NetSim®

Software: NetSim Standard v15.0

Localization in WSN using Convex Position Estimation

Project Information

Configuration Files

`https://github.com/NetSim-TETCOS/Localization_in_WSN_using_Convex_Position_Estimation-v15.0/archive/refs/heads/main.zip`

The NetSim configuration files used for the localization scenarios are available at the above link.

Contents

1 Introduction

Wireless sensor localization estimates the position of an unknown sensor from anchor sensors whose positions are known. In range-based localization, the estimated position is obtained from anchors that satisfy the communication range constraint with the unknown sensor.

In this experiment, NetSim packet trace is used to identify one-hop anchors. The coordinates of these anchors are then used by the convex position estimation method to calculate the unknown sensor position.

2 Convex Position Estimation

Let $p_i = (x_i, y_i)$ be the known position of anchor i, and let $p = (x, y)$ be the unknown node position. If the unknown node hears anchor i, then the unknown node is assumed to lie within a circle of radius r_c around that anchor:

$$\|p_i - p\| \le r_c \tag{1}$$

The range constraint follows from how reception works under a range-based radio. The unknown node can decode a packet from anchor i only when anchor i is within communication range. Hearing anchor i is therefore equivalent to the geometric statement that the unknown node lies within distance r_c of that anchor, that is $\|p_i - p\| \le r_c$. Each heard anchor contributes one such disc, and the unknown node must lie in the intersection of all of them.

NetSim range-based pathloss fits this assumption. The model applies a hard distance cutoff: a packet is received if and only if the link distance is within the configured wireless range, with no reception beyond it. The boundary is sharp rather than tapered by fading, so the region from which an anchor can be heard is a disc of radius equal to the wireless range. Setting r_c to the 41 m wireless range makes the modeling disc and the radio coverage identical. A log-distance or fading model would give a probabilistic boundary, and the disc would only be an approximation.

For one anchor, the localization condition can be represented through the following symmetric matrix:

$$F_i = \begin{bmatrix} r_c & 0 & x_i - x \\ 0 & r_c & y_i - y \\ x_i - x & y_i - y & r_c \end{bmatrix} \tag{2}$$

The Schur complement of the leading 2 × 2 block gives

$$r_c - \frac{(x_i - x)^2 + (y_i - y)^2}{r_c} > 0 \tag{3}$$

which is equivalent to

$$(x_i - x)^2 + (y_i - y)^2 < r_c^2 \tag{4}$$

When the unknown node hears M anchors, the matrices F_i are arranged in block-diagonal form:

$$F = \operatorname{diag}(F_1, F_2, \ldots, F_M) \tag{5}$$

The point (x, y) is accepted when all blocks satisfy the positive definiteness test. The feasible set is therefore the intersection of circular discs around all one-hop anchors. The estimated position is the centroid of all accepted grid points.

3 NetSim Scenario Configuration

The scenarios are configured in NetSim Standard v15.0 using WSN nodes with IEEE 802.15.4 interfaces. The same radio, routing, traffic, packet trace, and simulation time settings are used across all sensor counts, so the comparison is based on anchor density and placement pattern.

3.1 Part A: Uniform Sensor Placement

For uniform sensor placement, the anchor sensors are placed over the 100 m × 100 m grid using square-grid layouts. The unknown sensor is added at $(35, 45)$ and configured as the destination for all CBR applications.

