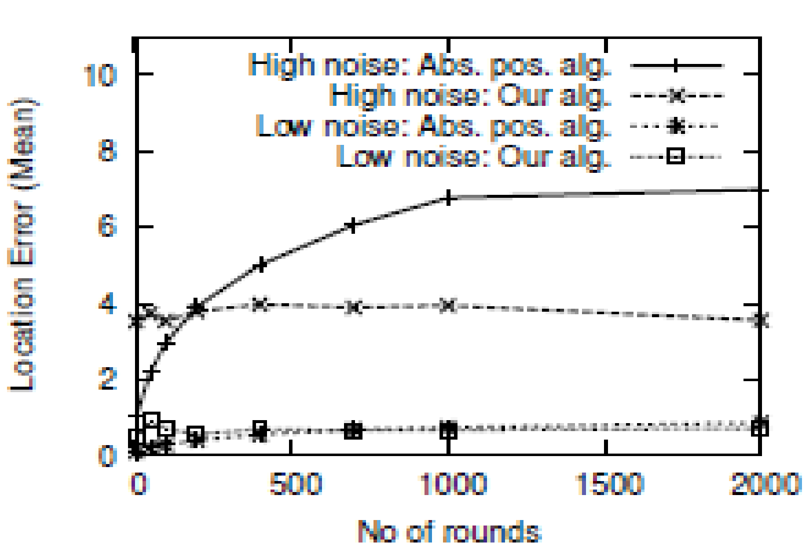
Simulation Result

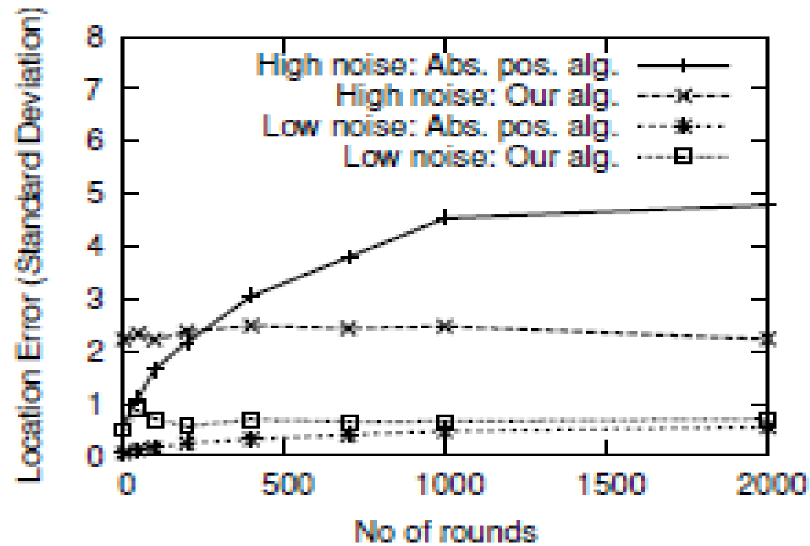
Comparison with an absolute positioning algorithm

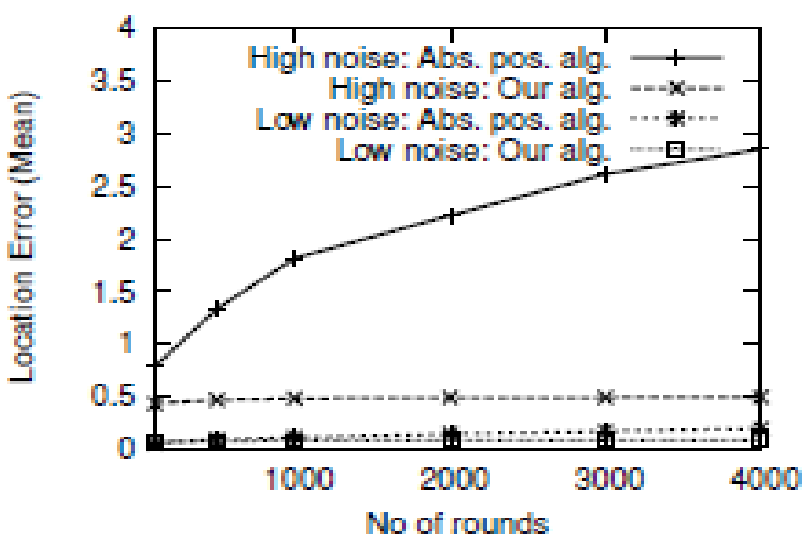
- Units are placed with initially perfect data and record their own movement from there
- Performed random movement and directed movement experiments
- 2000 rounds, averaged error at 6 places

