RF Network Localization Method for Robotics Rovers

Abstract:

Researchers have been studying Global Positioning System (GPS) free localization systems in recent years due to its wide variety of potential applications. This method of localization uses a technique known as Received Signal Strength Indication (RSSI) between multiple mobile nodes, allowing each node to know the position of every other node in the system. This paper presents a new method of localization, with accuracy and efficiency surpassing previously implemented techniques. It uses a 434 MHz RF localization system that can provide less interference than typical 2.4 GHz systems such as Bluetooth and Wi-Fi technologies. This localization system is affordable, reliable indoors, and successful in environments with lots of extraneous electronic noise. Finally, the results of this paper reveal potential for integration with autonomous rover systems.

I. Introduction

In recent years, wireless network localization systems have attracted research interest due to the wide variety of potential applications. The ability to ascertain locations of units in a wireless network is invaluable in industrial, commercial, and military settings, as well as for public safety and everyday life. Industrial applications include the tracking of products and vehicles for manufacturing firms and finding specific items in warehouses. Commercially, there is an increasing need for indoor localization systems to track patients, the elderly, and people requiring close attention in residential communities, nursing homes, or hospitals. A few of the many public safety and military applications include tracking inmates in prisons, or navigating police, firefighter, and soldiers in the field.[1-3]

In addition, the growing interest in autonomously navigating Unmanned Robotic Systems (URS) that can navigate in GPS-denied environments has added to the research in this field. Recent developments of small and inexpensive transceiver units that can be fitted on the boards of these autonomously navigating robots has facilitated research in this area.[1] Such localization networks can be used to operate robots or Unmanned Aerial Vehicles (UAVs) in areas where GPS signal strength is low including indoors or in urban environments. Localization techniques provide promising solutions for autonomous indoor navigation and accurate environment scanning and mapping in the future.

Most wireless protocols operate in the 2.4 GHz range, which interferes with many other domestic wireless devices that operate on the same frequency bands. This includes WLAN access points such as microwave ovens, personal laptops, mobile devices, and Bluetooth devices.[4] In addition, most localization techniques incorporate either the IEEE 802.11 standard

(Wireless LAN)[3] or IEEE 802.15.4 (ZigBee).[5] We proposed a different frequency band to avoid unnecessary package collisions with these pre-existing communication bands.

Many studies of localization techniques have compared RSSI (Received Signal Strength Indication) with other wireless physical layer protocols such as Wi-Fi, Bluetooth, and ZigBee. RSSI provides a cost-effective and feasible method for indoor navigation.

Overall, this research has shown that we can build a position localization system using signals sent and received between units in a system. This paper offers a unique and new approach to RF localization in order to increase the accuracy and precision of the position estimates and provide a gateway to future implementation indoors and for autonomous rovers void of GPS, Wi-Fi, and Bluetooth.

A. RF Localization Methods

There are several methods of localization.

- Received Signal Strength Indication (RSSI) Based
- Angle of Arrival (AoA)
- Time of Arrival (ToA)
- Time Difference of Arrival (TDoA)

Our experiment explores the RSSI based method of localization. In comparison to AoA, ToA, and TDoA localization approaches, RSSI provides an attractive approach since it is reliable and cost-effective.[6,7] For the purposes of our experiment, we constructed a linear regression model, which we use to calibrate the RSSI values at pre-determined locations. This model is explored further in the methods section of the paper.[1]

The AoA is a common metric used in direction-based systems. This method requires additional hardware in order to measure the angle of incidence and incoming signal. While the AoA technique works well in areas with direct line of sight, but its accuracy can be affected by situations that include Non-Line of Sight (NLOS) signals and reflection from objects including walls and ceilings.[3,8] These complications make it a poor option for indoor localization systems.

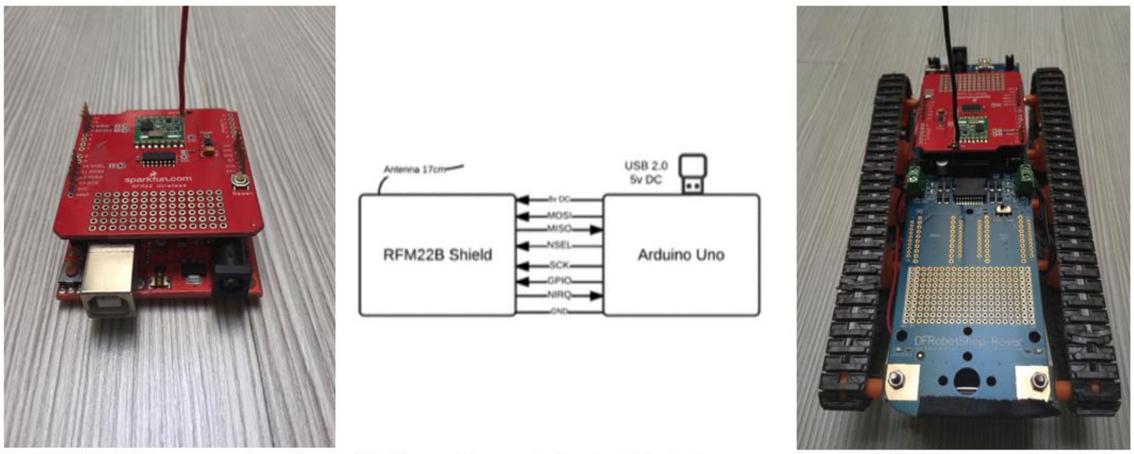
The ToA method depends on the accurate synchronization of the source and all the receivers. After the message is received, the transmission delay and speed of the signal is used to calculate the distance between the source and the receiver.[9] This method does provide an accurate localization model; the synchronization requirements and delay inaccuracies make the cost of these devices relatively high.[1]

Unlike the ToA model, the TDoA only requires synchronization between receivers. The TDoA localization technique is to determine the distance by evaluating the difference in arrival time between receivers, using the relative time at each receiver rather than the absolute time measurements.[10] The cost of these systems is relatively high; however, less expensive systems are emerging. A TDoA system comparison may be the basis of our future work.

B. Proposed Hardware

For our research, we used the RFM22B-S2 module which is a 434 MHz radio transceiver. We used a 17.3 cm conductor as a monopole antenna.