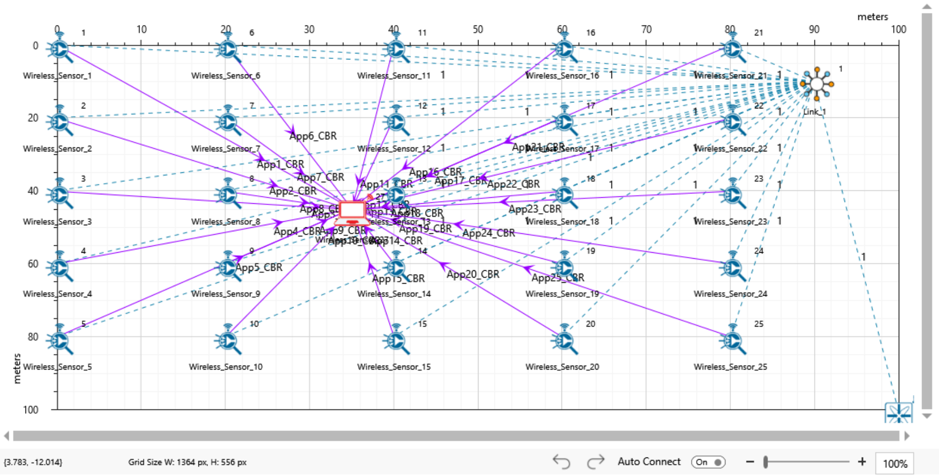

Figure 3-1: *25-sensor uniform placement scenario in NetSim*

Table 3-1: *Common scenario properties*

Property	Value
Network	WSN
Simulation area	100 m × 100 m grid
Anchor placement	Uniform square grid
Unknown sensor position	$(35, 45)$ m
Mobility model	No mobility
Wireless interface	Zigbee / IEEE 802.15.4
PHY data rate	250 kbps
Frequency band	2400 MHz
Modulation	O-QPSK
Channel characteristics	Pathloss
Pathloss model	Range based
Wireless range	41 m
Network layer	IPv4 with AODV routing
Transport/Application	UDP unicast CBR
Traffic generation rate	400 bps per application
Packet size	50 bytes
Inter-arrival time	1000000 μs
Application start time	0 ms
Application end time	100000 ms
Simulation time	100 s
Packet trace	Enabled

Table 3-2: *Uniform placement cases*

Anchor sensors	Grid layout	Density (sensors/m^2)
25	5×5	0.0025
36	6×6	0.0036
49	7×7	0.0049
64	8×8	0.0064
81	9×9	0.0081
100	10×10	0.0100

3.2 Part B: Random Sensor Placement

For random sensor placement, the anchor sensors are distributed randomly over the 100 m × 100 m grid. The unknown sensor is configured with the same destination settings used in Part A. The radio, routing, traffic, packet trace, and simulation time settings are kept the same as the uniform placement cases.

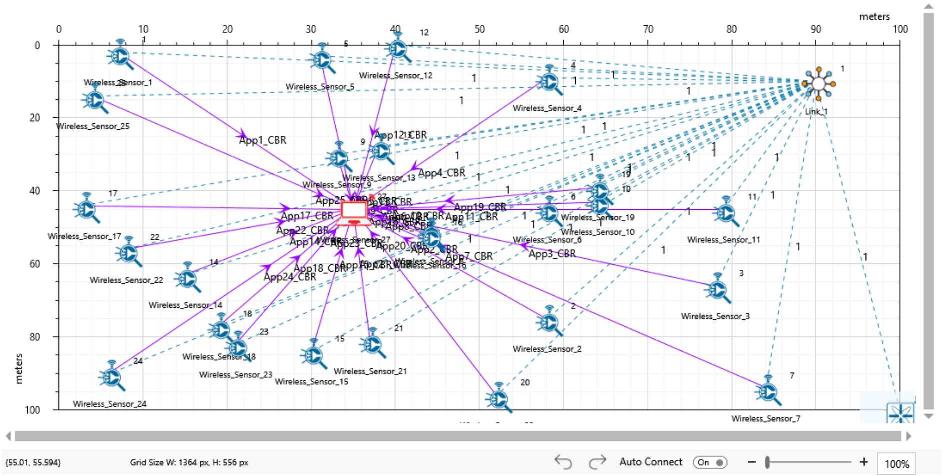

Figure 3-2: *25-sensor random placement scenario in NetSim*

Table 3-3: *Random placement cases*

Anchor sensors	Density (sensors/m^2)
25	0.0025
36	0.0036
49	0.0049
64	0.0064
81	0.0081
100	0.0100

4 Packet Trace Analysis

For each scenario, packet trace is enabled in NetSim. The packet trace records the source, destination, transmitter, receiver, protocol event time, packet type, and packet status for every logged packet. This trace is used to identify the anchors that are one hop away from the unknown sensor.

