Review of RFID system early testing

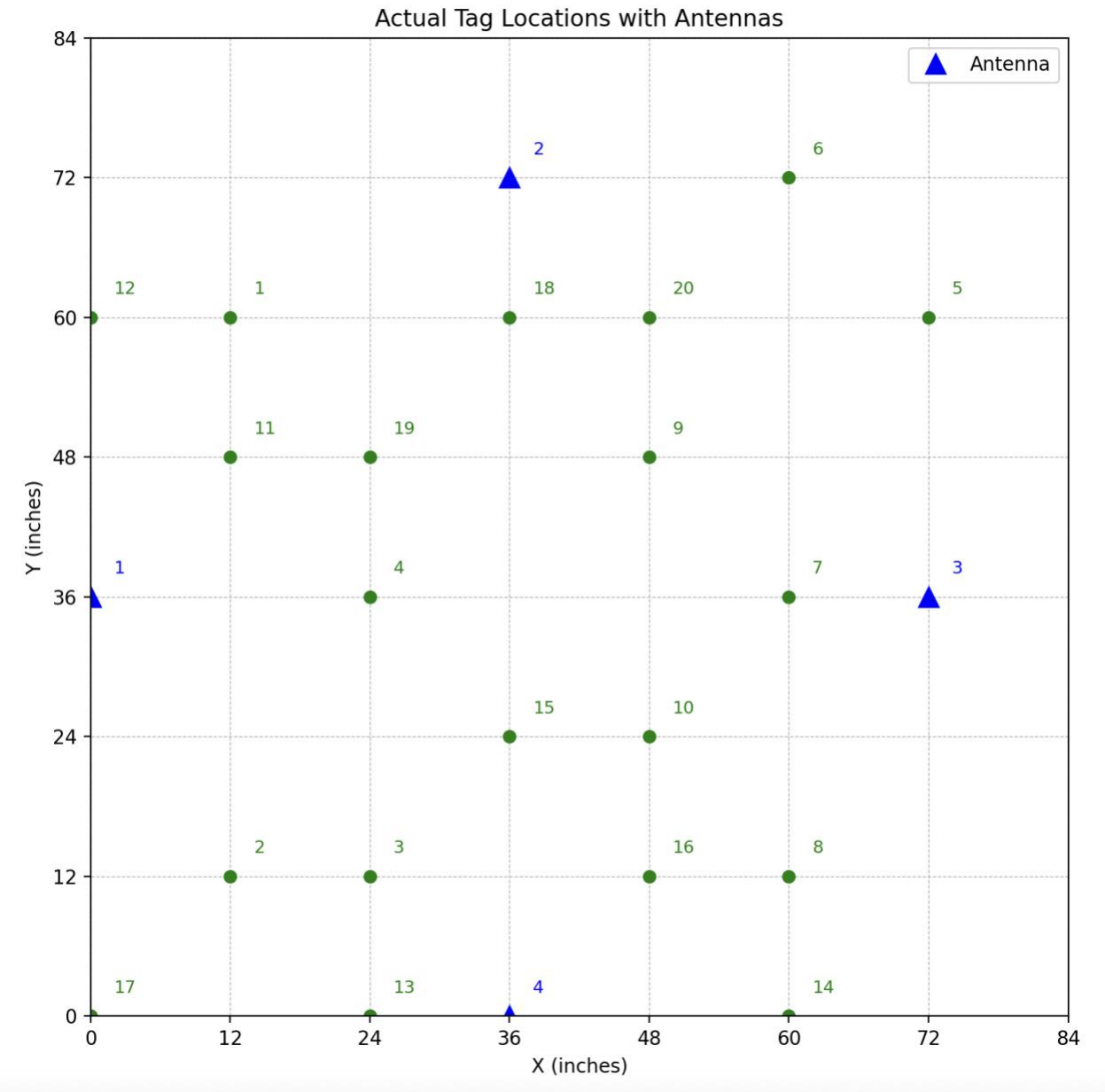
Cachengo

Overview:

This preliminary evaluation of an RFID-based location system tested the accuracy of tag positioning using data collected from four antennas via a Python script leveraging the slurp library. Due to limitations in the library, only one timestamped reading per antenna per cycle was recorded, and the dataset included redundant entries. Calibration data helped define baseline RSSI values for multilateration, but results showed significant inaccuracies in estimated tag locations. These errors were likely due to data collection constraints and RSSI variability. Future testing will use HTTP Post methods to improve data quality and explore other approaches for enhanced accuracy.