The communication between the microcontrollers (Arduino UNO and DFRobotShop Rover) and the modules was done by Arduino Serial Peripheral Interface. Each node is powered through USB and a RFM22B shield was used to route the connectors to the microcontroller boards. The DFRobotShop Rover has all the same capabilities as the Arduino UNO and is completely compatible with the Arduino IDE programming software as well as the RF22MB shield, which enabled our rover implementation. Both the Arduino unit and the DFRobotShop Rover unit are powered using USB 2.0 protocol devices. Figure 1 below shows the images of a wireless node and rover and the electrical block diagram for both.



(a) RFM22B shield connected to the Arduino UNO microcontroller board

(b) Connections and electrical block diagram of each node/rover

(c) RFM22B shield connected to the DFRobotShop Rover

Figure 1: Hardware Configuration

There are multiple advantages to the proposed hardware for our research. First, the Arduino UNO, RFM22B shield, and DFRobotShop rover are all readily available at a low cost. In addition, the Arduino UNO and DFRobotShop rover is completely compatible with the RFM22B shield and there are no additional modifications to integrate the two. Also, because of the low frequency of the RFM22B transceiver, there is no need for a complex antenna; a simple monopole antenna is all that is required. Lastly, the RadioHead library contains many API functions that are compatible with the Arduino UNO and DFRobotShop Rover. The programming of the nodes and rovers is relatively easy with the library functions that allow sending and receiving as well some more complex network and datagram configuration.

II. Methods

A. Alternative Methods of Localization

Our network localization system improves upon previously implemented system approaches tested at the University of California at Santa Cruz.[1] In previous methods tested there, a set of stationary nodes (anchor nodes) is surrounded by a singular tag node. The tag node broadcasts a wireless beam, which is received by all the stationary nodes. The received information is used by the computer to calculate the position of the tag node and this information is presented to the user.

Figure 2 below shows this previous configuration of the nodes. In this system, we set one of the anchor nodes as the main anchor node to which all the other stationary nodes send their data from the tag node. The main anchor node sends the information through USB to the client computer, which uses the RSSI values to compute the position of the tag node and display it.

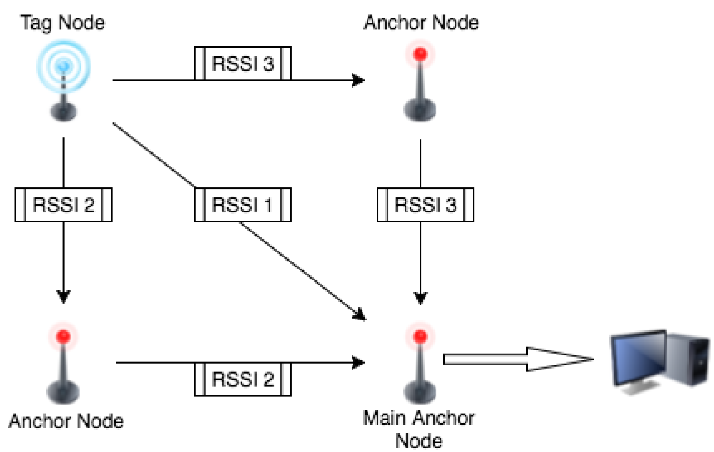


Figure 2: Previous Localization Method

We found that this method of localization could be improved in a number of ways. We proposed a system reconfiguration that would comprise only of mobile nodes, removing the need for a client computer. We also decided to implement a new system protocol and calculation method to improve the accuracy of the localization position estimates. The comparison of the two methods and their respective results can be seen in the Discussions section of the paper.

B. RF Node System Reconfiguration

Our RF Localization reconfiguration consists solely of a number of mobile nodes all stationed within the same wireless range. The system is configured so that each mobile node has the ability to both broadcast and receive from every other node, which allows for each mobile node to calculate the position of all other nodes in the network.

Figure 3 below shows this new configuration.

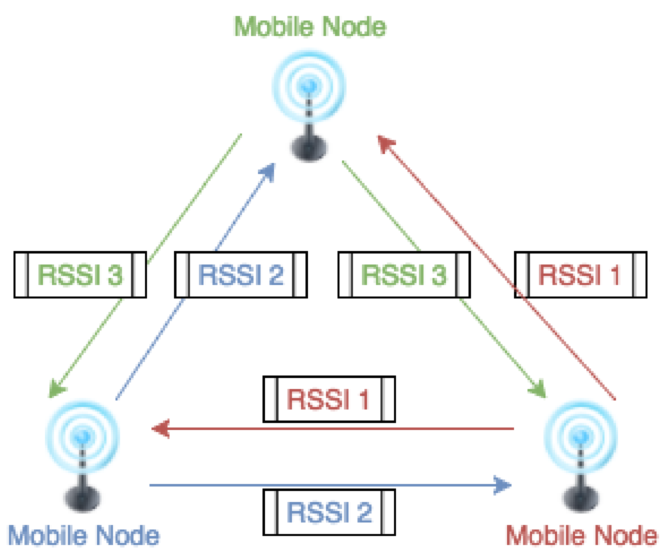


Figure 3: New Localization Method

Each node sends an RSSI signal to every other node in the system and receives an RSSI signal from every other node in the system. Unlike the old configuration, described in Section A above, where the mobile node would only send out its signal once, this configuration allows the nodes to continuously receive and broadcast ensuring that each node can know the position of each other node in the system.

C. Reliable Datagram Communication

To aid in this system reconfiguration, we proposed the implementation of a new system protocol. While in previous methods, the nodes would enter the broadcasting function after receiving a RSSI value from any single one of the nodes, we introduced what we term the “round robin” protocol, which allows the nodes to enter the broadcast function only after the node has received a message from every other node, as seen in Figure 4 below.