In a multi-hop AODV path, the source sensor may reach the unknown sensor through intermediate sensors. Therefore, the `SOURCE_ID` field alone is not sufficient for this analysis. The one-hop anchors are obtained from the transmitter and receiver columns in `Packet Trace.csv`. A sensor is counted as a one-hop anchor when it appears as the transmitter in a row where the unknown sensor appears as the receiver and the packet status indicates packet reception.

The packet trace analysis uses the following fields:

- `TRANSMITTER_ID`: sensor that transmitted the packet over the wireless link.
- `RECEIVER_ID`: sensor that received the packet over the wireless link.
- `PACKET_STATUS`: packet reception status.

After the one-hop anchor IDs are identified, their coordinates are taken from the node positions in `Configuration.netsim`. These anchor coordinates are used as the input for the localization calculation.

5 Localization Algorithm

The localization algorithm is implemented in `localization.py`. It performs the convex position estimation calculation using the one-hop anchor coordinates obtained from packet trace analysis. The algorithm applies the 41 m communication range constraint and searches the candidate position grid. A candidate point is retained only when it lies inside the range disc of every one-hop anchor. The centroid of the retained points is used as the estimated position.

The localization error is calculated as:

$$E = \frac{\sqrt{(x_a - \bar{x})^2 + (y_a - \bar{y})^2}}{41} \times 100$$

where (x_a, y_a) is the actual unknown sensor position and $(\bar{x}, \bar{y})$ is the estimated position. For the initial localization cases, $(x_a, y_a) = (35, 45)$.

6 Results

Table 6-1: *Uniform placement localization results from 100 s NetSim runs*

Sensors	Density	One-hop anchors	X estimate	Y estimate	Error (%)
25	0.0025	13	33.65	46.35	4.65
36	0.0036	18	34.47	47.81	6.96
49	0.0049	26	35.76	42.97	5.29
64	0.0064	35	35.33	45.00	0.81
81	0.0081	42	35.50	45.83	2.37
100	0.0100	52	35.00	45.00	0.00

Table 5-1 gives the estimated position for each uniform placement case. The number of one-hop anchors is obtained from packet trace analysis using the transmitter-receiver relationship described in Section 4. The one-hop anchor count increases as the uniform grid becomes denser.

The 25-sensor case uses 13 one-hop anchors and gives an error of 4.65%. The 36-sensor and 49-sensor cases use more one-hop anchors, but their estimates shift because the one-hop anchors are not distributed symmetrically around the unknown sensor. The 64-sensor case reduces the error to 0.81%. The 100-sensor case gives the lowest error for uniform placement, estimating the unknown sensor position as $(35.00, 45.00)$.

Table 6-2: *Random placement localization results from 100 s NetSim runs*

Sensors	Density	One-hop anchors	X estimate	Y estimate	Error (%)
25	0.0025	15	34.59	52.42	18.12
36	0.0036	17	41.85	38.06	23.78
49	0.0049	24	39.38	44.73	10.69
64	0.0064	31	35.60	44.80	1.54
81	0.0081	44	35.60	45.80	2.44
100	0.0100	48	35.50	44.50	1.72

For the random placement cases, packet trace analysis gives 15 to 48 one-hop anchors as density increases from 0.0025 to 0.0100 sensors/m^2. The 36-sensor random case gives the highest error because its one-hop anchors produce an uneven feasible region. The 64-sensor, 81-sensor, and 100-sensor random cases give lower errors as the anchors constrain the unknown sensor position from more directions.