Procedure and Methods:

The graph below shows the experimental set up for location testing using all 4 antennas. The tag was attached to a small notebook and was manually placed face-up at each location during each run.



The data was acquired by running a python script utilizing the 'slurp' python libraries which proved compatible with the reader's port. The data types obtained from each run were: RunTimestamp, Timestamp, EPC, AntennaPort, RSSI, PhaseAngle, Frequency, and

DopplerFrequency. The slurp libraries were unable to obtain data from all 4 antennas simultaneously, therefore the code iterated data collection through antenna 1 to 4 each second (0.25 seconds per antenna). 3 full testing runs were done, and only the first has been analyzed for the purpose of this document.

Additionally, readings were taken from the very center of the grid to obtain calibration values for the distance formula used. The first reading was done with the tag face-up, the second reading was done with the tag sideways facing antenna 1.

See the GitHub repo for the raw csv data files from the testing: https://github.com/crozenboom/RFID-Locate/tree/main/Locate .

Analyzing the raw data and hypothesizing error origins:

The first key observation is that there was excessive repetitive data: i.e. 9 separate rows of data showing the exact same timestamp and tag values. This may be due to the slurp libraries functionality. The repetitive data itself is harmless except that it hides a more important issue: only one timestamp of data is obtained each time the code switches to an antenna. This means that while attempting to collect a large amount of calibration data by leaving the reader running for 30 seconds on each antenna before switching, only 1 timestamp of data was obtained.

Additionally, the RSSI readings fluctuated greatly across a data set. The fluctuation of RSSI values may be normal but more information about the expected accuracy of the readers will help to determine whether this will be a prevailing issue with determining location.

