AN EXPERIMENTAL COMPARISON OF WIRELESS POSITION LOCATING ALGORITHMS BASED ON RECEIVED SIGNAL STRENGTH

A Thesis

by

FELIX GUTIERREZ, JR.

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2008

Major Subject: Electrical Engineering

AN EXPERIMENTAL COMPARISON OF WIRELESS POSITION LOCATING ALGORITHMS BASED ON RECEIVED SIGNAL STRENGTH

A Thesis

by

FELIX GUTIERREZ, JR.

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Scott Miller
Committee Members,	Don Halverson
	Jim Ji
	Radu Stoleru
Head of Department,	Costas Georghiades

December 2008

Major Subject: Electrical Engineering

ABSTRACT

An Experimental Comparison of Wireless Position Locating Algorithms Based on Received Signal Strength. (December 2008)

Felix Gutierrez, Jr., B.S., The University of Texas at Austin

Chair of Advisory Committee: Dr. Scott Miller

This thesis presents and discusses research associated with locating wireless devices. Several algorithms have been developed to determine the physical location of the wireless device and a subset of these algorithms only rely on received signal strength (RSS). Two of the most promising RSS-based algorithms are the LC and dwMDS algorithms; however each algorithm has only been tested via computer simulations with different environmental parameters. To determine which algorithm performs better (i.e., produces estimates that are closer to the true location of the wireless device), a fair comparison needs to be made using the same set of data.

The goal of this research is to compare the performance of these two algorithms using not only the same set of data, but data that is collected from the field. An extensive measurement campaign at different environments provided a vast amount of data as input to these algorithms. Both of these algorithms are evaluated in a one-dimensional (straight line) and two-dimensional (grid) setting. In total, six environments were used to test these algorithms; three environments for each setting.

The results show that on average, the LC algorithm outperforms dwMDS in most of the environments. Since the same data was inputted for each algorithm, a fair comparison can be made and doesn't give any unfair advantage to any particular algorithm. In addition, since the data was taken directly from the field as opposed to computer simulations, this provides a better degree of confidence for a successful real-world implementation.

DEDICATION

To my loving family,

Teresa, Joseph, and Amy

ACKNOWLEDGEMENTS

I would like to thank my academic advisor, Dr. Scott Miller, for providing wonderful guidance and support throughout my years at Texas A&M University. I am truly honored to have worked with him and could not have had a better advisor. I'd also like to thank my colleagues in helping me conduct the measurement campaign as well as my committee, Dr. Halverson, Dr. Ji, and Dr. Stoleru, for their time and efforts in helping me complete this thesis.

I'd like to thank the faculty and staff of the Electrical and Computer Engineering Department and the Office of Graduate Studies at Texas A&M University. They have provided me with a great experience and a world-class education that I will cherish for many years.

Finally, thanks to my family for their love and support.

NOMENCLATURE

RSS	Received Signal Strength
LC	Linear Combination
dwMDS	Distributed Weighted-Multidimensional Scaling
RMSE	Root Mean Squared Error
AP	Access Point
NIC	Network Interface Card
TX	Transmitter
RX	Receiver
W	Watts
mW	Milliwatts
dB/dBm	decibels-per-watt/decibels-per-milliwatt

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

There is no doubt that the first decade of the 21st century has seen a remarkable rise in the creation and usage of wireless devices. However, with the freedom of untethered communication comes the price of knowing (with absolute certainty) the physical location of the transmitting device. To physically locate the device is a process known as "position locating." One can think of many reasons to locate wireless devices. For example, with the increased use of RFID tags on consumer products, the idea of having a building stocked with these wireless devices is not a farfetched idea. A person could use a position locating process to try and find a piece of inventory in a warehouse or perhaps a supermarket shelf. To implement such a system, how would an engineer locate the wireless device? A global positioning system is one solution, but if one considers the current cost involved per unit and a large number of units, that solution becomes infeasible. Instead, engineers have developed other means to try to locate wireless devices such as using directional antennas, timing, connectivity, and signal strength.

In the directional antenna case, a receiving unit (denoted as a "receiver" (RX)) can determine the direction of the impinging electromagnetic (EM) waves sent from a transmitting device (denoted as a "transmitter" (TX)). With a timing approach, if a RX knew when a signal was sent and recorded the time of arrival (TOA), the RX can calculate the distance to the TX by using the velocity of the EM wave (usually the speed

---

This thesis follows the style of *IEEE Transactions on Communications.*

of light in a vacuum). In a connectivity approach, a crude distance estimate is given simply if an RX can connect to a TX. If the RX can connect, one distance estimate is returned, if it can't, another distance estimate is returned. The advantage of this approach is the ease of implementation. The last approach is the one focused on in this research and that is received signal strength (RSS). If the transmitting power is known as well as a few environmental parameters, the RX can use the RSS to determine the distance to the TX. Using RSS as a way to locate wireless devices is a very attractive idea since the source of the information is readily available. If a device can simply connect to other devices, then it can try to locate itself. There is almost zero additional cost associated with implementing such a system. In fact, one would need at least three RSS measurements from different receivers in order to triangulate the physical location of the device in a 2-dimensional plane. However, the world is not perfect and obstacles can (and most likely will) interfere with readings causing random fluctuations. Obstacles and the environment in general can cause multiple time-delayed instances of EM energy to arrive at the RX from any direction. Whether a person uses the directional antenna, TOA, or RSS approach, each method has the problem of trying to accurately estimate position based on random fluctuations. Due to these random fluctuations, it seems there isn't a single universal method for estimating position.

Engineers have devised several sophisticated algorithms to try to estimate position given this type of random environment, and the two most promising algorithms based on RSS seem to be distributed weighted-multidimensional scaling (dwMDS) [1] and linear combination (LC) [2]. These two algorithms assume two different types of

devices in the wireless network: anchors and nodes. An anchor is a device that has complete knowledge of its own location. A node is a device in the network that has no knowledge of its physical location. Nodes are sometimes said to be "blindfolded," as a reference of not knowing where it is located. Nodes are able to communicate to anchors and/or other nodes. These two algorithms try to locate the position of each node in the network given the RSS between connecting nodes, and the coordinates of the anchors.

The basis of this research is to try and determine which algorithm outperforms the other. More detail about the problem will be discussed in the following section. This thesis is organized into four sections. Section 2 will discuss the problem in detail and discuss more information about the two algorithms. Section 3 will present information about the test equipment used and the acquisition of data. Section 4 will analyze the collected data and present the conclusions of the analysis.

2. PROBLEM

Of all the algorithms that estimate position based on received signal strength (RSS), two of the most promising are the linear combination (LC) algorithm and the distributed weighted-multidimensional scaling (dwMDS) algorithm. The authors of these algorithms have provided evidence to show how well their estimators perform, however each algorithm is tested using computer simulations of a wireless environment. The problem arises that a fair comparison can not be made if each algorithm is tested on a different set of data. The focus of this research is to compare these algorithms and determine which is better using the same set of data. In fact, the data will be collected from real world power measurements, not computer simulations. This will provide a fair comparison of the algorithms and also provide confidence of a successful real world implementation of these algorithms.

2.1 Comparing Algorithms

How do we compare algorithms? One way of comparing algorithms could be through efficiency such as number of computations and/or memory used. Another way is to determine which algorithm is better via performance, or how close the estimates are to the true locations. The research discussed in this thesis will investigate performance not by computer simulations, but by actual real world data. One way of measuring performance is Mean-Squared Error (MSE). MSE has had widespread use in technical literature as a measure of performance for many estimators and remains a popular choice

among engineers. If the estimator is unbiased (that is, on average, the estimator will produce a value equal to the parameter it is trying to estimate), then the MSE is a measure of the variance. The smaller the variance, the closer (on average) the estimates will be to the true locations. Rather than using MSE, a final square root will be taken to get the root mean-squared error (RMSE). The RMSE will be our measure of which algorithm performs better.

2.2 Position Locating Algorithms

The algorithms to be compared are the LC and dwMDS [1][2]. Each of these algorithms are distributed in nature and do not rely on a centralized node to calculate everyone's position. Each algorithm exploits not just connections to anchors, but connections to other nodes to help estimate position of a particular node. When a blindfolded node uses other blindfolded nodes to estimate its position, this is called "cooperative locating." Since each node has the potential to connect to every other node, a matrix can be created to store the RSS data. The RSS data stored as a 2-D matrix is known as a "power matrix." The diagonals of the power matrix will be null, and the off-diagonal numbers are valid power measurements between nodes. The terms "RSS" and "power measurements" will be used interchangeably throughout this document. Each of the algorithms uses a power matrix as the input data.

Both algorithms also use range estimators to convert power measurements into distances. These estimators are based upon a "log-normal shadowing" model of the environment [3, p. 139]. This model states that at a distance of d, the RSS will have a

log-normal distribution (normal in dB) and will be $P_{RSS,dB} = p_{o,dB} - 10n\log_{10}(d/d_o) - X$ where X is a zero-mean Gaussian random variable with variance, σ^2. d_o is some reference distance (ex: 1 meter) from a TX and $p_{o,dB}$ is the received power at that distance in dB. n is referred to as the "path loss exponent." Using this model, the developers of the LC algorithm [2] created an unbiased range estimator of the form $d = cd_o(P / p_o)^{-1/n}$ where $c = \exp(-b^2/2)$ and $b = (\sigma \ln 10)/(10n)$, while dwMDS uses an maximum likelihood (ML) range estimator of the form $d = d_o(P / p_o)^{-1/n}$. Note that P and p_o are in watts, not dB.

Parameters n and σ can be calculated from the power measurements, and more information of these two parameters will be discussed in section 4.1.1. The two other parameters used by the range estimators (d_o and p_o) are usually measured and calculated beforehand. The units of d_o (meters, feet, inches) will be the same units used for RMSE. The units of p_o (dBm) will be the same units used in the power matrix.

2.2.1 LC

This section is to provide some background of the LC algorithm. The LC algorithm was developed by Texas A&M graduate student, Wei-Yu Chen as part of his dissertation. The details of the algorithm have been provided via a technical report, but have yet to be released as public information. The algorithm can be divided into two parts. The first part is an initialization in which each node obtains an initial position estimate based upon the anchors. Once an initial estimate is found, then cooperative

locating begins and each node uses other nodes (and anchors) to estimate its final position. The algorithm relies on assigning weights to each RSS measurement and linearly combining the data to form a position estimate. The algorithm is iterative-based and each iteration moves the estimate until a stopping criterion is met. The larger the stopping criterion, the earlier the algorithm will stop and tend to have fewer iterations. In this research, a small stopping criterion is used to allow the algorithm to converge to a final estimate. The number of iterations is usually dependent on the density of nodes in the network, but one can expect to have a range of a couple to a few dozen iterations before a final estimate is reached. For further detail of the LC algorithm, please consult Appendix B.2 or [2].

2.2.2 dwMDS

This section is to provide some background of the dwMDS algorithm. The dwMDS algorithm was developed by Neal Patwari, Jose Costa, and Alfred Hero. The algorithm focuses on weighting RSS measurements as well, however by minimizing a cost function. Details of the algorithm can be found in [2]. Like LC, this algorithm is iterative-based and each iteration moves the estimates until a stopping criterion is met. The algorithm also comes equipped with the ability to select neighbors and will be discussed in section 4.1.4. The algorithm, by itself, assumes an initial position estimate is already provided for each node, and the authors of dwMDS do give suggestions on how to create this initial guess. They propose a 2-stage approach based on a random initialization. Essentially, a random (uniform distribution) initialization is used for each

node and dwMDS is run twice. The first run has a loose stopping criterion and is used to move the random estimates into a rough estimate. Based upon the rough estimate, weights are reassigned and the algorithm is rerun with a finer stopping criterion to arrive at a final position estimate. The weighting used in this algorithm is not strictly defined, but the authors suggest using an exponential form and as such, this weighting technique will be used in the analysis section. For further detail of dwMDS, please consult Appendix B.1 or [1].