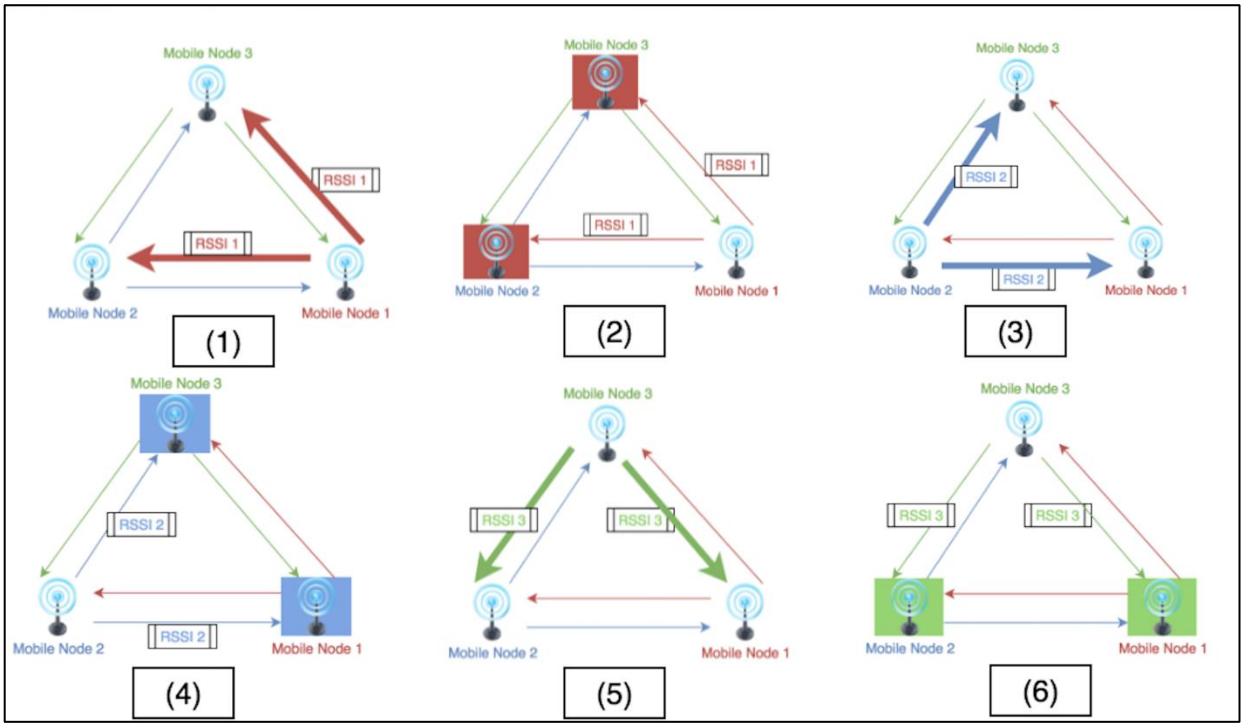


Figure 4: This figure displays the “round robin” schematic that we have implemented, described below. The basic premise of this system is that each node broadcasts and then waits to receive from the other nodes in the system. After this cycle is completed, it repeats again. This cycle begins with node 1 in the loop after the initialization and then continues successively.

The implementation for this reconfiguration begins in the Arduino setup, which acts as the initialization for each Arduino, before it enters a continuous loop. We have also implemented a unique initialization in the setup in addition to the “round robin” that each unit begins in the loop. Each node is given an address by the user that it uses to send data from one node to the next. Using the addressing, we created a function in which the node given Address 1 waits for a signal from every other node in the system one by one. After Node 1 receives a message from another node, it acknowledges that node, and that node then exits the setup and enters the loop waiting for Node 1 to enter the loop as well. Node 1 repeats this with all the nodes until they have all exited into the loop, and it then exits itself and begins the "round robin," starting with a broadcast. This initialization is very robust in its methodology because it is clean and fail-safe.

The only disadvantage of the "round robin" protocol is the increase in time between each message received, but we found that the benefits far outweigh this cost. This cost analysis is presented in greater depth in the Discussion section. Using the previous protocol, nodes would often receive from the same node every cycle, resulting in the nodes missing messages from other nodes in the system. In addition, this new configuration does not allow for collisions between messages. Through "round robin," we guarantee that each node will receive from every other node each cycle, creating a more robust system than the alternative method in Subsection A of the methods.

D. Localization Calculations

1. Free Space Path Loss (FSPL)

The Free Space Path Loss (FSPL) is the signal attenuation of a wave in a medium, and can be used in many areas to predict radio signal strengths in a radio system.[11] The principles of FSPL are central to any RF localization technique. In order to use FSPL and achieve accurate results, the RF nodes must have a line of sight in order to prevent reflection and diffraction. There are some other influencing effects on the radio signal propagation including polarization of fields, diffraction at edges, reflections of metallic objects, and refraction by media with different propagation velocity.[12]

The FSPL can be calculated as follows:[1,11]

$$\mathrm{FSPL(dB)} = 10\log(4\pi df / c)^2$$

Equation 1
d = distance
f = frequency
c = speed of light

The frequency value of 434 MHz and the speed of light can be plugged into this equation to solve for the distance:

$$d = 10^{\frac{|\mathrm{FSPL}|-25.2}{20}}$$

Equation 2

The FSPL can be calculated with the following equation:

$$\mathrm{FSPL} = \mathrm{B}_{\mathrm{dB}} - \mathrm{R}_{\mathrm{dB}}$$

Equation 3

2. Linearization and Calibration

In order to do the calibration, we start by making the assumption that the received signal strength is linearly related to the RSSI values, represented by this equation below:[1,]

$$\mathrm{R}_{\mathrm{dB}} = a\mathrm{RSSI} - b$$

Equation 4

a,b = linear coefficients from least squares calibration

Using the calibration results and the examples of fitted lines from three nodes in the calibration, we substituted equation 4 into equations 3 and 2 to derive the following analytical solution:

$$a\mathrm{RSSI}_i + b = \mathrm{B}_{\mathrm{dB}} - 20\log_{10}\sqrt{(x_m - x_i)^2 + (y_m - y_i)^2} - 25.2$$

Equation 5

x_m, y_m = coordinates for the mobile node of interest

x_i, y_i = coordinates for the i^{th} mobile node of interest

Linearization can be achieved through a first order Taylor expansion resulting in:

$$\mathrm{RSSI}_i = \frac{1}{a}\left(\mathrm{B}_{\mathrm{dB}} - 20\log 10\sqrt{(x_m(t_0) - x_i)^2 + (y_m(t_0) - y_i)^2} - 25.2 - \mathrm{b} + - \frac{20\log 10\sqrt{(x_m(t_0)-x_i)^2+(y_m(t_0)-y_i)^2}}{dxdy}\begin{bmatrix} x_m - x_m(t_0) \\ y_m - y_m(t_0) \end{bmatrix}\right)$$