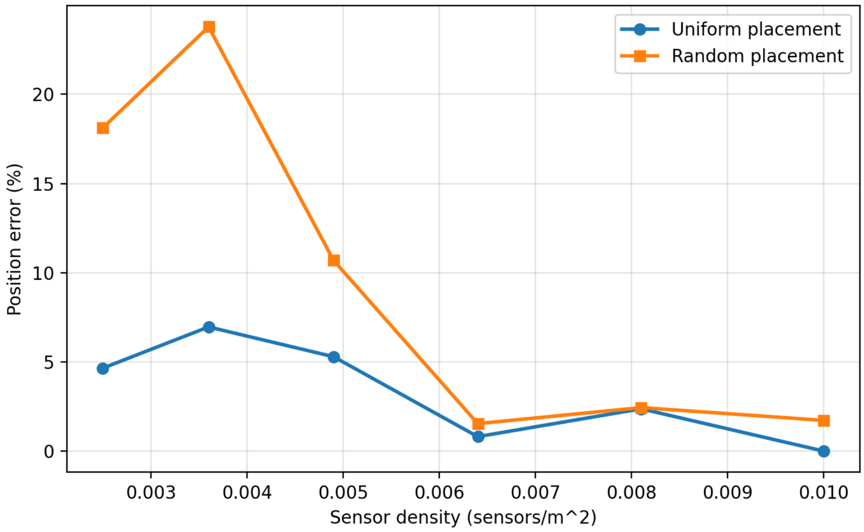

Figure 6-1: *Position error versus sensor density*

Observation

Increasing sensor density increases the number of anchors that can reach the unknown sensor in one hop. However, the estimation error depends on the geometric distribution of those anchors around the unknown sensor. A higher one-hop count improves the feasible region only when those anchors constrain the position from multiple directions.

7 Localization Plots

Black dots denote one-hop anchors, red stars denote accepted candidate points, the blue point denotes the actual unknown sensor position, and the green point denotes the estimated position.