3. TEST SETUP AND DATA ACQUISITION

In this section, the test equipment and test environment will be discussed. Figures of the data collected will also be presented.

3.1 One-Dimensional Testing

There were three environments chosen to conduct the one-dimensional measurement campaign. In each environment, eleven equally spaced locations were selected along an imaginary line. Each of the locations was separated by 2.5 feet (76.2 cm). With 11 locations, the total length of the track was 25 feet (7.62 m). Figure 1 provides a top view illustration of the node locations. Given eleven locations, the end locations can be considered the anchors and the nine interior locations can be considered the blindfolded nodes.

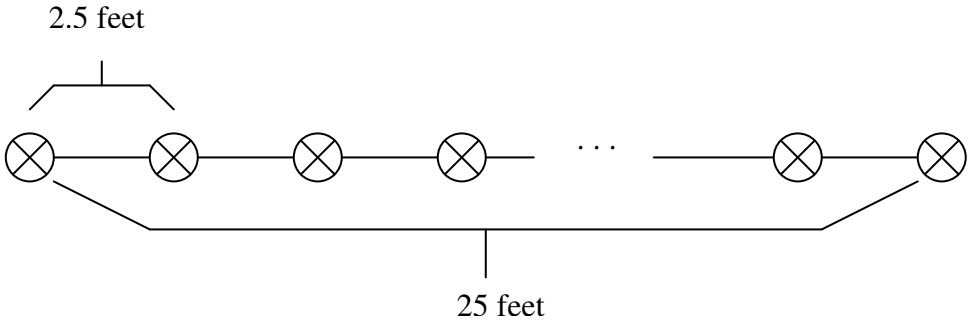

Fig. 1. A typical layout for one-dimensional testing. Eleven equally spaced locations are selected along a line. Each location is separated by 2.5 feet (76.2 cm).

3.1.1 1-D Locations

Three environments were chosen for the one-dimensional testing: an office space, a hallway, and a large indoor open space. The office space chosen was a vacant area of offices located on the 2nd floor of Wisenbaker Engineering Research Center (WERC) located at Texas A&M University. The entire track of locations spanned across three offices as seen in Figure 2.

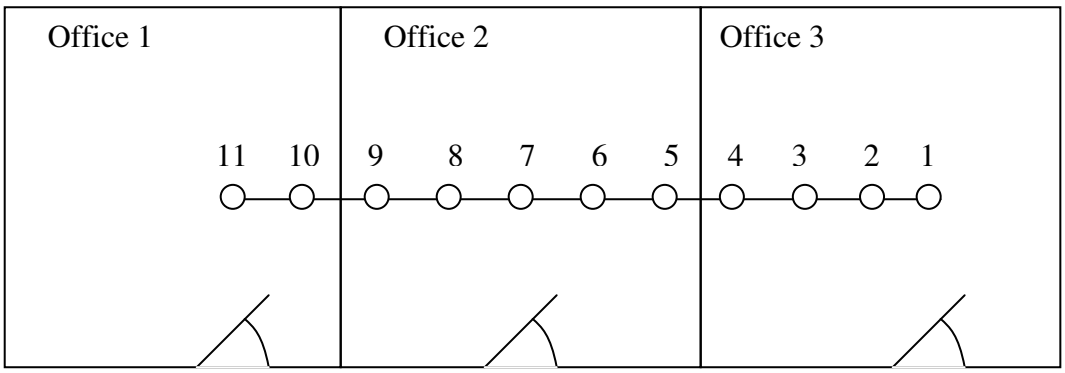

Fig. 2. Layout of the track in WERC. The track spanned 3 offices. Walls separated location 9&10 and 4&5.

The 2nd and 3rd one-dimensional environments were both located in Zachry Engineering Center at Texas A&M University. The hallway was on the ground floor and the open space was a 1st floor lobby area. There were no walls or obstacles between the nodes in these two environments as seen in the office-type environment.

3.1.2 1-D Equipment

In order to record real world power measurements, both a transmitter and receiver are needed. The TX must be transmitting at a constant power. Since the position locating algorithms only require path loss exponents of the environment, any frequency of operation can be used. Since there are many commercially available hardware and software products designed for 2.44 GHz, this unlicensed frequency band was chosen as the frequency of operation to collect the power measurements. The chosen TX was an off-the-shelf D-Link® DI-514 wireless (802.11b) access point (AP) set up to simply broadcast its service set identifier (SSID) at its maximum power of 17 dBm [4]. The wireless AP has a single omni-directional antenna. The RX was chosen to be an off-the-shelf Zonet® ZEW1501 (802.11b/g) wireless network interface card (NIC) [5]. The NIC has a standard PC Card connection to a laptop computer. Finally, a commercial software program called WirelessMon™ [6], developed by Passmark™ Software, was installed on the wireless-enabled laptop in order to monitor and record real-time signal strengths from any nearby access points. A screenshot of WirelessMon™ in action can be seen in Figure 3. Using the wireless AP, the wireless laptop, and the monitoring software, real world power measurements can be collected at all three environments.

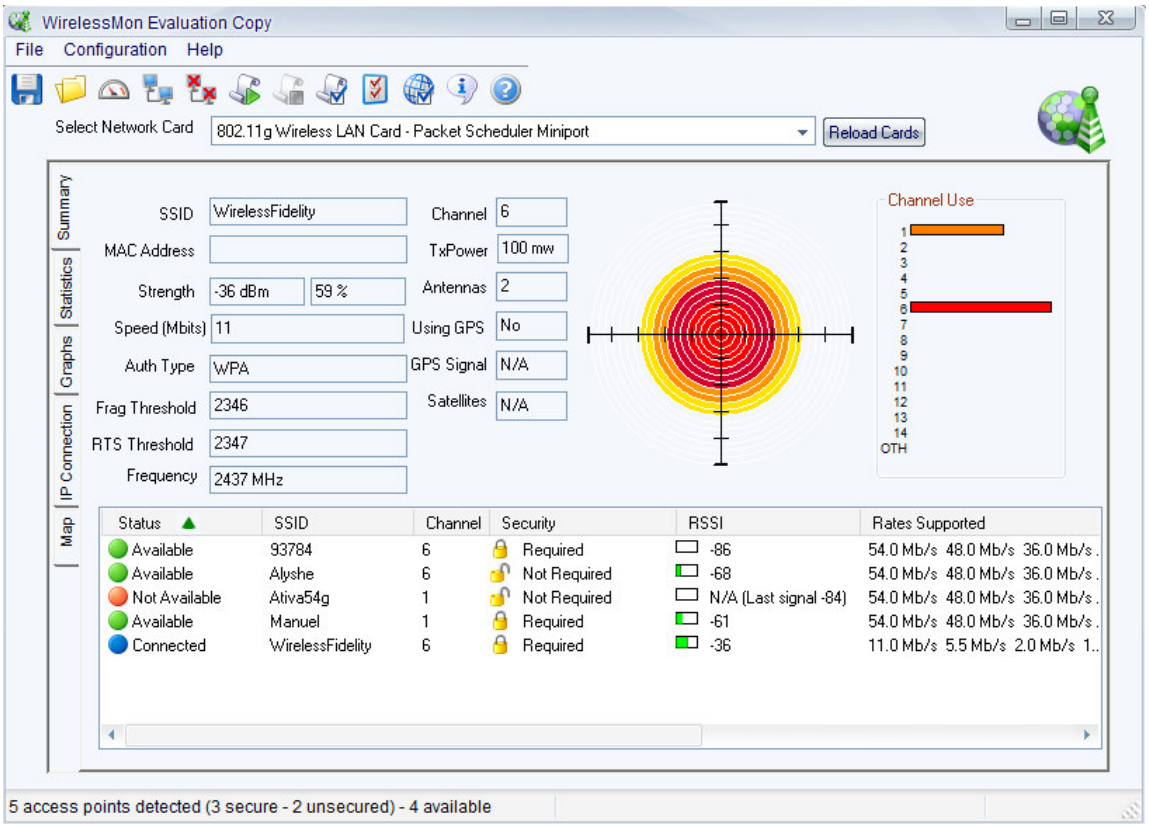

Fig. 3. A screenshot of WirelessMon, the software used to collect RSS measurements.

With the environments selected and the equipment set up, markings were labeled on the floor of each environment to indicate where the recording equipment should be placed. The TX equipment was placed and centered on a moving cart. The receiving laptop was also placed and centered on a different moving cart. Both carts were of equal height. Also, the equipment was placed in such a way that the antennas were slightly overhanging the edge of the carts to try to minimize any grounding effects from the carts. The antennas of the equipment were centered directly above the markings on the floor. Figure 4 provides a sample illustration. It should be noted that during the recording process, the RSS reading would fluctuate due to randomness in the

environment. A final reading and value of RSS was recorded when it appeared that the fluctuation stabilized and was on a constant received power.

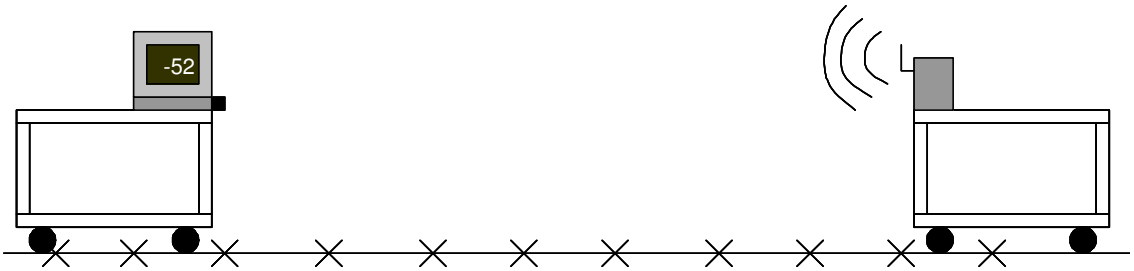

Fig 4. An illustration of how the 1-D test equipment was arranged.

Unless the recording of power measurements is fully automated, any researchers in the environment being tested will have some influence on the received power. During recording, the researchers tried as much as possible not to influence the results. The researchers stood as far as possible from the antennas and tried not to obstruct any line of sight path. It is believed that if there was any influence simply caused by the presence of the researchers, it would be represented by the shadowing variance, σ^2. In addition, from a practical point of view, if a company or organization decided to implement a position locating system, there will probably be humans and/or obstacles in the environment while the system is active and operating. Thus, it seems that the presence of researchers in the test environment seems like a more realistic scenario as compared to no researchers at all.

3.2 Two-Dimensional Testing

With two-dimensional testing, the number of anchors usually is increased from two to four. Additionally, based upon simulations from the authors of the algorithms [1][2], the density of nodes also needs to increase to get accurate location estimates. This increased density requires more locations and thus, more time to record measurements. Luckily, there were a few previously recorded power measurements taken from other researchers. The first set of data came from a previous Texas A&M student by the name of Rajukumar Samuel. This set of data contains 15 locations arranged in a 10 x 20 meter grid inside a corridor. Each adjacent node was separated by five meters as seen in Figure 5. Nodes 5, 15, 1, and 11 will be considered the anchors. The second set of data came from Dr. Neal Patwari (one of the key researchers of dwMDS) with support from Motorola Labs' Florida Communications Research Lab. This set of data is currently available to the public on Dr. Patwari's website and consists of power measurements from a remarkable 44-node array [7]. The amount of time and effort in recording these measurements is quite remarkable and duly noted. Figure 6 shows a top view of how this 44-node array was arranged. Nodes 1, 11, 36, and 44 will be considered the anchors. With two sets of data readily available, there was only need for one more environment to record 2-D measurements.