Equation 6

$x_m(t_0)$, $y_m(t_0)$ = mobile node coordinates to be linearized about

This equation assumes that the initial condition for the mobile node of interest is known and the system will be linearized about the initial condition. Including all the nodes results in a system of equations:

$$RSSI \approx A\begin{bmatrix} x_m \\ y_m \end{bmatrix} + BD_{dB} + C$$

Equation 7

where the variables A, B, and C are represented by these equations below:

$$A = \begin{bmatrix} -\frac{10}{a_1}\frac{2(x_m(t_0)-x_1)}{(x_m(t_0)-x_1)^2+(y_m(t_0)-y_1)^2} & -\frac{10}{a_1}\frac{2(x_m(t_0)-x_1)}{(x_m(t_0)-x_1)^2+(y_m(t_0)-y_1)^2} \\ \vdots & \vdots \\ -\frac{10}{a_n}\frac{2(x_m(t_0)-x_n)}{(x_m(t_0)-x_n)^2+(y_m(t_0)-y_n)^2} & -\frac{10}{a_n}\frac{2(x_m(t_0)-x_n)}{(x_m(t_0)-x_n)^2+(y_m(t_0)-y_n)^2} \end{bmatrix}\begin{bmatrix} x_m \\ y_m \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{a_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{a_n} \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{1}{a_1}\left(-20\log_{10}\sqrt{(x_m(t_0)-x_1)^2+(y_m(t_0)-y_1)^2} - 25.2 - b_1\right) \\ \cdot \\ \cdot \\ \cdot \\ \frac{1}{a_n}\left(-20\log_{10}\sqrt{(x_m(t_0)-x_n)^2+(y_m(t_0)-y_n)^2} - 25.2 - b_n\right) \end{bmatrix}$$

Equations 8, 9, 10

n = number of nodes

III. Results

No attempts to isolate the testing area’s Wi-Fi, climate control, or other electronics were made, ensuring that the testing would be done in a realistic environment where interferences and disturbances are to be expected.

The nodes were tested in triangle formation, with one node in the center, through which we were receiving the RSSI values. For our first test, we kept the outer nodes stationary while moving the center node to different positions, in a manner modeling after the old method of testing. We ran these tests at various locations within the triangle in order to compare the accuracy and precision of the new system to the accuracy and precision of the old method. Figures 5-7 show this method of testing.

1. ***Three Mobile Node Configuration***

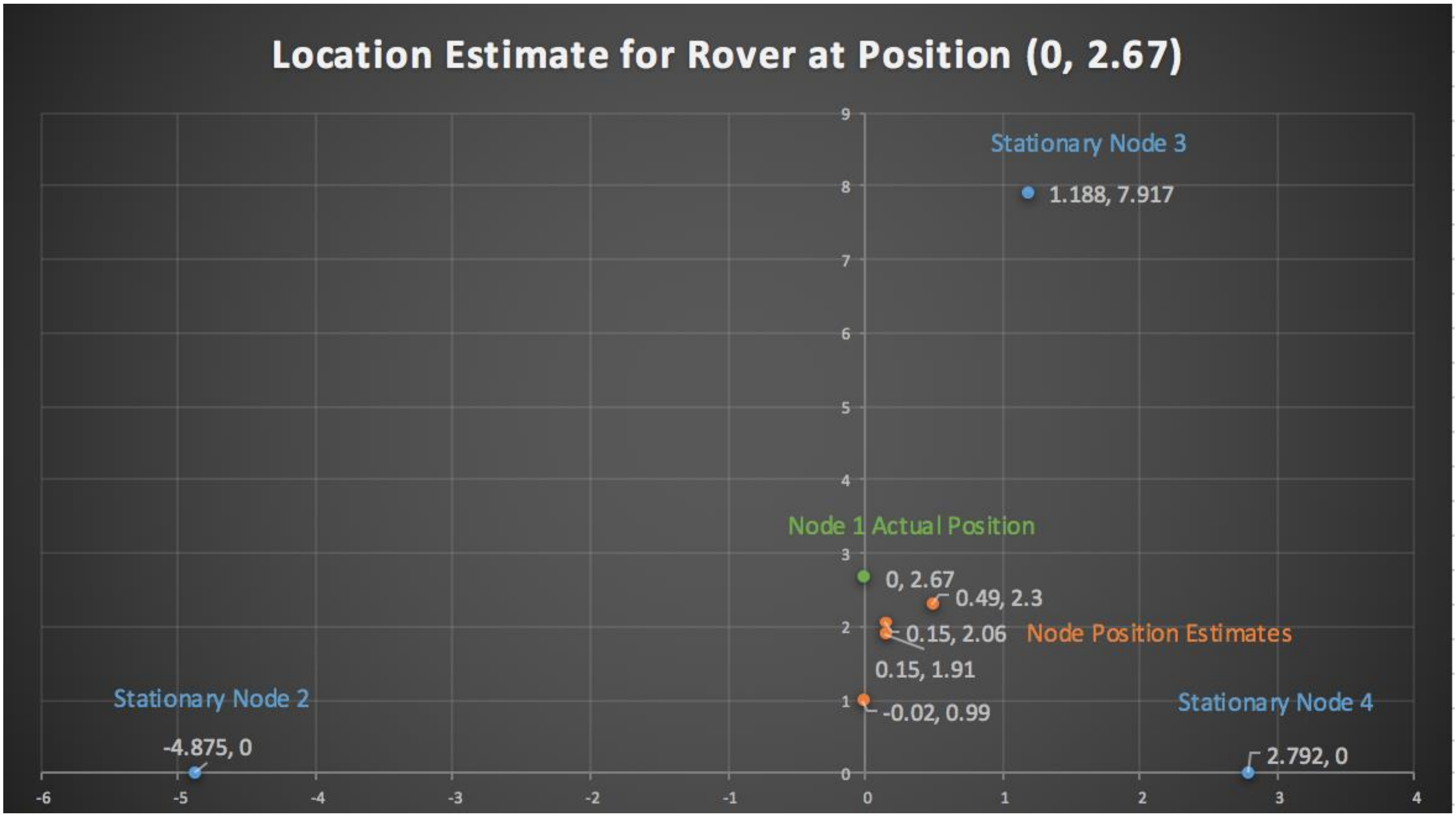


Figure 5: Each round we recalibrated the center node and placed it in the (0, 2.67) position. This graph shows the results of each test and the distance value that the center node calculated for its position.