7.1 Part A: Uniform Sensor Placement

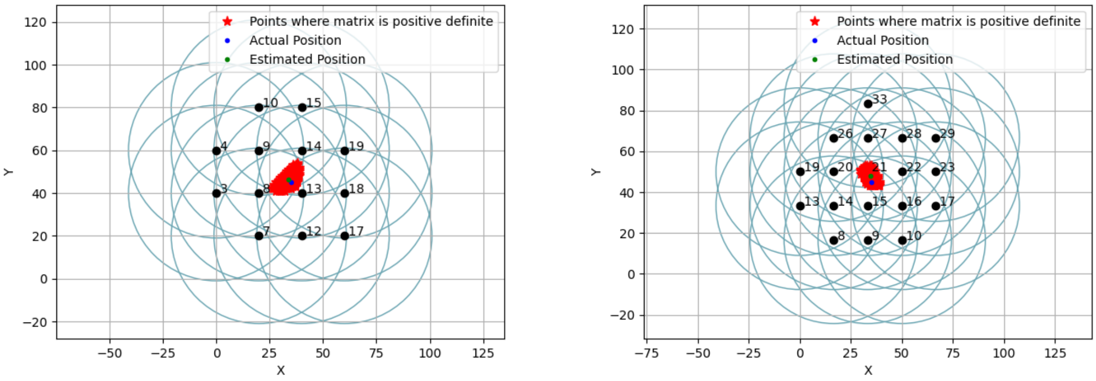

Figure 7-1: *Localization plots for 25 and 36 sensors*

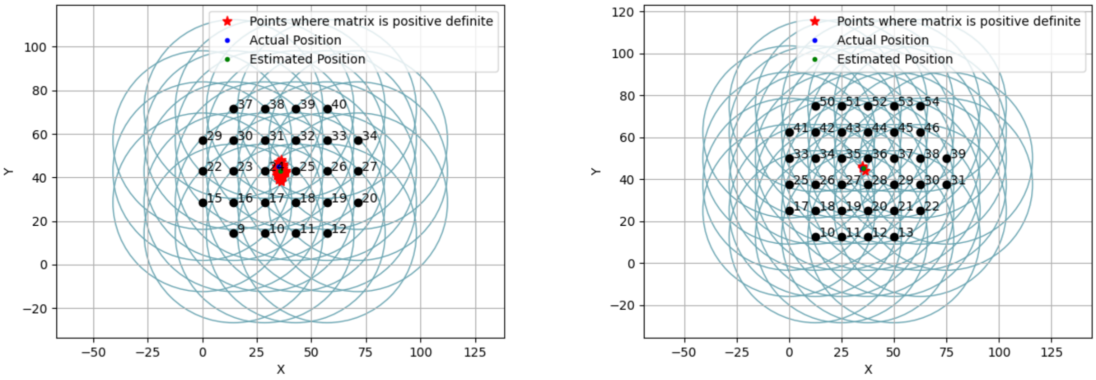

Figure 7-2: *Localization plots for 49 and 64 sensors*

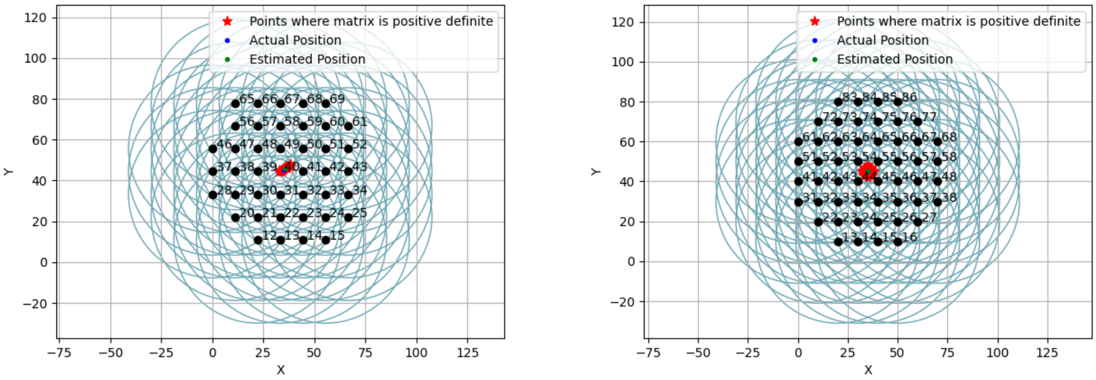

Figure 7-3: *Localization plots for 81 and 100 sensors*

7.2 Part B: Random Sensor Placement

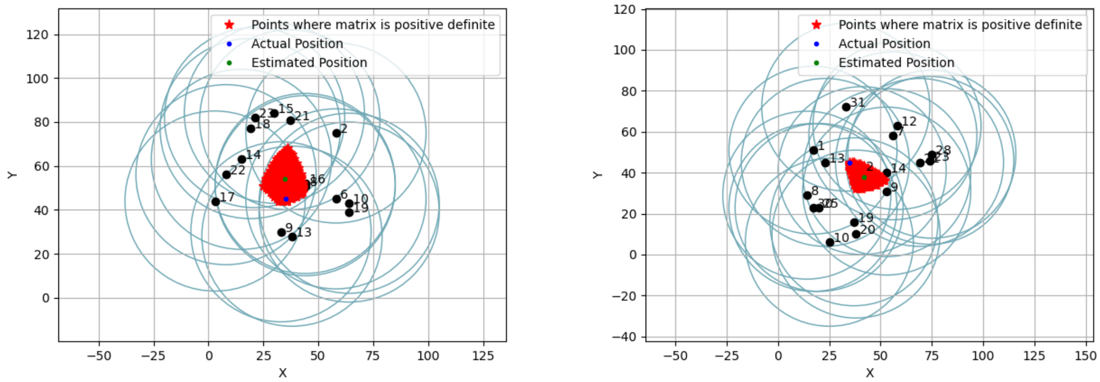

Figure 7-4: *Localization plots for 25 and 36 sensors with random placement*

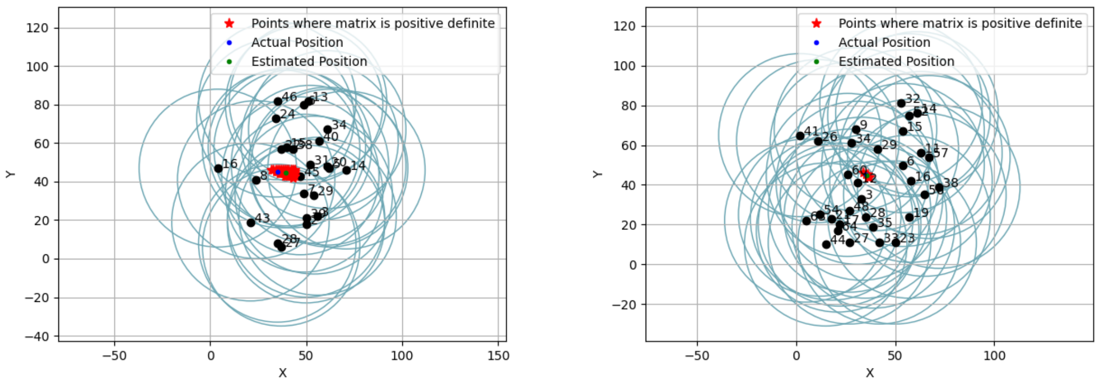

Figure 7-5: *Localization plots for 49 and 64 sensors with random placement*

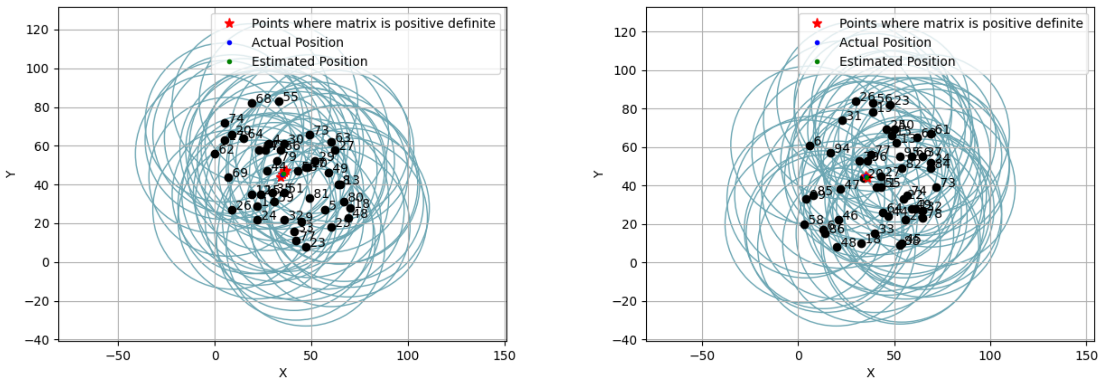

Figure 7-6: *Localization plots for 81 and 100 sensors with random placement*

7.3 Monte Carlo Simulation

Monte Carlo simulation is carried out for uniform sensor placement using ten unknown sensor positions: $(35, 45)$, $(22, 68)$, $(48, 31)$, $(73, 54)$, $(16, 27)$, $(61, 82)$, $(87, 19)$, $(39, 76)$, $(52, 12)$, and

$(28, 43)$.

For each unknown sensor position, packet trace analysis is used to identify one-hop anchors. The localization search grid is centered around the actual unknown sensor position and limited to the 100 m × 100 m simulation area. The search bounds are:

$$X_{\min} = \max(0, x_a - 25), \quad X_{\max} = \min(100, x_a + 25)$$