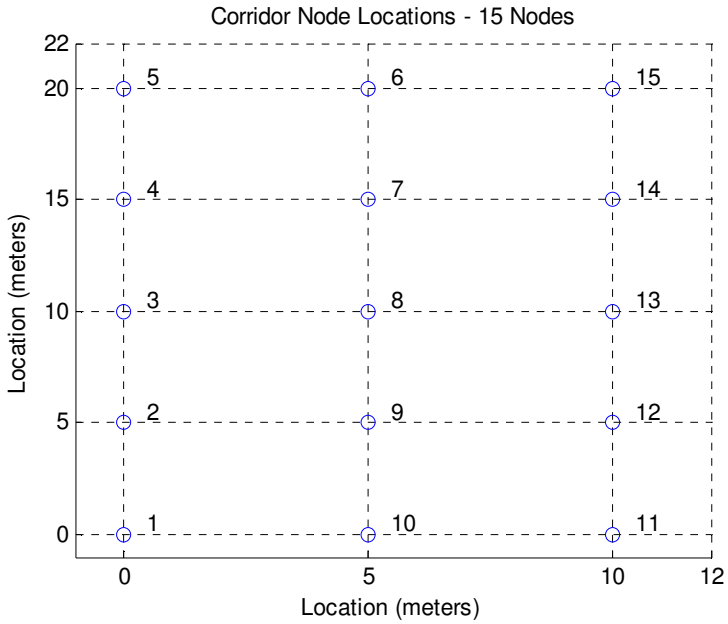

Fig. 5. Node locations for corridor environment arranged in 10 x 20 grid. Data courtesy of Rajukumar Samuel.

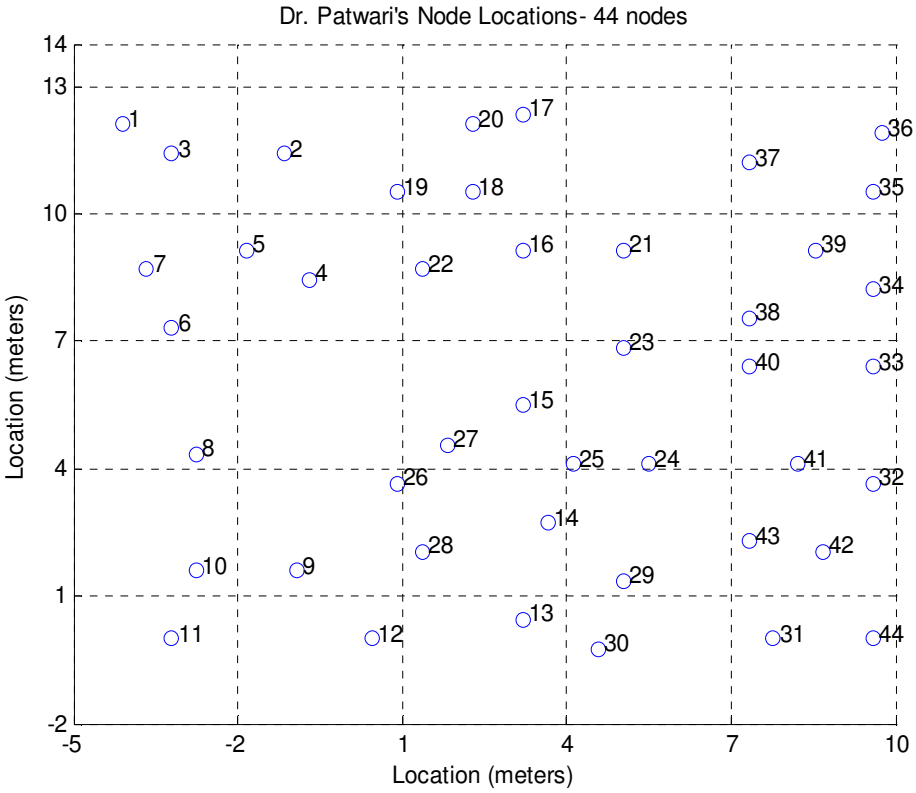

Fig. 6. Dr. Patwari's node placement in a 14x14 meter cubicle environment [7].

3.2.1 2-D Location

The 2-D case seems more likely to be implemented in a real world setting compared to 1-D, and as such, an environment was chosen to try and reflect a real world application. It was decided to find an environment that mimicked a warehouse or supermarket that consisted of metallic shelves with various items. A library was the perfect environment. The Sterling C. Evans Library located on the Texas A&M University campus was chosen to conduct the 2-D power measurements. A square area of roughly 40x40 feet (12.2x12.2 m) located at the corner of the 1st floor was chosen. This area had seven library stacks filled with books and encyclopedias. In all, 16 possible locations were selected to form a rough 4x4 node grid. The grid was not exactly uniformly spaced since some locations made it difficult to place the equipment. As such, a compromise had to be made at some locations in order to have the recording equipment placed directly on the shelves. All locations were placed at the same shelf height. Figure 7 displays a top view of where the nodes were located. Nodes 1, 4, 13, and 16 will be considered the anchors.

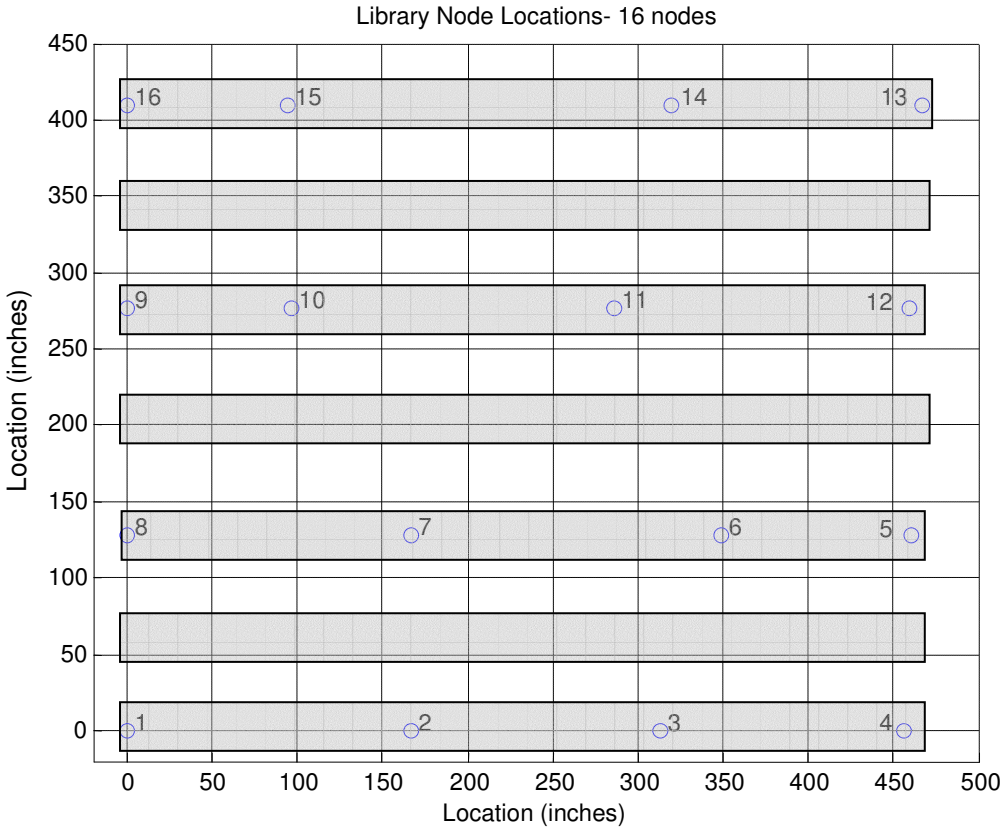

Fig. 7. Node locations for the 2-D environment. Nodes were placed in a rough grid formation of 4x4. The grey bars represent the library stacks.

3.2.2 2-D Equipment

Seeing an opportunity to cut down the measurement time in half, two transceivers were used instead of a transmitter and receiver separately. When placed in an *ad-hoc* mode, wireless NICs can create a personal wireless network and broadcast a unique SSID. This allows a laptop to act as both a transmitter and receiver, or in other words, a transceiver (XCVR). The wireless laptop used in the 1-D case was reused in this experiment as the first XCVR. A second laptop was used as the second XCVR and was equipped with an integrated Intel® 3945ABG (802.11a/b/g) wireless NIC [8]. Both

laptops were installed with the WirelessMon™ software and able to record RSS values from each other. Figure 8 provides a sample illustration.

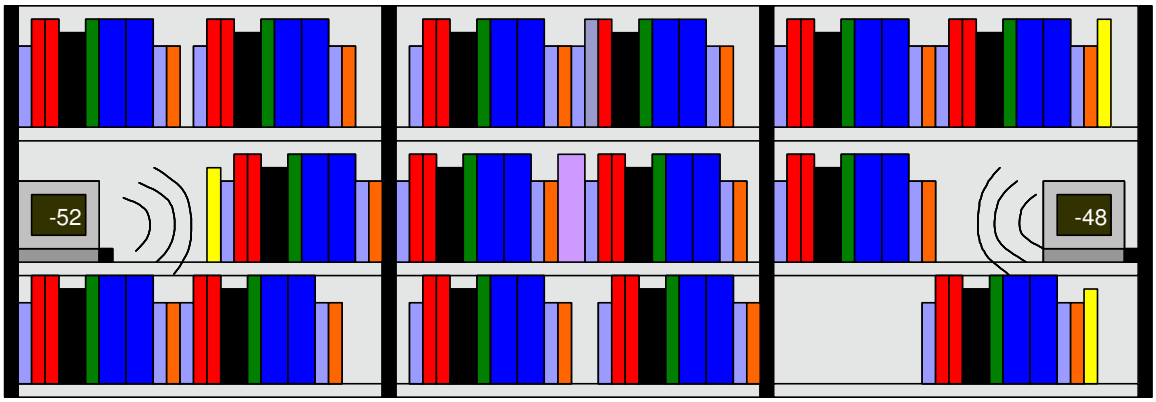

Fig. 8. An illustration of how the 2-D test equipment was arranged in the library.

By having two XCVRs, there was some savings in the amount of time to record data. With this savings, instead of recording a single value of RSS, ten RSS values were recorded as it fluctuated. These ten measurements can be averaged to get a final reading, but can also be used to calculate a more accurate value of the path loss exponent, n, and the shadowing variance, σ^2.

3.3 Total Data Collected

This section will display the collected data from both the 1-D and 2-D environments. Figure 9 shows the results for the three 1-D environments, and Figure 10 shows the results for the 2-D environment. Lastly, Figure 11 shows the averaged data for the 2-D environment. The power matrices and raw numbers can be found in Appendix A in Tables 4 – 8. Dr. Patwari's raw data can be downloaded from his website [7].