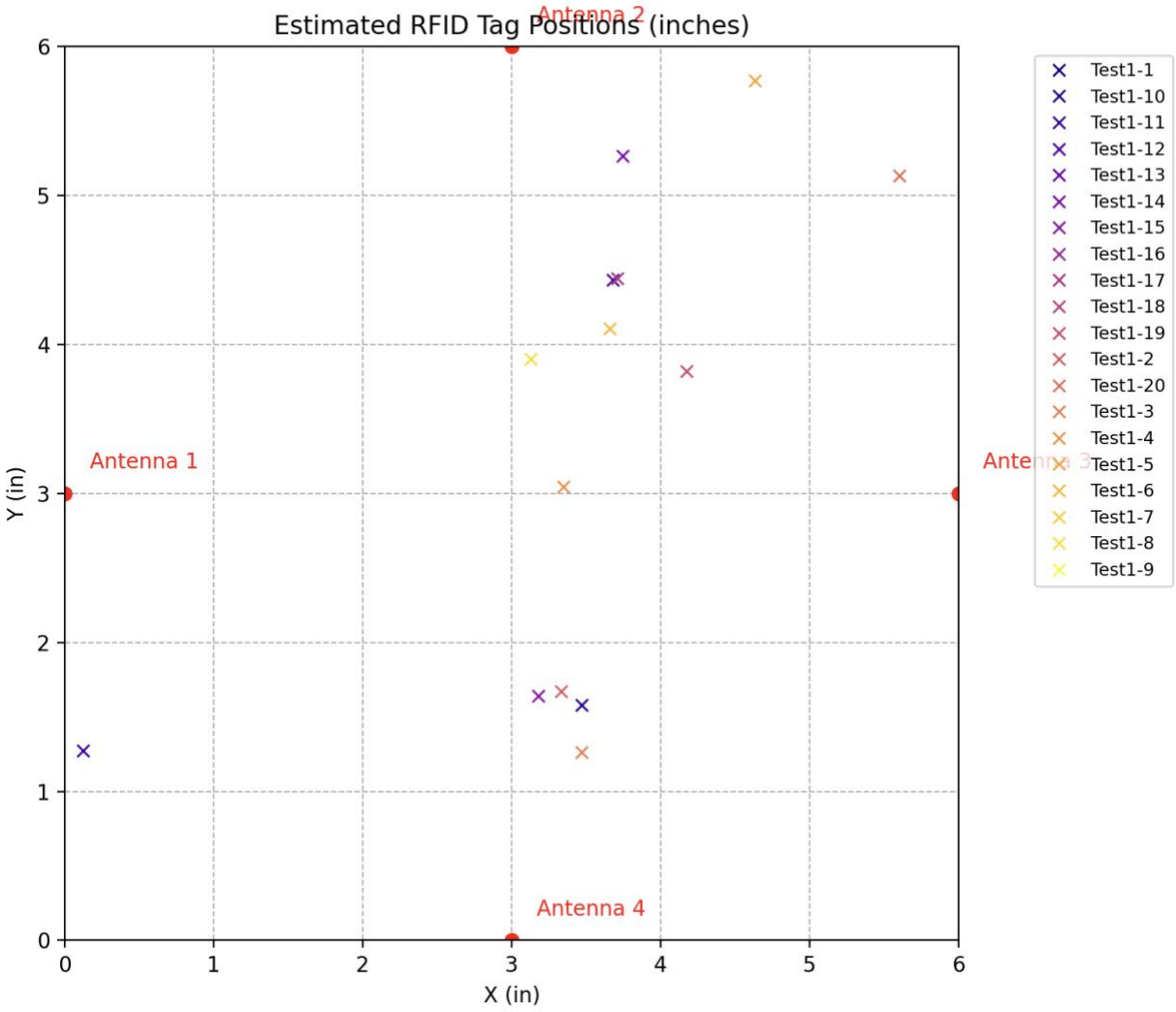
Moving forward there will be a pivot from using the slurp libraries to obtaining data via HTTP Post, so the issues stated here are arbitrary except for the context of this first round of testing and the extent of this report.

Determining tag position:

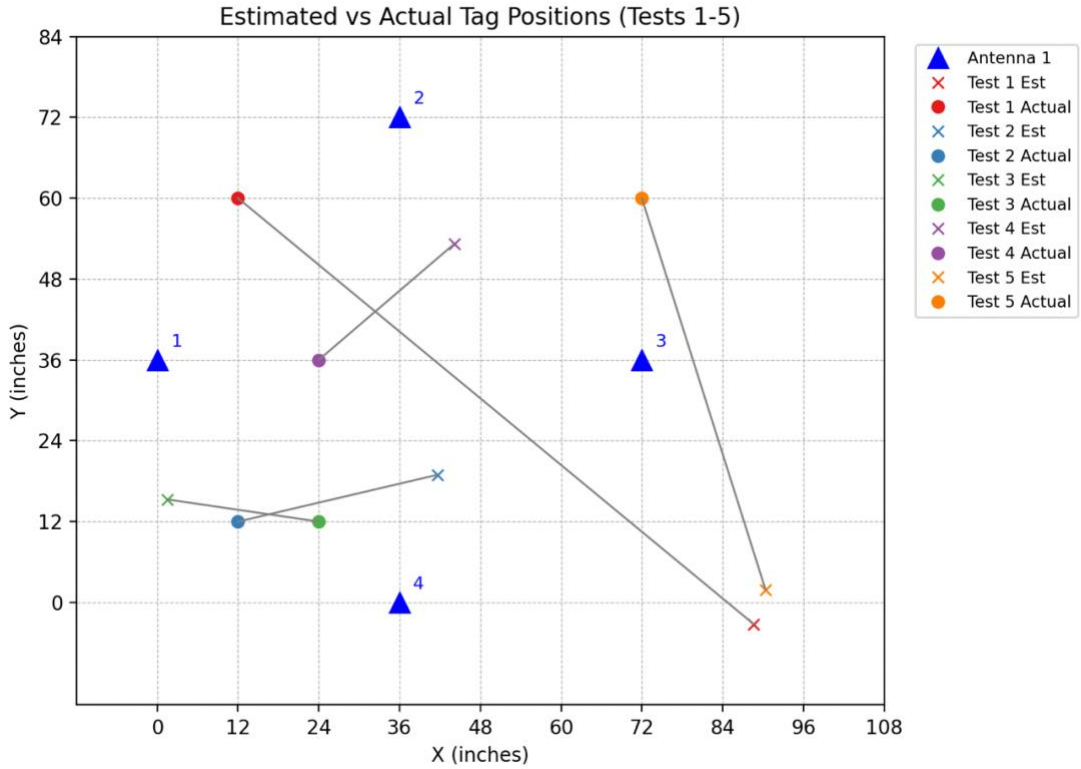
1. Parse the raw testing data and calibration data to remove repetitive rows
2. Calculate the average RSSI value across all 4 antennas for the central location used in the calibration tests (36 inches away from each antenna). This value was -65 dBm. Both values were put into the distance formula.
3. Calculate distance values for each non-repetitive row across all 20 tests