	X (feet)	Y (feet)	Distance (feet)
Mean Error	0.2	0.32	0.92
Max Error	0.49	0.63	1.68
Min Error	0.02	0.14	0.61

Table 1: Calculated X, Y, and Distance Error at Position (0, 2.67)

The results at position (0, 2.67) showed an average error of less than 1 foot for x, y, and distance errors, indicating a high level of accuracy in model. In addition, the results were very precise, all within a 0.47-foot range in the x direction and a 1.31-foot range in the y direction.

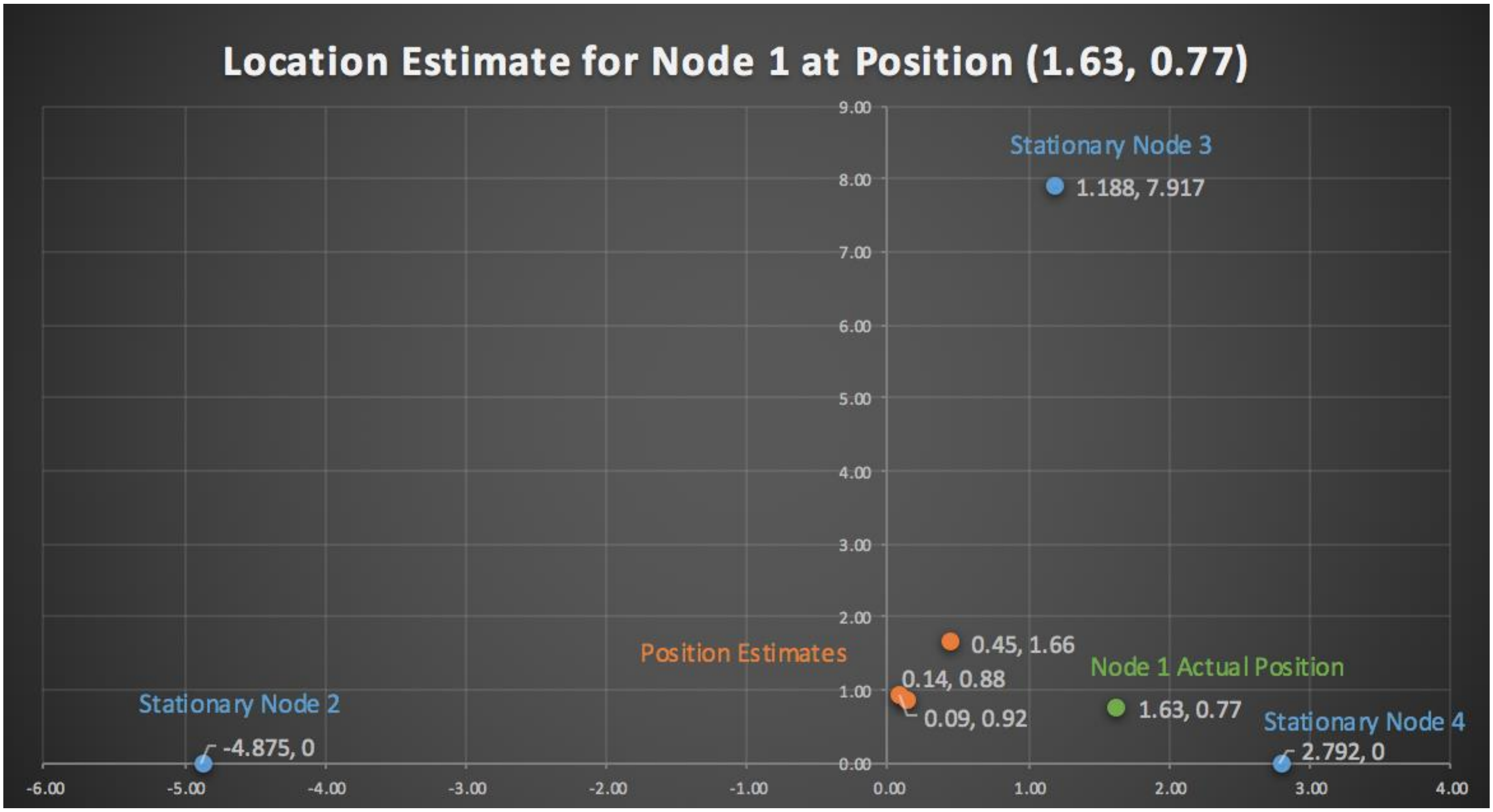


Figure 6: Each round we recalibrated the center node and placed it in the (1.63, 0.77) position. This graph shows the results of each test and the distance value that the center node calculated for its position.

	X (feet)	Y (feet)	Distance (feet)
Mean Error	0.86	0.5	1.5
Max Error	0.94	1.16	1.54
Min Error	0.72	0.14	1.48

Table 2: Calculated X, Y, and Distance Error at Position (1.63,0.77)

The results at position (1.63, 0.77) showed an average error of less than 1 foot for the x and y measurements while the distance error is equal to 1.5 feet. While the distance error was less accurate at this position, we were still able to maintain a good level of accuracy. The results were once again very precise, all within a 0.35-foot range in the x direction and a 0.74-foot range in the y direction.