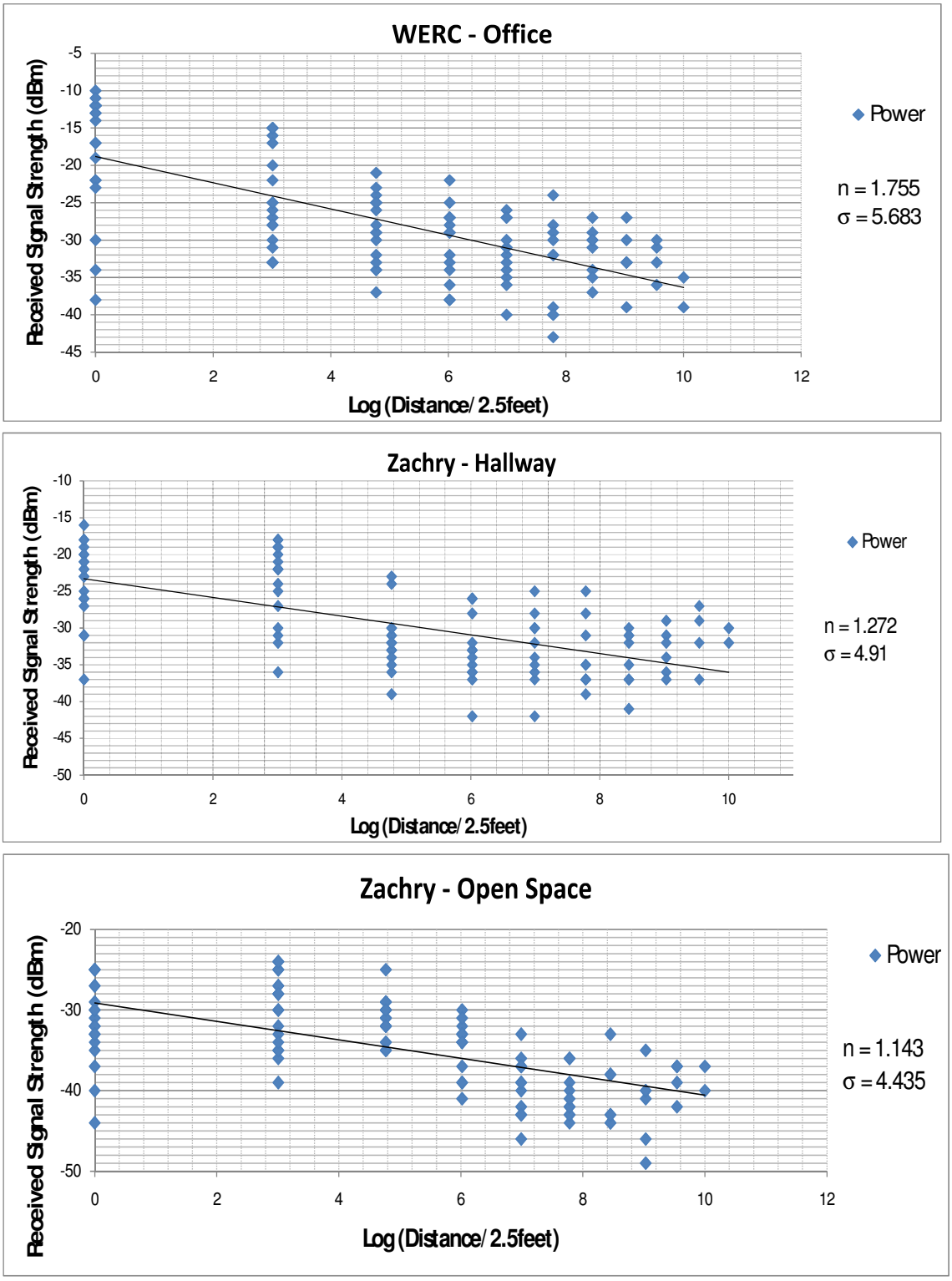

Fig. 9. The RSS measurements for the three 1-D test environments. Both path loss exponent, *n*, and shadowing variance, σ^2, have been calculated.

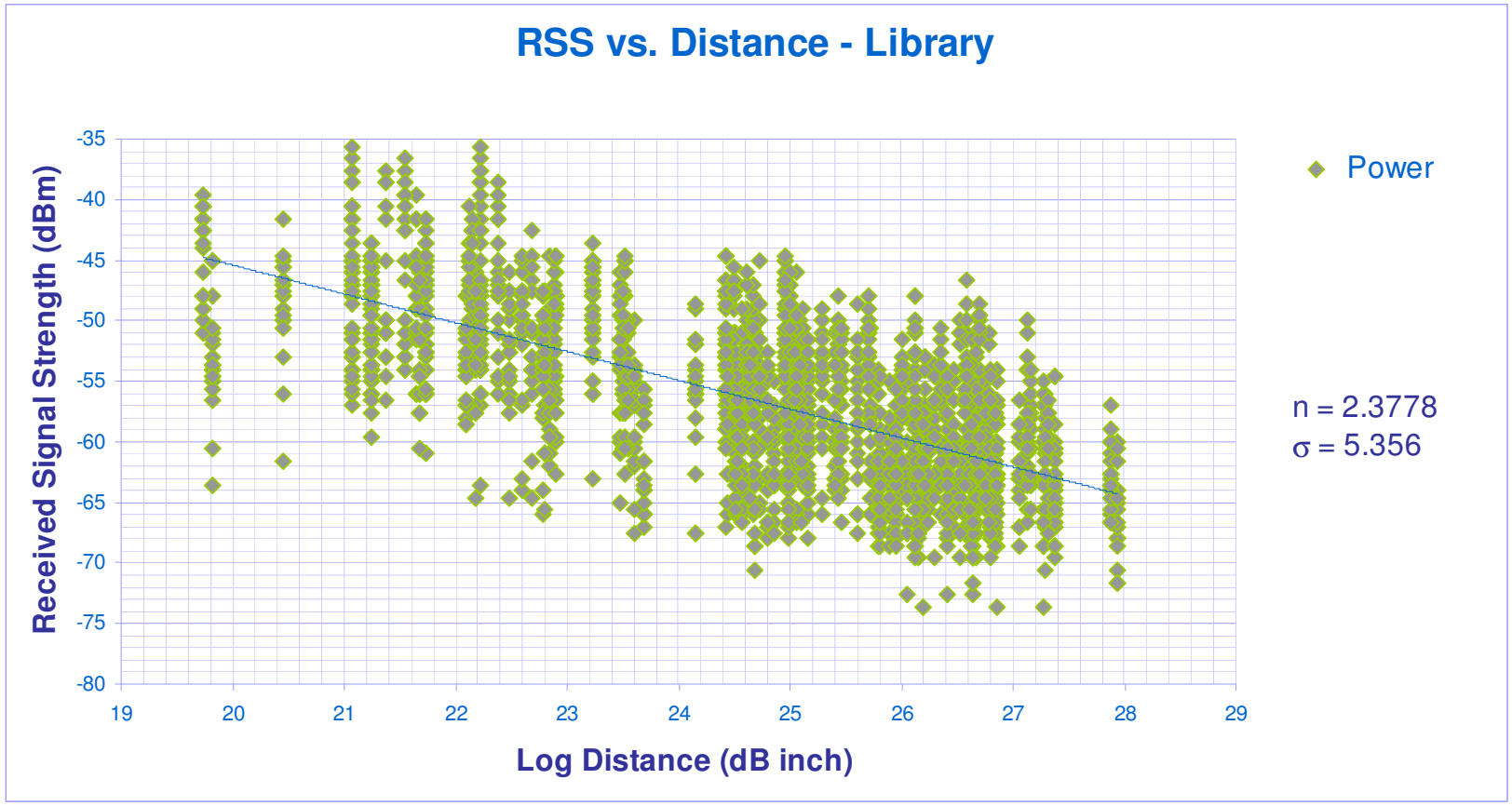

Fig. 10. The RSS measurements for the 2-D environment from laptop 1 and laptop 2. Each TX-RX connection has 20 measurements to calculate an accurate path loss exponent, *n*, and shadowing variance, σ^2.

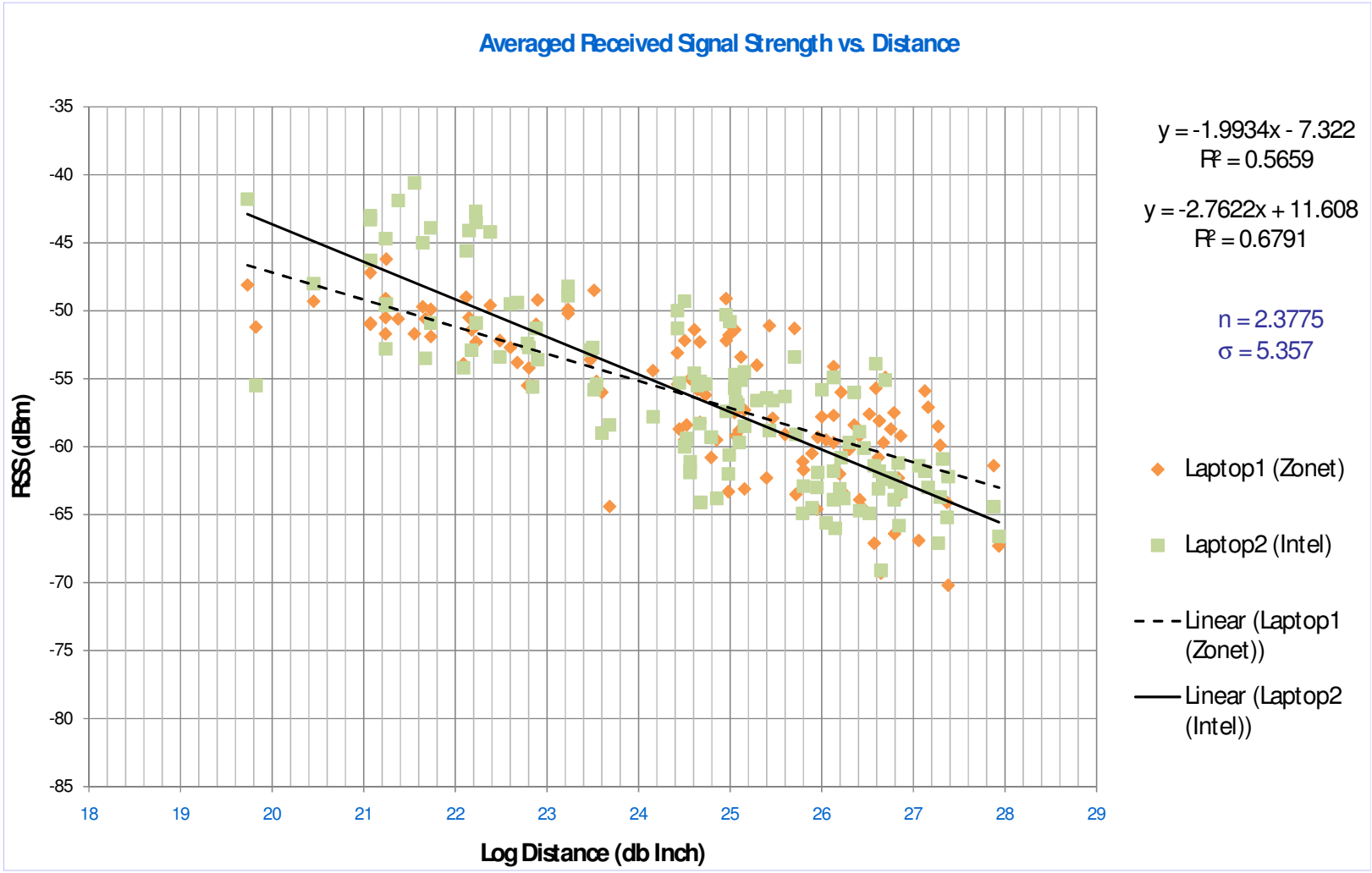

Fig. 11. The averaged RSS measurement for the 2-D environment. Both the path loss exponent, n, and shadowing variance, σ^2, will be averaged as well.

4. ANALYSIS AND CONCLUSIONS

4.1 Analysis

This section will present analysis and results comparing the two algorithms in both the 1-D and 2-D cases using real world power measurements. Since each algorithm has various tuning parameters, the analysis was conducted using common tuning variables used by the authors of each algorithm.