$$Y_{\min} = \max(0, y_a - 25), \quad Y_{\max} = \min(100, y_a + 25)$$

The bounded search rule keeps the candidate grid inside the NetSim area while allowing the search region to move with the unknown sensor position.

Table 7-1: *Monte Carlo average error for uniform placement*

Sensors	Density	Average error (%)
25	0.0025	15.45
36	0.0036	14.45
49	0.0049	12.77
64	0.0064	11.29
81	0.0081	10.24
100	0.0100	10.66

The average error decreases from 15.45% for 25 sensors to 10.24% for 81 sensors. The 100-sensor case gives 10.66%, which is close to the 81-sensor result. Increased density improves the average localization result, while the unknown sensor position and one-hop anchor geometry still affect the final estimate.

Table 7-2: *Monte Carlo average error for random placement*

Sensors	Density	Average error (%)
25	0.0025	19.09
36	0.0036	17.60
49	0.0049	21.13
64	0.0064	12.73
81	0.0081	10.90
100	0.0100	11.38

For random sensor placement, the average error is 19.09% for 25 sensors, 17.60% for 36 sensors, 21.13% for 49 sensors, 12.73% for 64 sensors, 10.90% for 81 sensors, and 11.38% for 100 sensors. The one-hop anchor geometry varies with the random placement pattern, and this affects the feasible region used for position estimation.