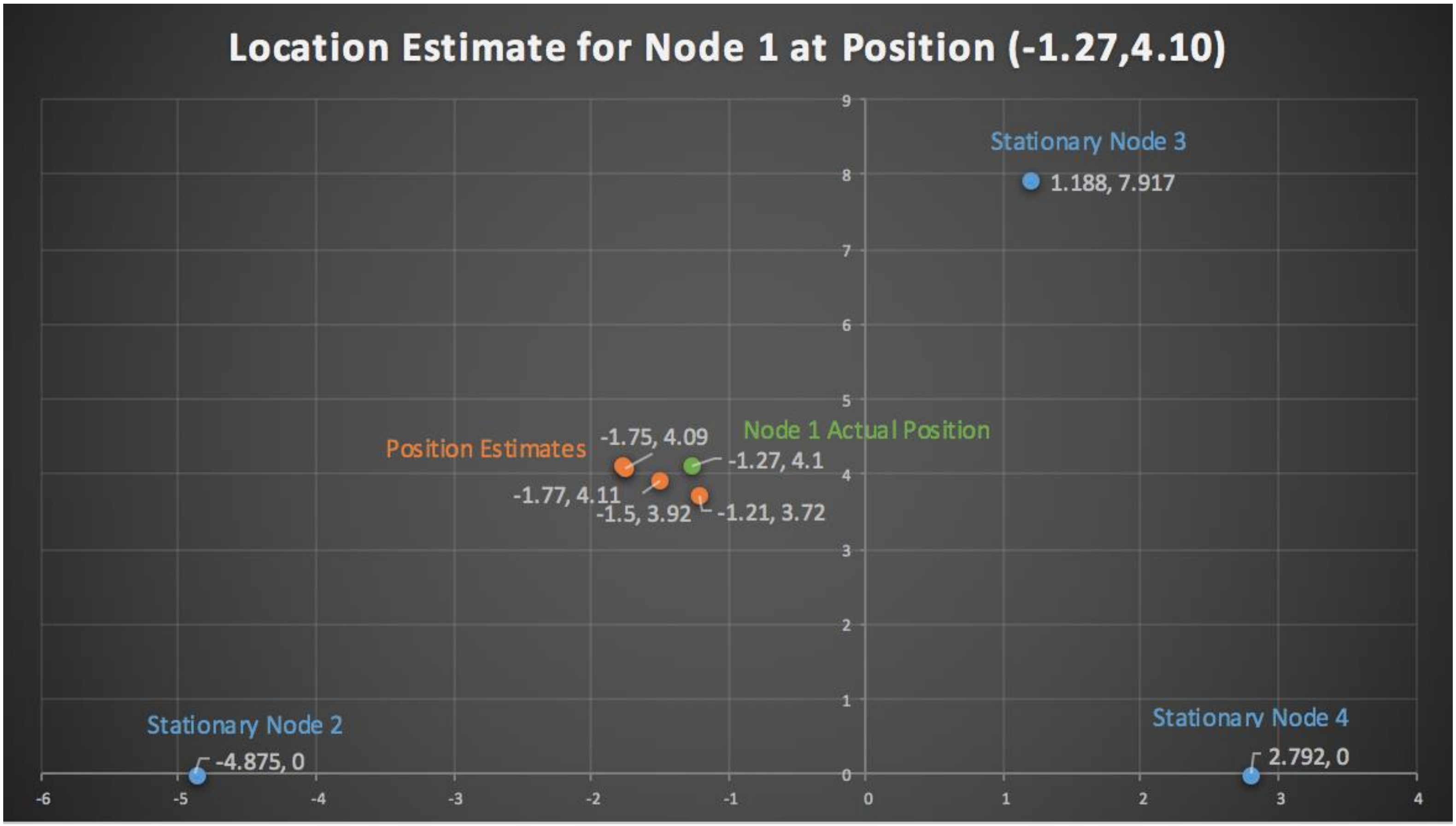


Figure 7: Each round we recalibrated the center node and placed it in the (-1.27, 4.10) position. This graph shows the results of each test and the distance value that the center node calculated for its position.

	X (feet)	Y (feet)	Distance (feet)
Mean Error	0.25	0.03	0.42
Max Error	0.39	0.09	0.5
Min Error	0.05	0.002	0.29

Table 3: Calculated X, Y, and Distance Error at Position (-1.27, 4.10)

The results at position (-1.27, 4.10) once again showed an average error of less than 1 foot for x, y, and distance errors, proving the accuracy once again. In addition, these results proved to be the most precise, all within a 0.56-foot range in the x direction and a 0.37-foot range in the y direction.

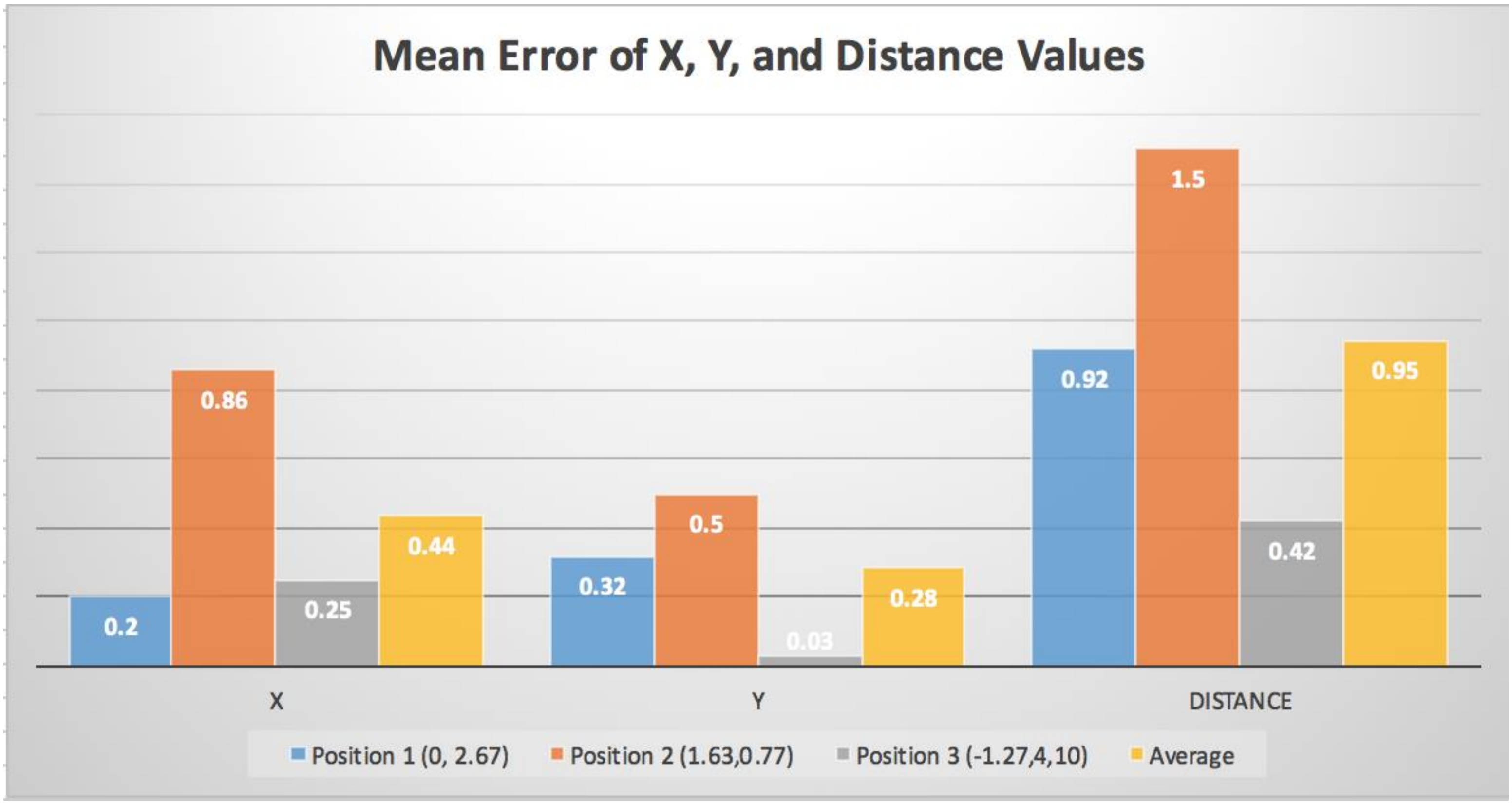


Figure 8: Comparison of the X, Y, and Distance errors from each position along with average error for each category

We compared the x, y, and distance values at the different positions, and calculated the average for each category shown in Figure 7 above. The average value for each category was lower than 1 foot, proving the accuracy of the system.