4.1.1 Path Loss Exponent and Shadowing Variance

To begin our analysis, both algorithms use range estimators to estimate distances based on RSS. The first step is to estimate an environmental parameter that the range estimators rely on, the path loss exponent (commonly symbolized as "*n*"). The path loss exponent is a general measure of how fast the signal strength is degrading over distance. The larger the value of *n*, the weaker the signal strength will likely be at a certain distance. In addition, if the range estimator wishes to be unbiased, then an additional environmental parameter called the shadowing variance (commonly symbolized as "σ^2" or "σ^2_{dB}") is needed. This nonnegative number is a measure of how much random fluctuation exists in the RSS. Recall that obstacles in the environment may cause the RSS to undergo random fluctuations. A larger value of σ^2 will result in a larger swing of power readings. If σ^2 was equal to zero, there would be no randomness and consecutive power readings would be constant at a given distance. Larger values of *n*

and smaller values of σ will tend to produce more accurate position estimations in the algorithms.

The RSS data was used to estimate both *n* and σ. Using a simple linear regression, a best fit line can be found from the data. The negative slope of this line is the path loss exponent. To estimate the shadowing variance, simply take the average of the squared errors between the observed data and the best fit line. Take the square root of the average to get σ. In the 2-D library environment, 20 RSS measurements were taken for each TX-RX pair, and the results show that *n* and σ were around 2.3 and 5.3 dB, respectively. In fact, a value of 2 and 5 for *n* and σ, were common across environments. Figures 9 –11 in section 3.3 have both *n* and σ labeled. The *n* and σ for Dr. Patwari's data was calculated to be 2.30 and 3.92 dB, respectively, and verified in [9, p. 2142].

Equipped with both the path loss exponent and shadowing variance, the range estimator can convert RSS into distances, which will be used in both algorithms.

4.1.2 Comparing with Typical Values

To start the comparison, typical values for both algorithms are used. For both algorithms, a small stopping criterion, ε, is used. With a small ε, the algorithms will tend to execute more iterations and will allow their RMSEs to converge to a final value. For the LC algorithm, an unbiased range estimator is used. For the dwMDS algorithm, a 2-stage random initialization approach is used. The range estimator is maximum likelihood, and the weights are assigned as exponentials using the same method in [1, p.

49]. For both algorithms, full connectivity is assumed, that is, every node is able to connect to every node, and every node will use all of its RSS measurements to estimate its own position. As will be seen later, the full connectivity assumption can be relaxed in a process known as "Neighborhood Selection."

Since dwMDS has a random initialization, the final estimation will likely be random as well. It could be that some initializations might provide a better estimation compared to LC than other initializations. Due to this, dwMDS will be run 100 times with the exact same parameters and data; however, each run will have a different random initialization. An averaging will be used to find a final RMSE for dwMDS. These repeated runs will also estimate the probability that the LC algorithm provides a better estimation than dwMDS. It is impractical to present 100 different graphs of possible dwMDS solutions, so instead, the results of the LC will be shown and the corresponding dwMDS average will be plotted alongside. If the reader is curious to see how a sample dwMDS solution compares with the LC across iterations, several example plots (Figures 30-32) are provided in the Appendix. First, the 1-D results will be presented in Figures 12-14, followed by 2-D results in Figures 15-17, and concluded by Table 1 to summarize the results.

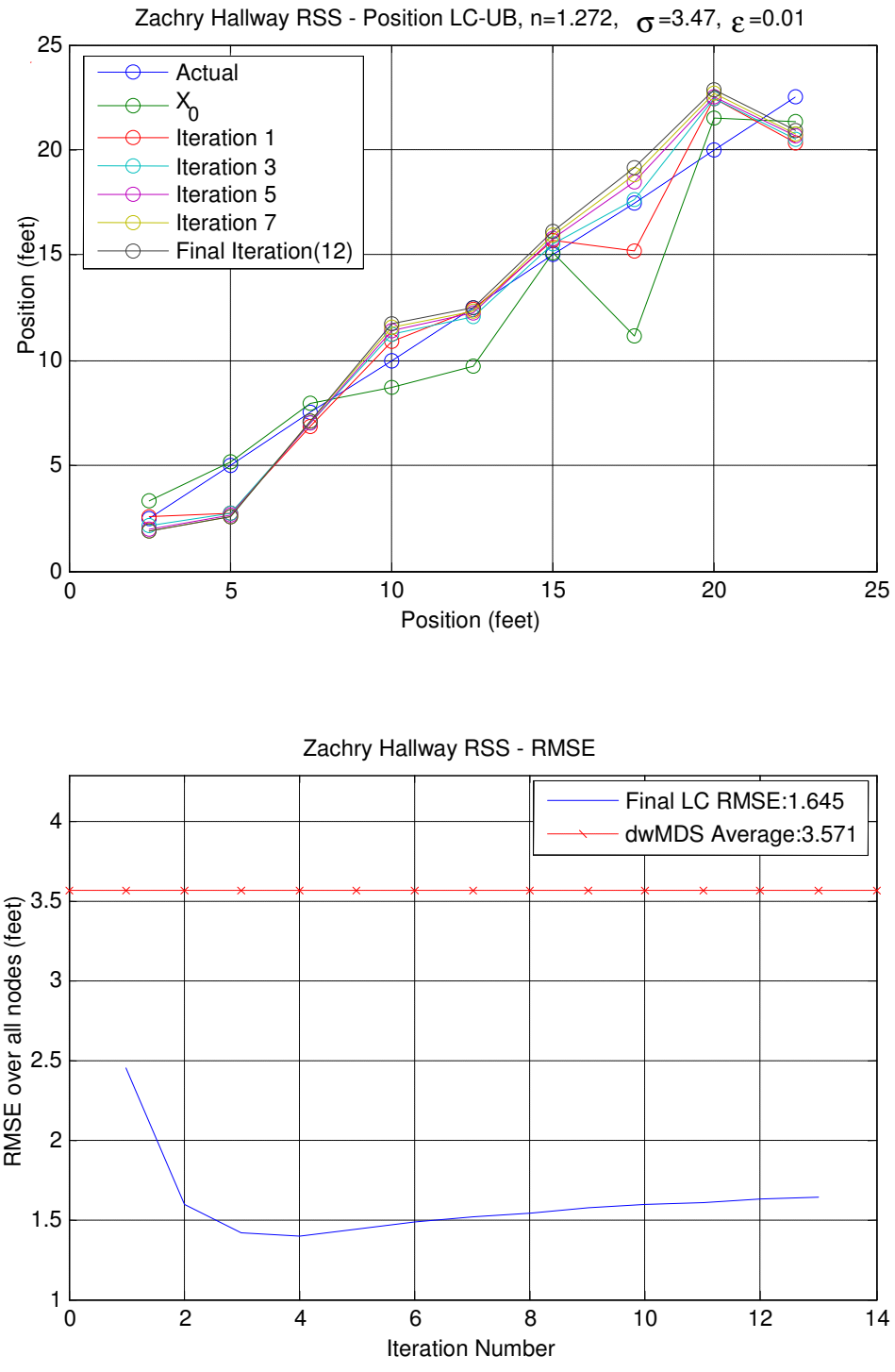

Fig. 12. 1-D hallway environment: estimations can be seen across several iterations (top), and the corresponding RMSE (bottom).

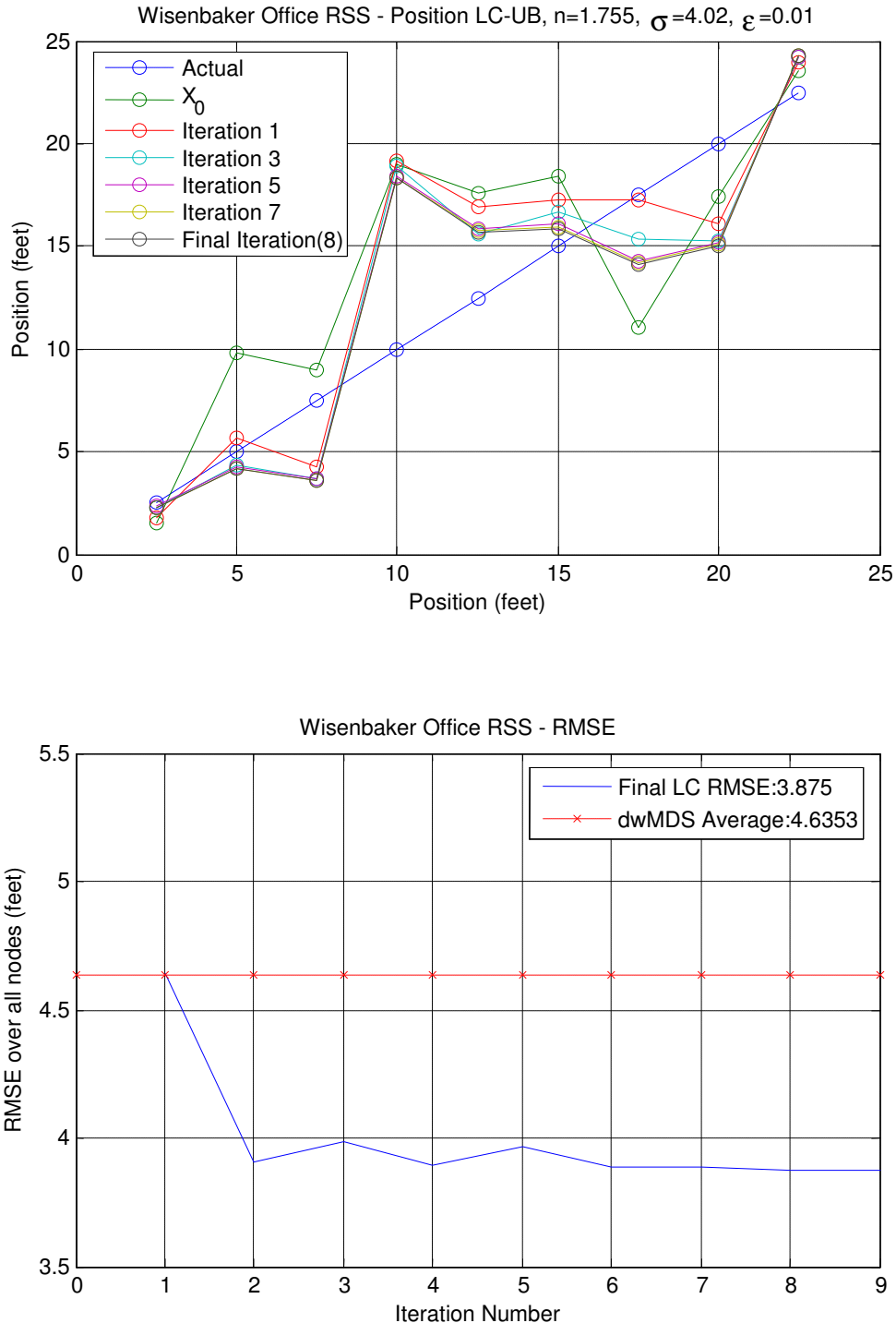

Fig. 13. The 1-D results of office environment.