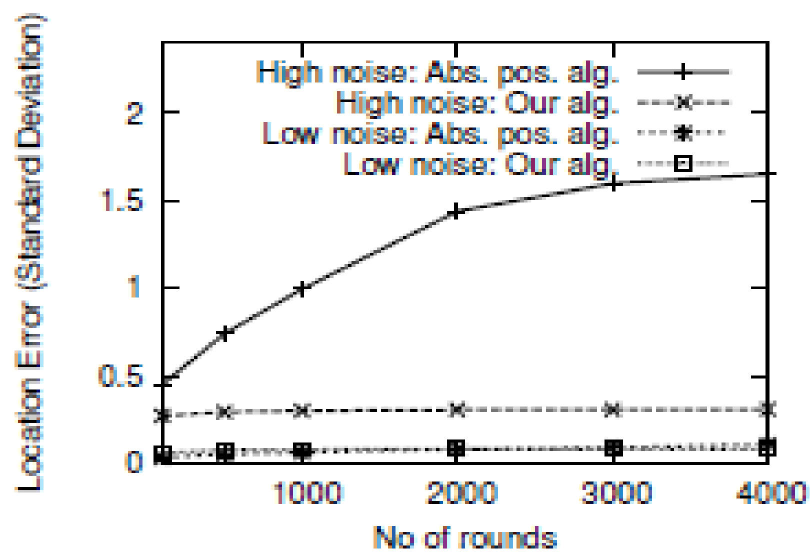
		Our alg.		Abs. pos. alg.	
Movement	Noise	Mean	Stdev	Mean	Stdev
Random	low	0.71	0.69	0.86	0.55
	high	3.55	2.23	6.98	4.78
Directed	low	0.08	0.08	0.19	0.11
	high	0.49	0.30	2.85	1.65

Simulation Results









Strength And Weakness

- Strengths:
  - Localization without the use of GPS.
  - Does not require any infrastructure
  - Information about relative position.
  - Gives error in the range of 3m.
- Weakness:
  - Need of compass at each node.
  - Cannot give the location information in terms of coordinates.
  - Comparison of algorithm with an absolute positioning algorithm whose name has not been mentioned.

Paper Significance to Project

- Useful for relative localization of nodes.
- Can be used at instance as no infrastructure is required.
- GPS for our project?

Algorithms for Nodes Localization in Wireless Ad-Hoc Networks Based on Cost Function

International Workshop on Wireless Ad- Hoc Networks(IWWAN)

Year of Publication : 2005

Introduction

- Distribute algorithm to localize sensor network nodes.
- Use of UWB for accuracy .
- Based on capacity of nodes to calculate Time of arrival(TOA) and Angle of Arrival(AOA).

Algorithm

1. Determine Node beacon distances

2. Compute the cost function of every node.

3. Compute the coordinate of the node using Davidson-Fletcher-Powell Quasi Newton algorithm (DFP) or Direct Method(DM) algorithm.

Relative Position Estimation

- Shortest Path Estimation:

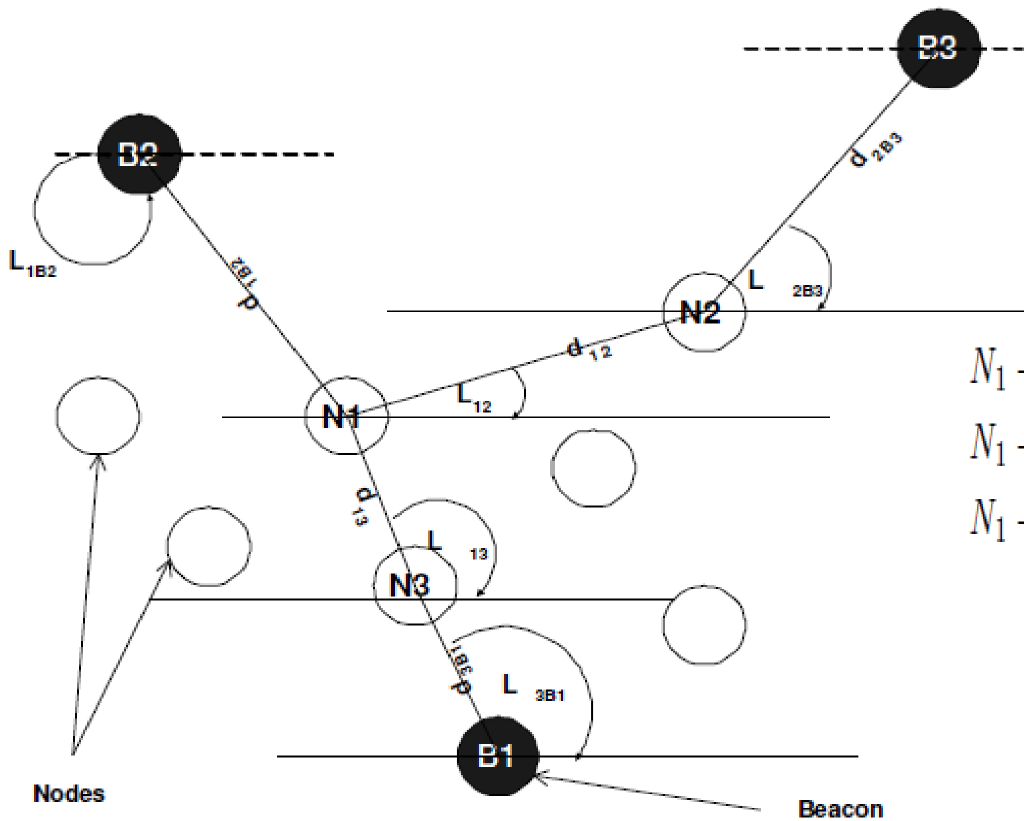


Fig. 1. Scheme of Wireless Sensor Network

$$
\begin{aligned}
N_1 \rightarrow B_1 &: \quad \Phi_1 = \{(N_1, N_3, B_1), (\hat{\alpha}_{13}, \hat{\alpha}_{3B_1}), (\hat{d}_{13}, \hat{d}_{3B_1})\} \\
N_1 \rightarrow B_2 &: \quad \Phi_2 = \{(N_1, B_2), (\hat{\alpha}_{1B_2}), (\hat{d}_{1B_2})\} \\
N_1 \rightarrow B_3 &: \quad \Phi_3 = \{(N_1, N_2, B_3), (\hat{\alpha}_{12}, \hat{\alpha}_{2B_3}), \\
&\quad\quad (\hat{d}_{12}, \hat{d}_{2B_3})\} \qquad (1)
\end{aligned}
$$

Relative Position Estimation

- Position Estimation:

  Cost Function α estimated angle,

  distance between nodes,

  no. of beacons,

  no. of hops between a node and the given beacon

$$E(x,y) =$$

$$\sum_{k=1}^{N_B}\left(\sqrt{\left(x_k-\left(x_k+\sum_{i=1}^{N_{hop}(k)}\hat{d}_i\cdot\cos(\hat{\alpha}_i)\right)\right)^2+\left(y_k-\left(y_k+\sum_{i=1}^{N_{hop}(k)}\hat{d}_i\cdot\sin(\hat{\alpha}_i)\right)\right)^2}-\sqrt{\left(x_k-x\right)^2+\left(y_k-y\right)^2}\right)^2$$

Positioning Algorithm

- Davidson-Fletcher-Powell Quasi Newton algorithm (DFP):
  - Gives position vector using the following equation:

$$f(p) = \sum_{i=1}^{N_B} \left[\sqrt{(x_e - x_i)^2 + (y_e - y_i)^2} - \hat{d}_{ei}\right]^2$$

$$\mathbf{p} = [x, y]^{\mathrm{T}}.$$

Positioning Algorithm

- Direct Method:

$$f(p) = \sum_{i=1}^{N_B} [\sqrt{(x_e - x_i)^2 + (y_e - y_i)^2} - \hat{d}_{ei}]^2$$

$$\begin{pmatrix} x_e \\ y_e \end{pmatrix} = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$$

Simulation Results

- Results show that algorithm is more sensitive to TOA than to AOA.