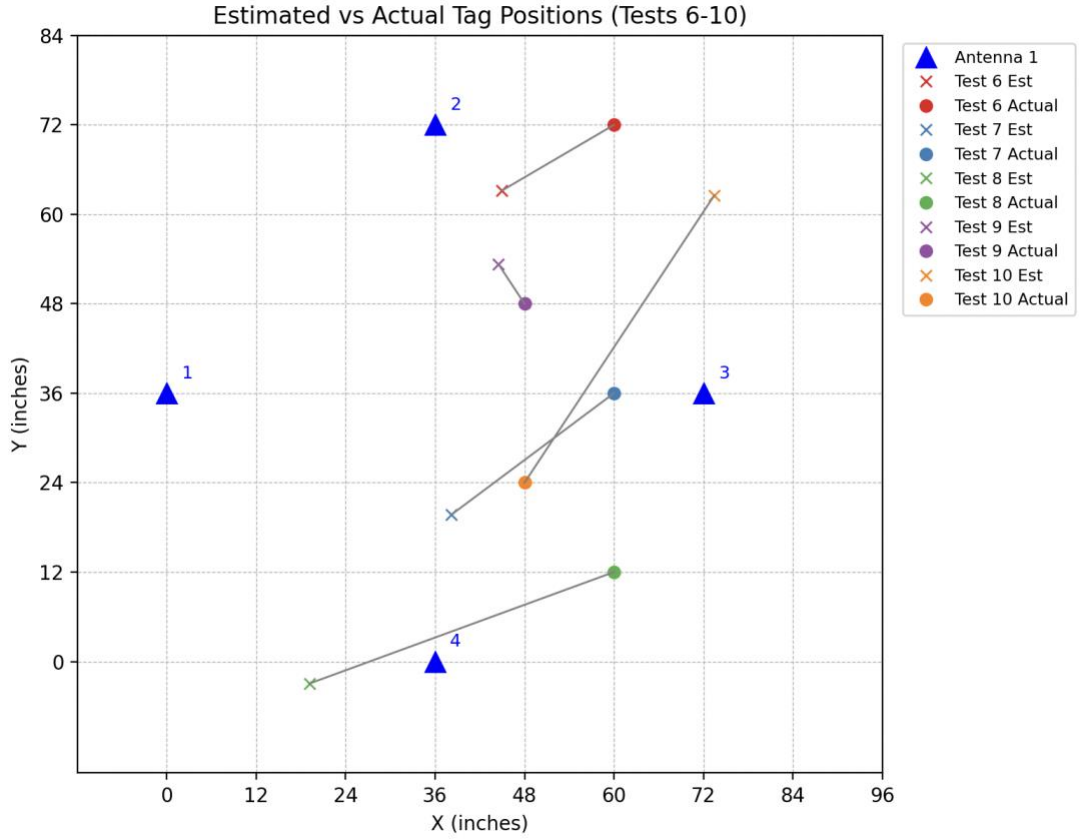
4. Calculate the average distance value for each antenna (1-4) for each test (1-20).
5. Use multilateration math to determine an estimated tag location utilizing the locations of each antenna and the distances from each antenna for every test.