Position	**Average Position Estimate**	**Node 2 Percent Error**	**Node 3 Percent Error**	**Node 4 Percent Error**
(0,2.67)	(0.20, 1.82)	3.0%	14.9%	18.0%
(1.63, 0.77)	(0.23, 1.15)	20.0%	4.6%	22.36%
(-1.27, 4.10)	(-1.67, 4.04)	5.5%	6.09%	4.3%

Table 4: Calculated Percent Error from Each Stationary Node at Each Position

We did a percent error calculation, as seen above in Table 4, to compare the accuracy as the distance changes from each stationary node, using the average of all the position estimates at each of the three locations. We discuss the differences in the Discussion section of the paper.

2. ***Rover Implementation***

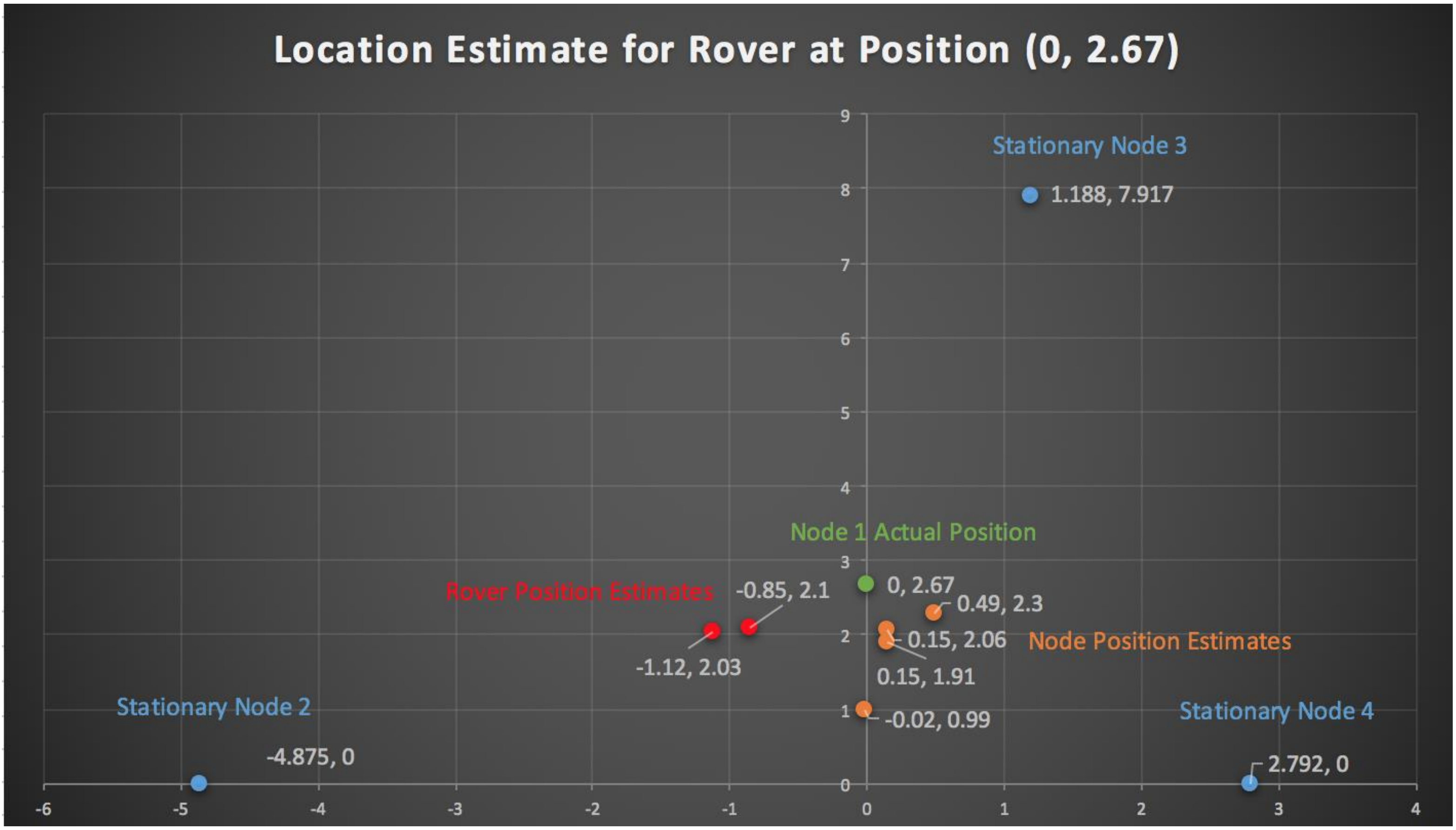


Figure 9: This graph shows the rover estimations in comparison To the position estimates of the nodes at position (0, 2.67)

	X (feet)	Y (feet)	Distance (feet)
Mean Error - rover	0.99	0.23	1.15
Mean Error - nodes	0.2	0.32	0.92

Table 5: This table compares the average error for the X, Y, and Distance categories.

We ran this experiment to test the accuracy of the rover in comparison to the nodes. We found that we were able to maintain a similar accuracy for the three categories. For the y and distance and errors, the difference in the two values were 0.09 feet and 0.23 feet respectively. However, the x error was larger at a 0.79-foot difference.

IV. Discussion

From our results, we were able to make many conclusions. First and foremost, we found that the new system was able to improve the accuracy in the X and Y directions as well as in Distance in comparison to the old implementation.[1] In the X direction, we were able to decrease the error by 0.57 feet, or 74%. In the Y direction, we decreased the error by 0.66 feet, or 67%. For Distance, we had decrease in error of 0.33 feet, or 27%. These accuracy improvements display the benefits of the new system and its methodology. While the distance error had a lower change in its accuracy, the directional errors have proved to be greatly improved by these results.