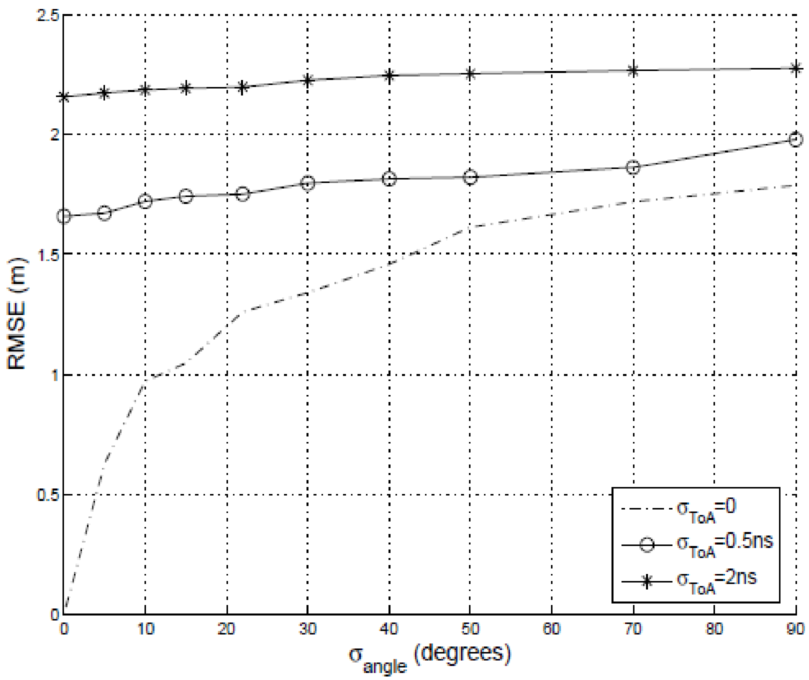


Fig. 2. Root Mean Squared error of position estimation with Direct Method. Only 5 beacons are chosen to position all the sensor nodes in the WSN. Three different values of the variance on the ToA error are chosen ($\sigma_{ToA} = \{0, 0.6, 2\}$ns).

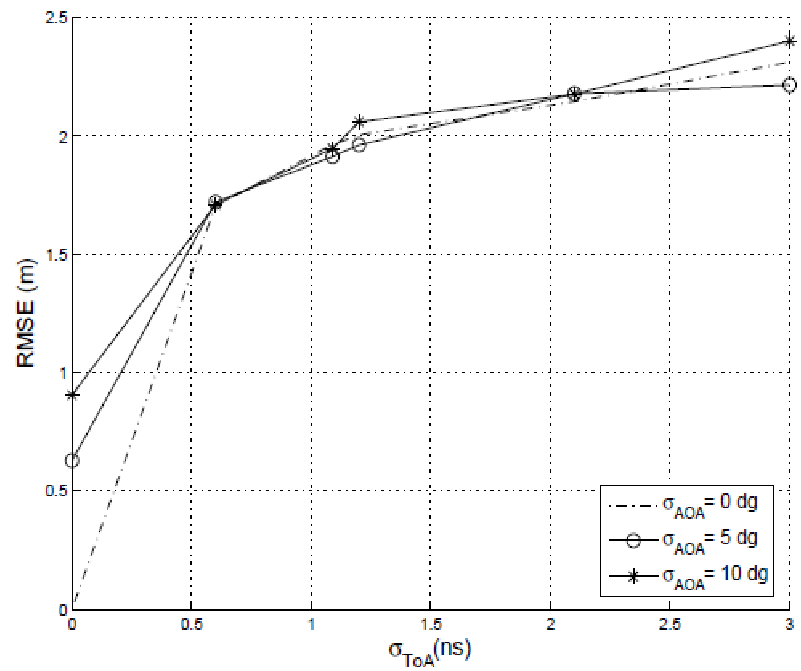


Fig. 4. Root Mean Squared error of position estimation with Direct Method. Only 5 beacons are chosen to position all the sensor nodes in the WSN. Three different values of the variance on the AoA error are chosen ($\sigma_{AoA} = \{0, 5, 10\}$degrees).