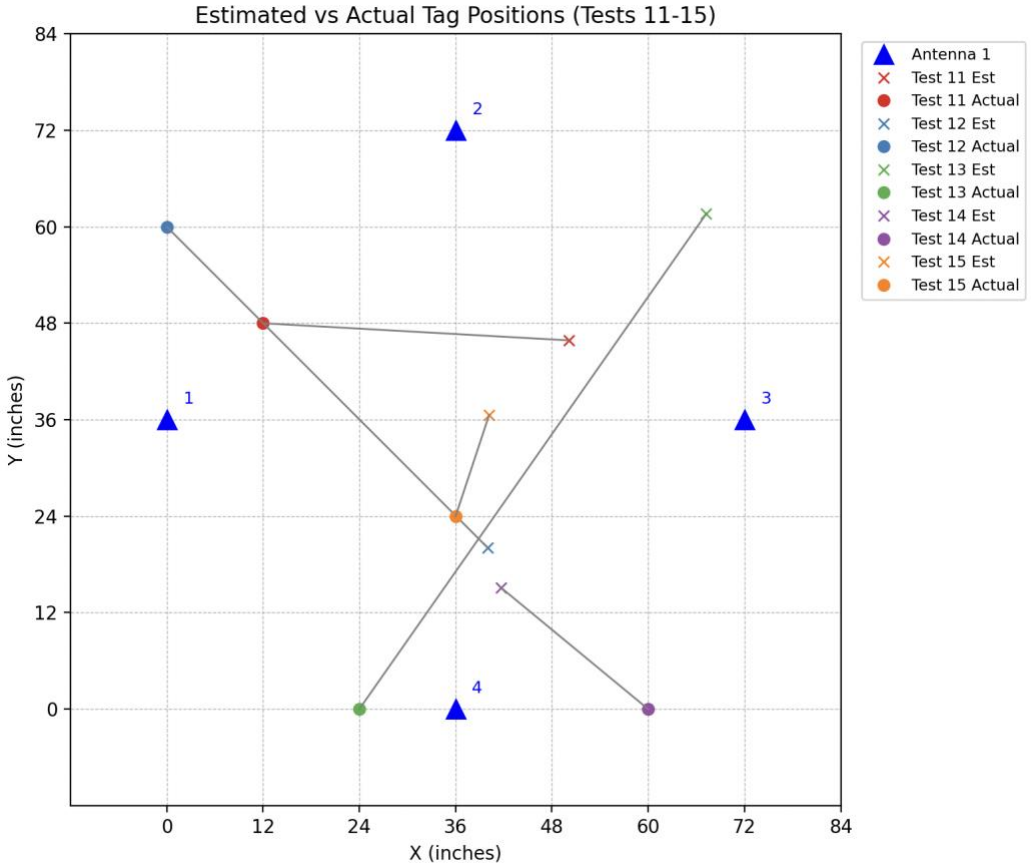
Plot of all estimated tag locations using the method above:

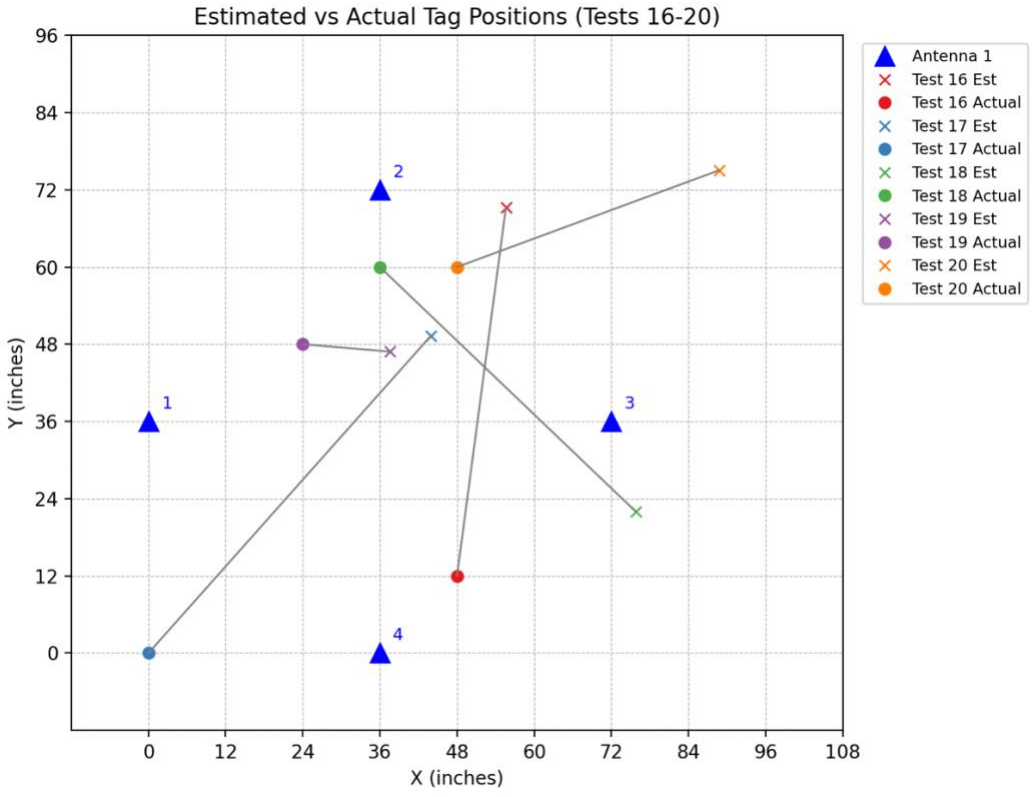


Visualizing the error:


Estimated vs Actual Tag Positions (Tests 1-5)
Antenna 1
Test 1 Est
Test 1 Actual
Test 2 Est
Test 2 Actual
Test 3 Est
Test 3 Actual
Test 4 Est
Test 4 Actual
Test 5 Est
Test 5 Actual
1
2
3
4
Y (inches)
X (inches)
0
12
24
36
48
60
72
84
96
108


Estimated vs Actual Tag Positions (Tests 6-10)
Antenna 1
Test 6 Est
Test 6 Actual
Test 7 Est
Test 7 Actual
Test 8 Est
Test 8 Actual
Test 9 Est
Test 9 Actual
Test 10 Est
Test 10 Actual
1
2
3
4
Y (inches)
X (inches)
0
12
24
36
48
60
72
84
96


Estimated vs Actual Tag Positions (Tests 11-15)
Antenna 1
Test 11 Est
Test 11 Actual
Test 12 Est
Test 12 Actual
Test 13 Est
Test 13 Actual
Test 14 Est
Test 14 Actual
Test 15 Est
Test 15 Actual
1
2
3
4
Y (inches)
X (inches)
0
12
24
36
48
60
72
84


Estimated vs Actual Tag Positions (Tests 16-20)
Antenna 1
Test 16 Est
Test 16 Actual
Test 17 Est
Test 17 Actual
Test 18 Est
Test 18 Actual
Test 19 Est
Test 19 Actual
Test 20 Est
Test 20 Actual
1
2
3
4
Y (inches)
X (inches)
0
12
24
36
48
60
72
84
96
108

Conclusions:

It is clear to see that the calculated positions of the tags are far from accurate to the actual testing locations. Before this brings you to tears, just know that this is why we are going to try to use ai. Also, better calibration will be done with the HTTP Post method to ensure that the distance formula is not the underlying issue. Stay tuned for better results.