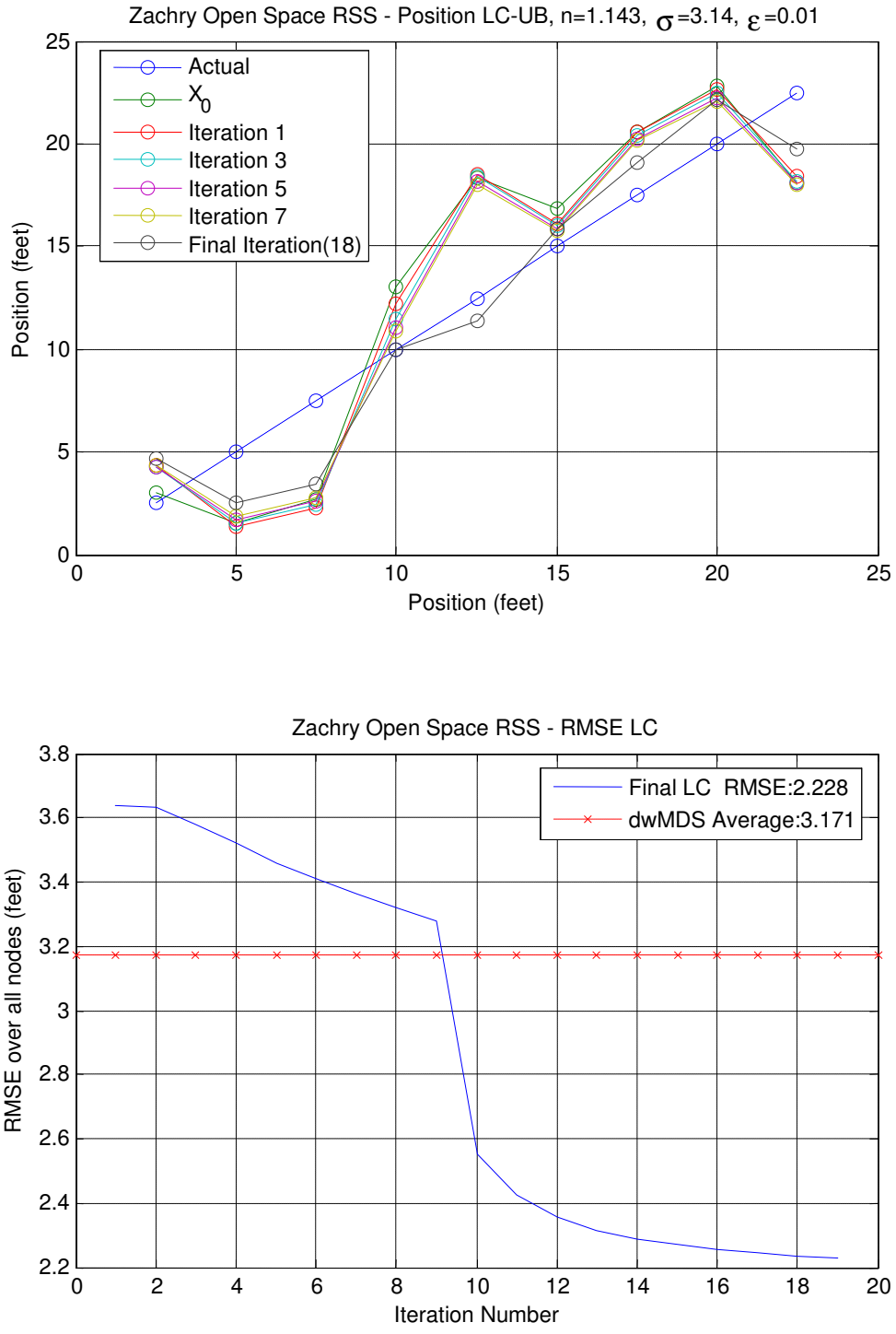

Fig. 14. 1-D results of open space environment.

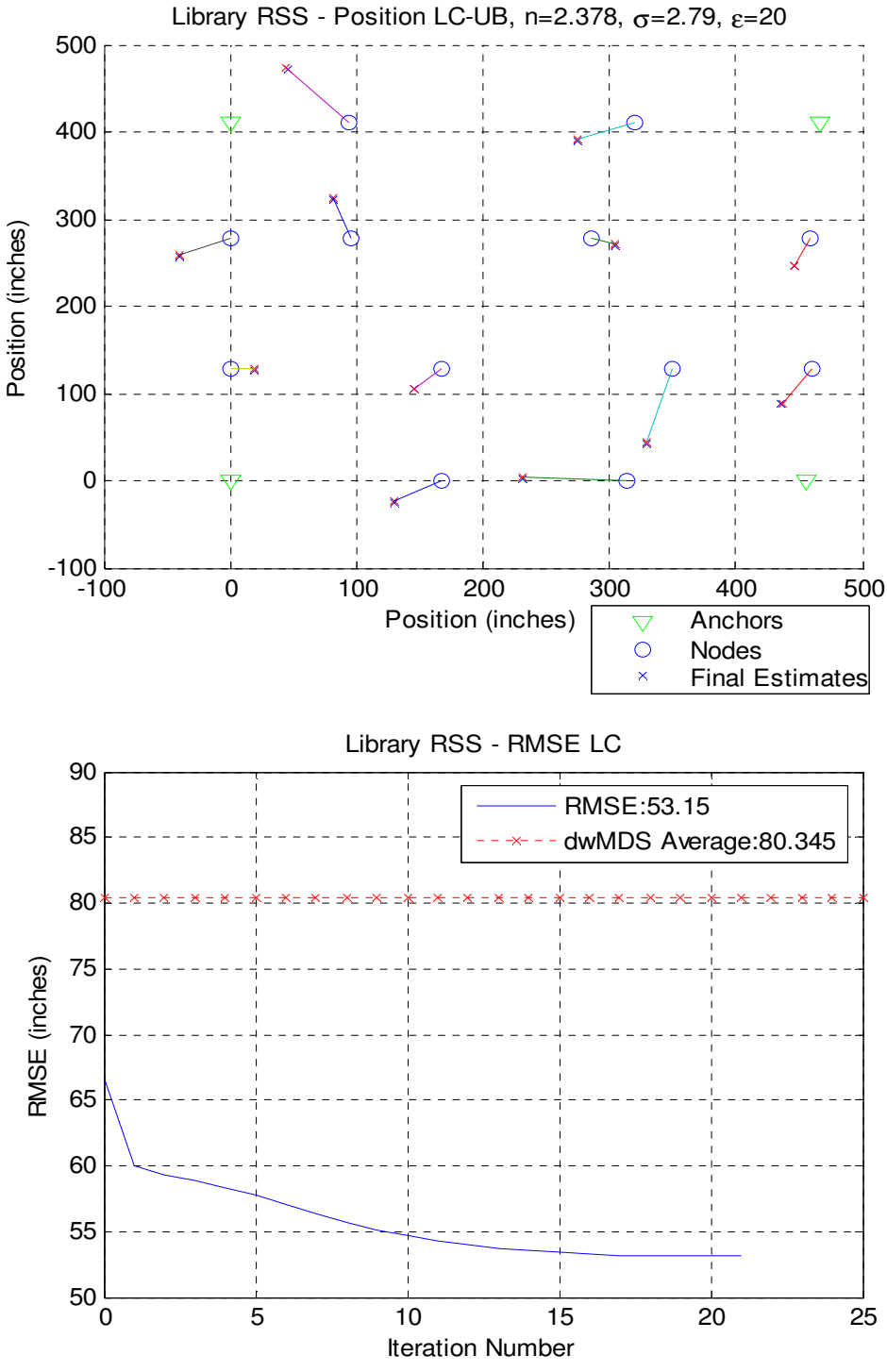

Fig. 15. 2-D Library (shelving) environment: final estimation is shown (top) and the corresponding RMSE (bottom).

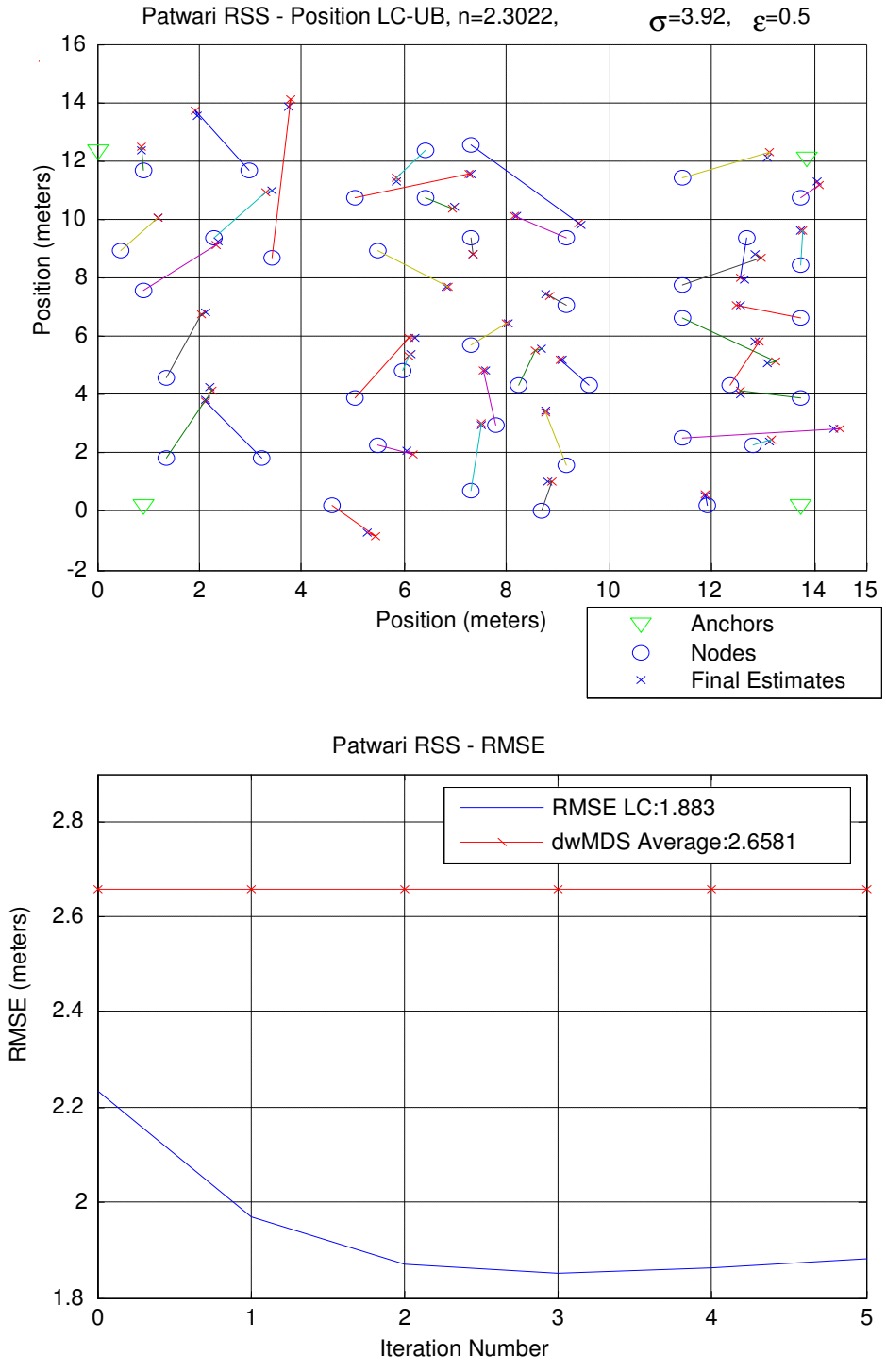

Fig. 16. 2-D results of cubicle environment. Data courtesy of Dr. Neal Patwari [7].