Simulation Results

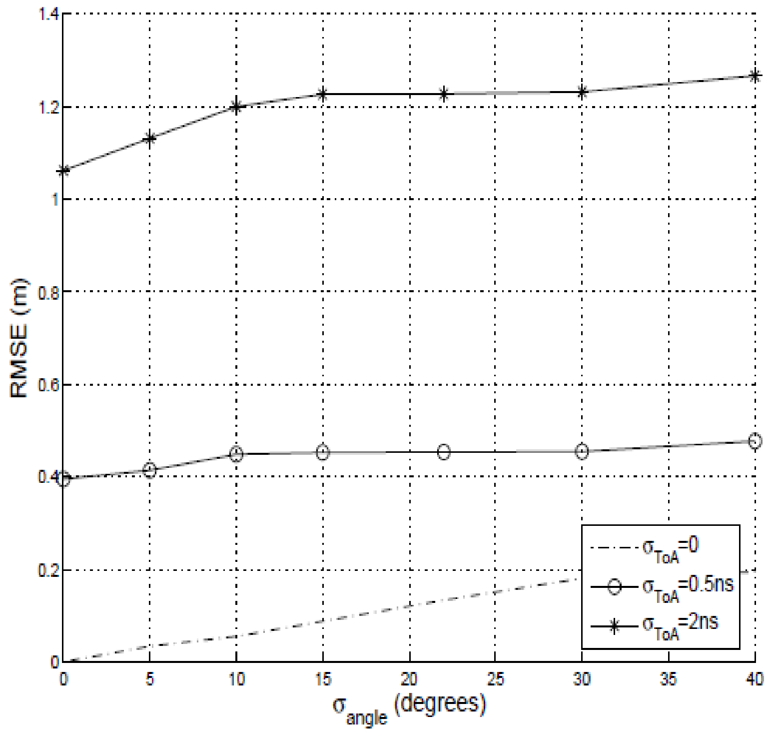


Fig. 3. Root Mean Squared error of position estimation with DFP algorithm. Only 5 beacons are chosen to position all the sensor nodes in the WSN. Three different values of the variance on the ToA error are chosen ($\sigma_{ToA} = \{0, 0.6, 2\}$ns).

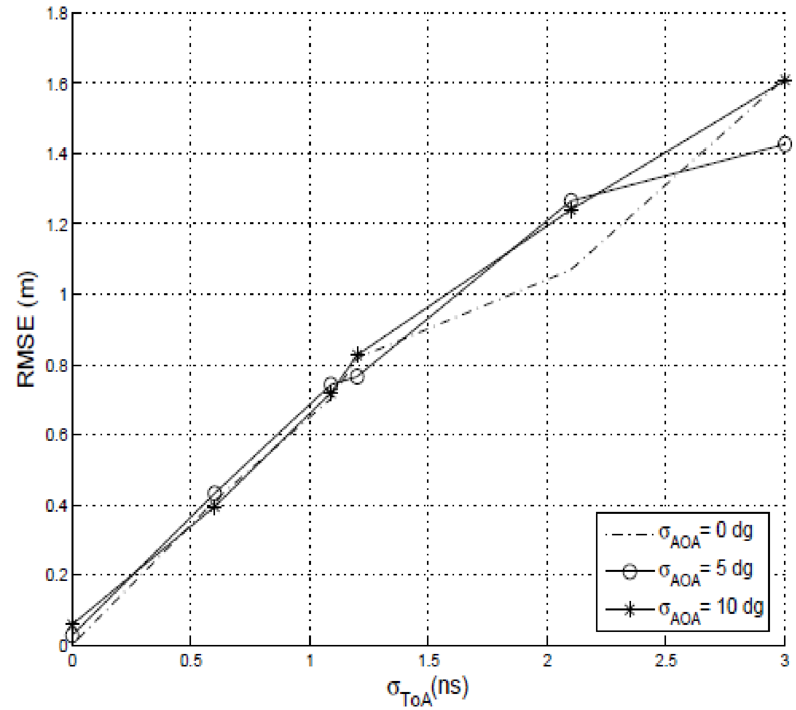


Fig. 5. Root Mean Squared error of position estimation with Direct Method. Only 5 beacons are chosen to position all the sensor nodes in the WSN. Three different values of the variance on the AoA error are chosen ($\sigma_{AoA} = \{0, 5, 10\}$ degrees).