As shown in Table 4, we were able to calculate the percent error from each stationary node at each of the three locations as well. This was calculated using the equation below:

$$\%\ \text{error} = \frac{\left|\left(\sqrt{(x_e-x_i)^2+(y_e-y_i)^2}-\sqrt{(x_a-x_i)^2+(y_a-y_i)^2}\right)\right|}{\sqrt{(x_a-x_i)^2+(y_a-y_i)^2}} \text{ x } 100$$

Equations 11

x_e, y_e = x and y position estimates of mobile node
x_i, y_i = x and y position coordinates of stationary node
x_a, y_a = actual x and y position coordinates of mobile node

The percent error values that we calculated show that a closer a node is to a single stationary node, the less accurate the position estimate. For example, at position (1.63, 0.77), the percent error values were much larger than the other two positions. While the percent error calculations were 20%, 4.6%, and 22.4% for stationary nodes 2, 3, and 4 respectively, the values at the more centered positions varied around 3%, 14.9%, and 18% for position (0, 2.67) and even lower at position (-1.27, 4.10), with percent error values of 5.5%, 6.09%, and 4.3%. The reasoning for this relates to the overflow of the node buffer while collecting RSSI values. The buffer is receiving the maximum RSSI value before reaching a theoretical maximum value. Overall, we have found that the more centered a mobile node is, the greater the accuracy of the values.

Our preliminary rover research shows how plausible an autonomous robotic localization system is. We once again compared the rover results to the accuracy results of the old system.[1] While the rovers were less accurate in the X direction, with an increase an error from the old system of 0.22 feet, or 29%, both the Y and Direction errors decreased. In the Y direction, there was a decrease of 0.75 feet, or 77%, and for Distance, there was a decrease of 0.10 feet, or 8%.

Though there have been a number of benefits to the new system, there is a cost due to increased calculation time. Because each node must receive from every other node before it broadcasts, the time between each receive has increased by a factor of 3 in comparison to the old system. Quantified, the old system had an approximate 0.67 second delay between each receive, while the new system has an approximate 2 second delay between each receive. When we continue on with autonomous rovers, this movement of the system units may have a greater effect on accuracy. However, we believe that there are ways to make the system more robust and prevent the time costs from hurting the system in the future. For example, we have implemented a timeout function through which the system will reset automatically if a node has been looping in the receive mode for too long, and it has already made an effect on time between each receive.

The benefits of the new system have outweighed this cost. We have been able to improve the accuracy by a factor of three in comparison to the old system. In addition, we have been able to increase the efficiency and usefulness of a localization system through the round robin schematic. There have no collisions in the system, and each node now has the ability to receive from all the other nodes in the system. Overall, we have progressed research in this field while bringing us closer to incorporating this system into possible applications mentioned earlier in the paper. Localization has many potential applications and is highly relevant to new technologies.

We hope that the advancements we have made through our new and different implementations will benefit the future of RF localization projects.

V. Conclusion and Future Work

All in all, we were able to both improve on a previously implemented localization system and begin the preliminary research for the full implementation onto the rovers. We also showed that a 434 MHz band can operate in indoor localization systems while avoiding interferences. In comparison to the old system, we were able to improve the accuracy by a factor of 3 and maintain this accuracy with the DFRobotShop Rover.

The newly implemented “round robin” scheme now allows for each node to know the position of every other node in the system. This system is useful for both sharing addresses with each other, but it also provides a foolproof and robust system in case one of the nodes were to fail and not be able to send data.

In continuing work, we are focusing on decreasing the time between when each node receives the RSSI values from all the other nodes in order to make the system more efficient. Because each node must receive RSSI values from other nodes before sending out its own, each round can take up to 2 seconds. There are ways to streamline this process and we hope to continue improving the time in the future. In addition, we hope to further implement the code on the rover, allowing it to travel autonomously in the system while collecting data, and eventually have the system consist solely of the rovers. A potential use for this includes having the rovers maintain a certain distance from the other nodes as it travels in an indoor area, or using one rover to find a location based on the signals from other robots. These efforts enable the use of RF localization for Unmanned Robotics Systems.

References

1. Kahnemouyi, S., et al. *RF Network Localization Method for Unmanned Robotics Systems*. American Institute of Aeronautics and Astronautics.

2. Ghosh, Amitabha, et al. *Radio Frequency Based Indoor Localization in Wireless Sensor Networks*. Department of Electrical Engineering, U. of Southern California.

3. Stella, Maja, et al. *RF Localization in Indoor Environment*. Dept. of Electronics, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, U. of Split.

4. Jiang, Landu. *A WLAN Fingerprinting Based Indoor Localization Technique*. 2012. U. of Nebraska, MS thesis.

5. ZigBee Alliance Board of Directors, compiler. *ZigBee Specification*. ZigBee Standards Organization, 17 Jan. 2008.

6. Yang, Jie, and Yingying Chen. *Indoor Localization Using Improved RSS-Based Lateration Methods*. New Jersey, Department of Electrical and Computer Engineering, Stevens Institute of Technology.

7. Luo, X., et al. *Comparative Evaluation of Received Signal-Strength Index (RSSI) Based Indoor Localization Techniques for Construction Jobsites*. U. of Texas at Austin, 2010.

8. *Wi-Fi Location-Based Services 4.1 Design Guide*. San Jose, Cisco Systems, 20 May 2008.

9. S, Ravindra, and Jagadeesha S. N. *Time of Arrival Based Localization in Wireless Sensor Networks: A Linear Approach*. Jawaharlal Nehru College of Engineering, Aug. 2013.

10. Kim, Sangdeok, and Jong-Wha Chong. *An Efficient TDOA-Based Localization Algorithm without Synchronization between Base Stations*. Seoul, Department of Electronics and Computer Engineering, Hanyang University, 5 Sept. 2015.

11. Poole, Ian. "Free Space Path Loss: Details, Formulas, Calculator." *Radio-Electronics.com*, www.radio-electronics.com/info/propagation/path-loss/free-space-formula-equation.php.

12. Blumenthal, Jan, et al. *Weighted Centroid Localization in ZigBee Sensor Networks*. U. of Rostock, 2010.