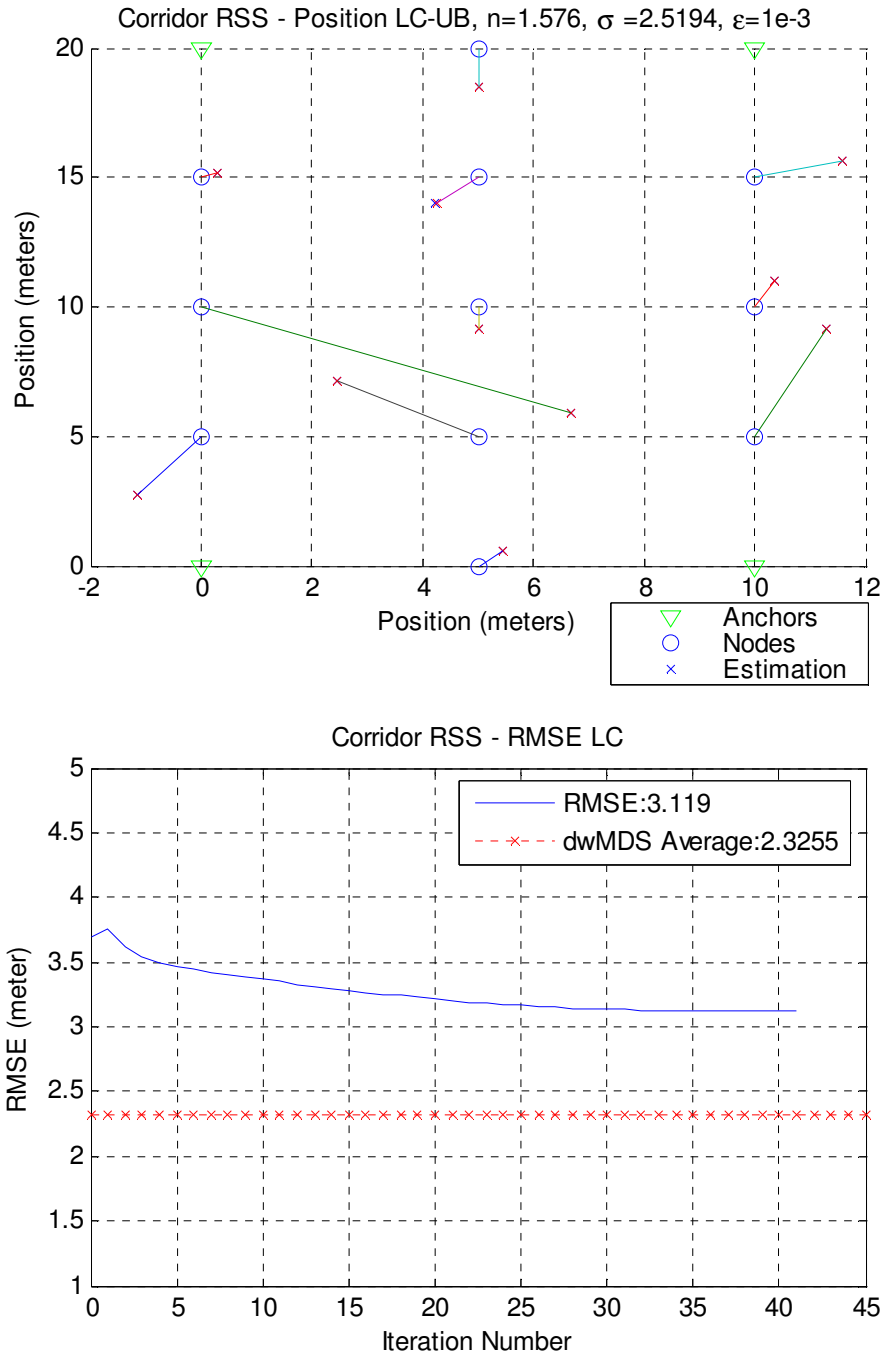

Fig. 17. 2-D results of corridor environment. Data courtesy of Rajukumar Samuel. The raw data can be obtained from Table 8.

Table 1

RMSE and probability results for the 1-D and 2-D environments.

1-D Environments	LC RMSE (feet)	Average dwMDS RMSE (feet)	Probability that LC had lower RMSE
Zachry Hallway	1.645	3.571	1
WERC Offices	3.875	4.63	.66
Zachry Open Space	2.228	3.17	1
2-D Environments	LC RMSE	Average dwMDS RMSE	Probability that LC had lower RMSE
Library (shelving)	53.15 in.	80.34 in.	1
Cubicle	1.883 m	2.6581 m	1
Corridor	3.119 m	2.3255 m	0

The dwMDS algorithm was allowed to run for many iterations. It could be that the last iteration had the lowest RMSE, or somewhere midway through the iterations, the lowest RMSE was reached. Regardless, the "Average dwMDS RMSE" was computed by taking the minimum RMSE (whether it was the last iteration or somewhere prior to that) from each of the 100 realizations. Despite giving dwMDS this advantage, the LC algorithm outperformed dwMDS a majority of the time in all but one environment, the 2-D Corridor. Recall that typical settings were used for both algorithms. The next sections will look into adjusting some of these settings.

4.1.3 Symmetric vs. Asymmetric

Both algorithms rely on a "power matrix". Recall that a power matrix is the 2-D matrix that holds the RSS values for each TX-RX pair. Reciprocity is the assumption that the RSS of the TX and RX are equal if the TX and RX swapped roles. So for every TX-RX combination, there is only one RSS measurement. If reciprocity is assumed, the power matrix will be symmetric. In fact, in each of the developed algorithms [1][2], the authors use symmetric power matrices as their examples to demonstrate their performance. However, in the real world, reciprocity might not be observed, and, in fact, during the measurement campaign, there were different RSS values when TX and RX reversed roles.

In this section, a comparison between symmetric and asymmetric power matrices is presented and how it affects the position estimation. By default, asymmetric power matrices were recorded from the measurement campaign, however, a symmetric version (used in section 4.1.2) was obtained by simply averaging the RSS readings for each TX-RX pair. Since the 2-D library environment had 20 readings for each TX-RX pair, more averaging was used here. Below are the results for both 1-D (Figures 18-20) and 2-D environment (Figure 21) when symmetry is disabled. The algorithms were run with nearly identical parameters as the figures presented in 4.1.2. Table 2 provides a summary of the results.

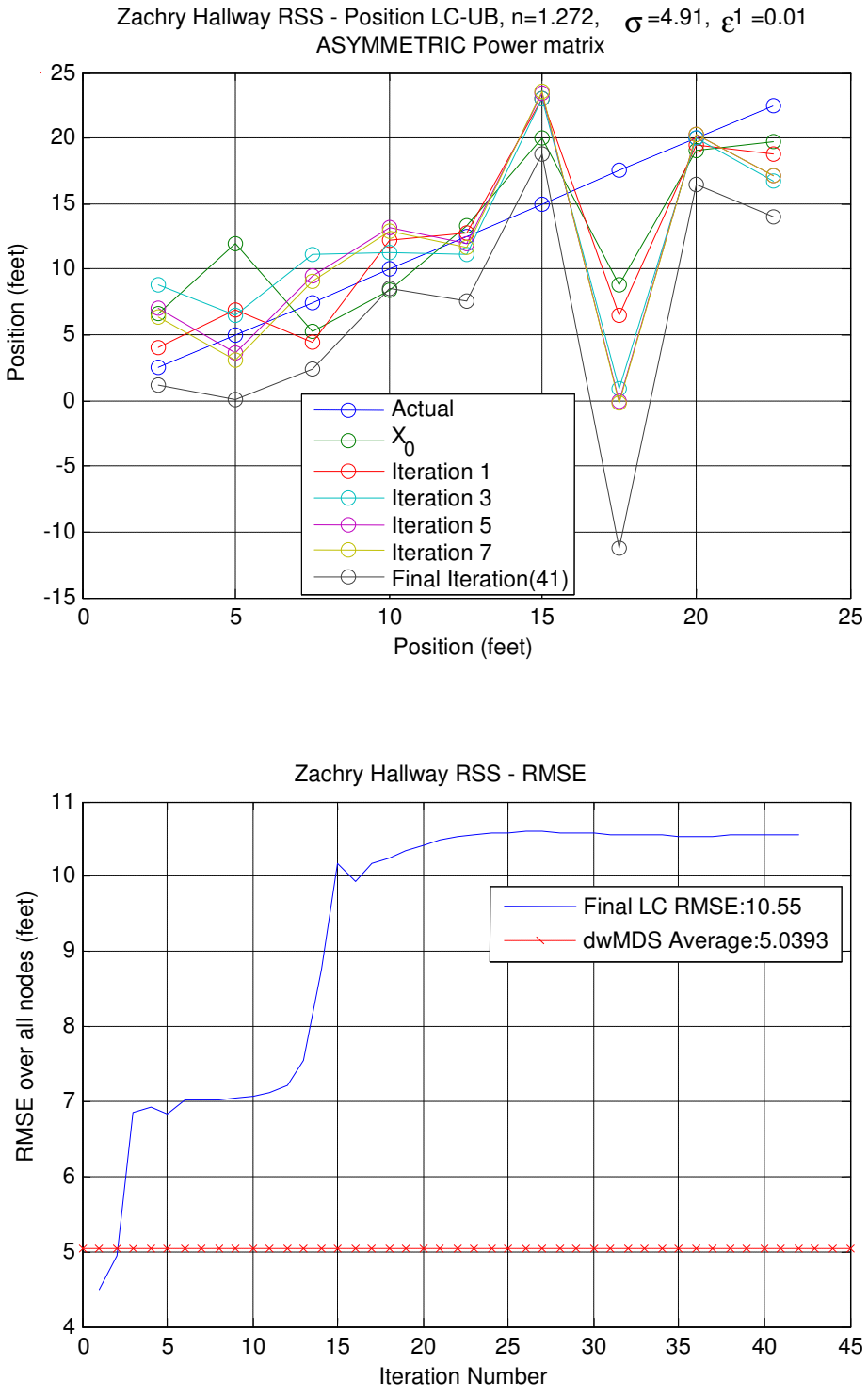

Fig. 18. 1-D hallway environment with an asymmetric power matrix: estimations can be seen across several iterations (top), and the corresponding RMSE (bottom).