Simulation Results

- DFP is positionally more accurate than DM

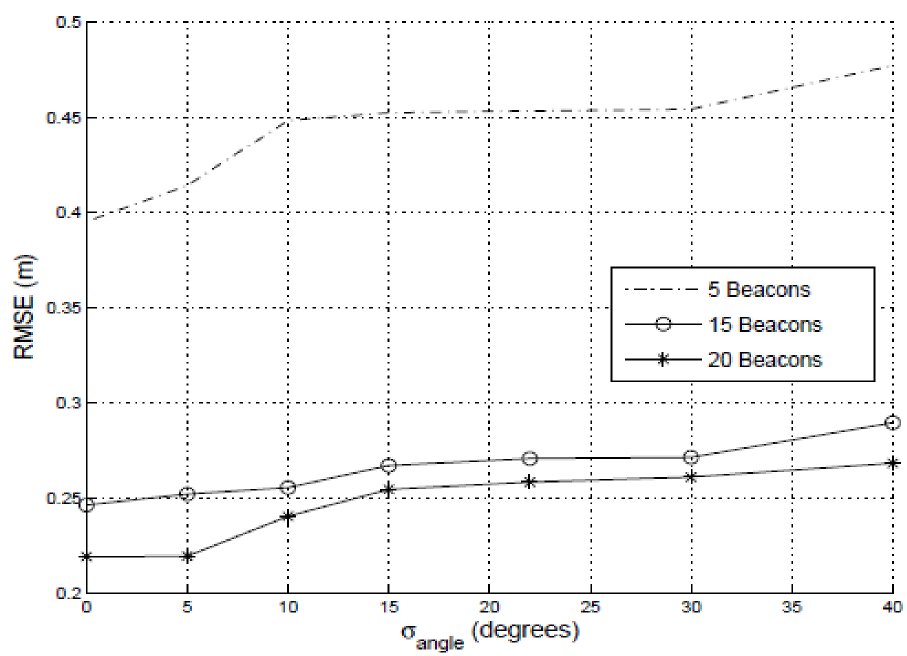


Fig. 6. Root Mean Squared error of position estimation with DFP using 5, 15 and 20 beacons. The σ_{ToA} is equal to $0,5$ ns.

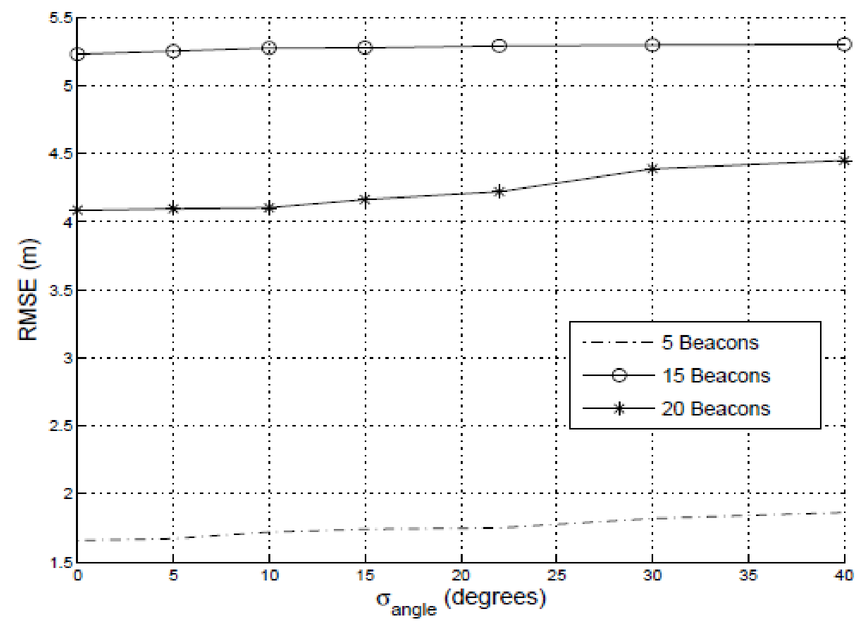


Fig. 7. Root Mean Squared error of position estimation with DM using 5, 15 and 20 beacons. The σ_{ToA} is equal to $0,5$ ns.

Strengths and Weakness

- Strengths:
  - Gives exact position along with coordinates of node with error of about 1.3 meters.
  - Distributed algorithm.
  - No use of GPS and use of UWB.
  - No extra devices or instruments needed on the node.

Strengths and Weakness

- Weakness:
  - Algorithm not compared with other existing algorithms.
  - Need of beacons along with their coordinates to estimate the position of nodes.
  - Depends upon the density of beacons for accuracy.

Paper Significance to the Project

- Doesn’t rely on GPS

- Useful in determining the location with exact coordinates.