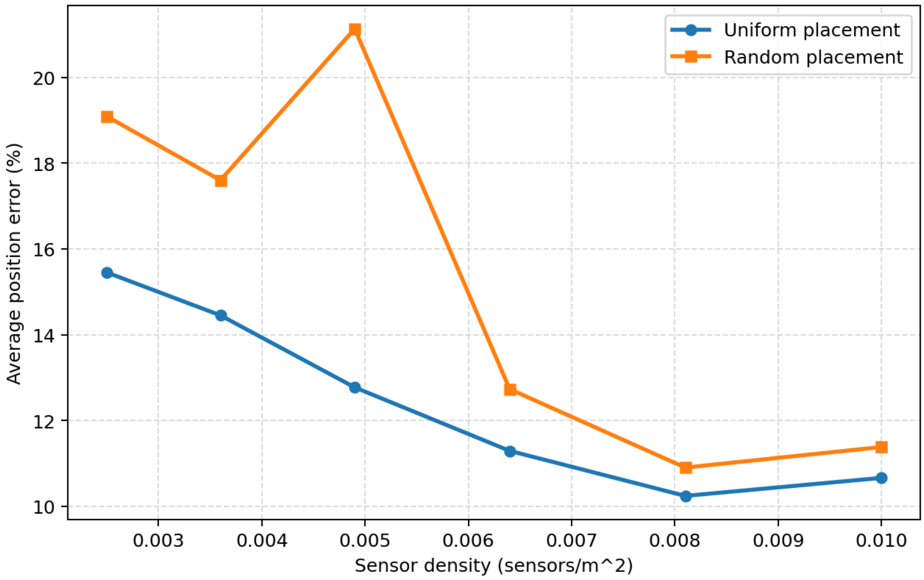

Figure 7-7: *Monte Carlo average position error versus sensor density*

Tables 6-1 and 6-2 report single runs for one unknown-sensor position, $(35, 45)$. Tables 7-1 and 7-2 report Monte Carlo averages over ten unknown-sensor positions.

Position $(35, 45)$ sits near the centre of the grid. The convex method estimates the position as the centroid of the feasible region, so when the one-hop anchors surround the unknown node symmetrically, the centroid coincides with the true point. This is why the single-run error falls to 0.00% for the 100-sensor uniform case.

The Monte Carlo set includes off-centre and near-edge positions. For these, the one-hop anchors lie mainly on one side, the feasible region is lopsided, and its centroid is pulled toward the anchors. This is a systematic geometric bias that does not cancel under averaging, so the average error stays near 10 to 15% even at high density. The bounded search box, clipped at the grid edges for near-boundary positions, adds to this. The single-run tables therefore show the best case, while the Monte Carlo tables show the representative accuracy.

8 Conclusion

This experiment evaluates convex position estimation for a WSN localization scenario in NetSim Standard v15.0. AODV is used at the network layer, UDP CBR applications are configured from the anchors to the unknown sensor, and packet trace is enabled for one-hop anchor identification.

The packet trace analysis identifies one-hop anchors by checking the transmitter and receiver fields for packets received by the unknown sensor. The one-hop anchor coordinates are then used by the localization algorithm in `localization.py`. For the 100-s uniform placement runs, the one-hop anchor count increases from 13 in the 25-sensor case to 52 in the 100-sensor case. For the 100-s random placement runs, the one-hop anchor count increases from 15 in the 25-sensor case to 48 in the 100-sensor case. The random results show that anchor geometry affects localization error even when the one-hop anchor count increases.

The Monte Carlo simulation for uniform placement uses ten unknown sensor positions. The search bounds are limited to the 100 m × 100 m simulation area and shifted according to the unknown sensor position. The average error decreases from 15.45% for 25 sensors to 10.24% for 81 sensors, with 10.66% for 100 sensors. For random placement, the average error is 19.09% for

25 sensors, 17.60% for 36 sensors, 21.13% for 49 sensors, 12.73% for 64 sensors, 10.90% for 81 sensors, and 11.38% for 100 sensors. The 100-sensor uniform case gives the lowest error among the initial localization results.