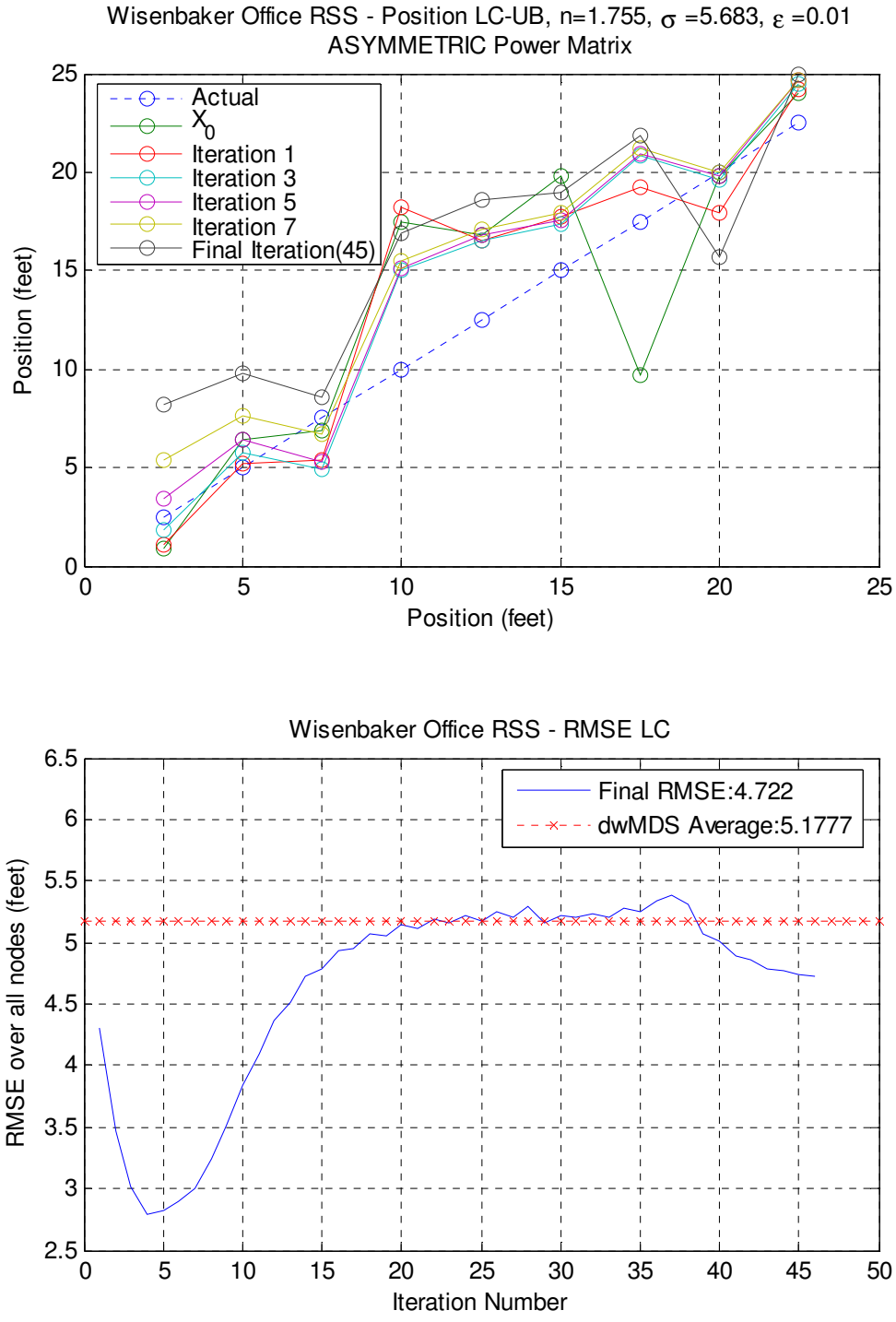

Fig. 19. 1-D results for office environment with asymmetric power matrix.

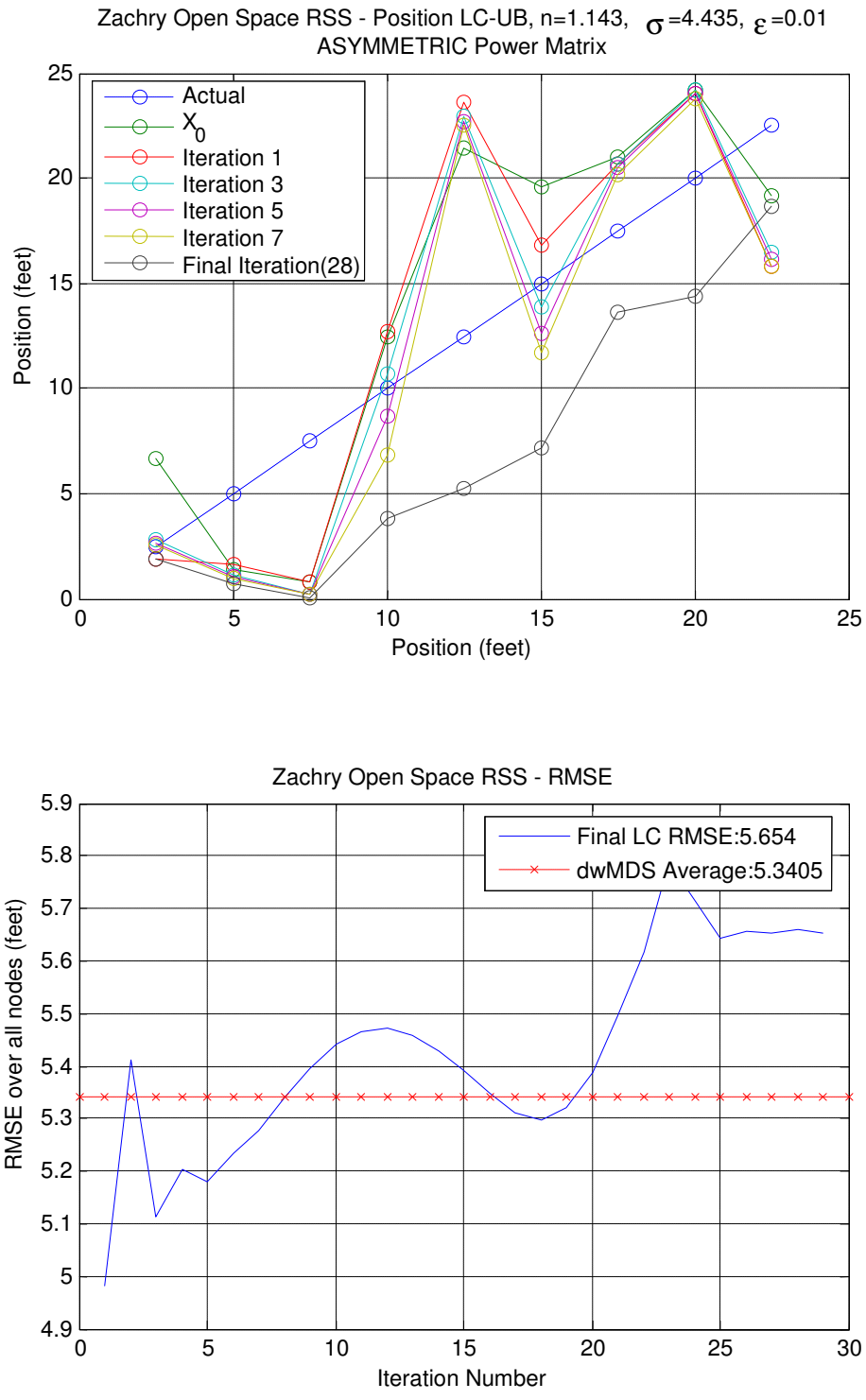

Fig. 20. 1-D results for open space environment with asymmetric power matrix.

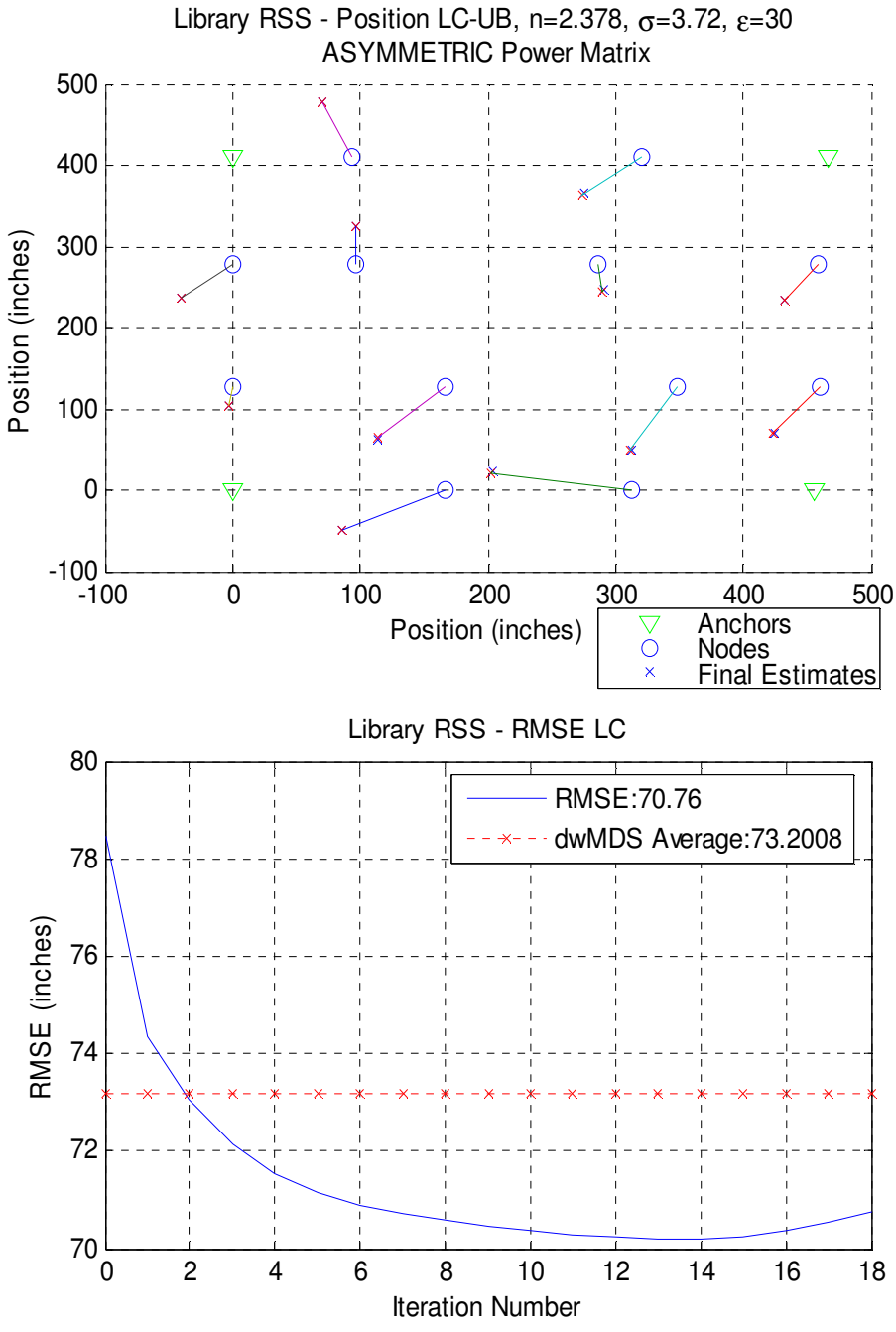

Fig. 21. 2-D results for library (shelving) environment with asymmetric power matrix.

Table 2

RMSE and probability results for both 1-D and 2-D using asymmetric power matrices.

1-D Environments ASYMMETRIC	LC RMSE (feet)	Average dwMDS RMSE (feet)	Probability that LC had lower RMSE
Zachry Hallway	10.55	5.04	.04
WERC Offices	4.72	5.17	.23
Zachry Open Space	5.654	5.34	0
2-D Environments ASYMMETRIC	LC RMSE	Average dwMDS RMSE	Probability that LC had lower RMSE
Library (shelving)	70.76 in.	73.20 in.	.69

Clearly in the 1-D case, both algorithms have degraded performance by not having a symmetric power matrix. In the 2-D case, comparisons can only be made in the library environment since the other two sets of data are, by default, symmetric. The 2-D case doesn't show as much of a degradation in performance, and, in fact, dwMDS seems to have improved slightly. Despite the improvement, dwMDS only showed better results 31% of the time. It should be worth noting that by averaging RSS measurements, the variance begins to decrease. This is the reason the variance is slightly different than the figures in 4.1.2.

4.1.4 Full Connectivity vs. Neighbor Selection

A final comparison looks at how removing the assumption of full connectivity can change the position estimation (by either improving or degrading). Recall that full connectivity is the scenario that each blindfolded node can connect to every other node, and all the connections are used to estimate location. Without full connectivity, the blindfolded nodes must select "neighbors" to use in the algorithm. The idea of not being able to fully connect to all nodes (including anchors) seems more realistic.

The creators of dwMDS proposed a 2-stage algorithm which includes an adaptive neighborhood selection process [1, p. 55]. However, at the time of this writing, the authors of the LC algorithm do not have a formal neighbor-selecting process yet. One needs to be created in order to fairly compare each algorithm. One solution was proposed in which initial neighbors are chosen via a power threshold. Consulting the power matrix, neighbors can be selected based upon meeting or exceeding the power threshold. However, if this list of neighbors remains static throughout the algorithms' iterations, a bias remains and position estimates are slowly pulled toward the center as seen in Figure 22. For more information on the bias and why it exists, please consult [1, p. 54].

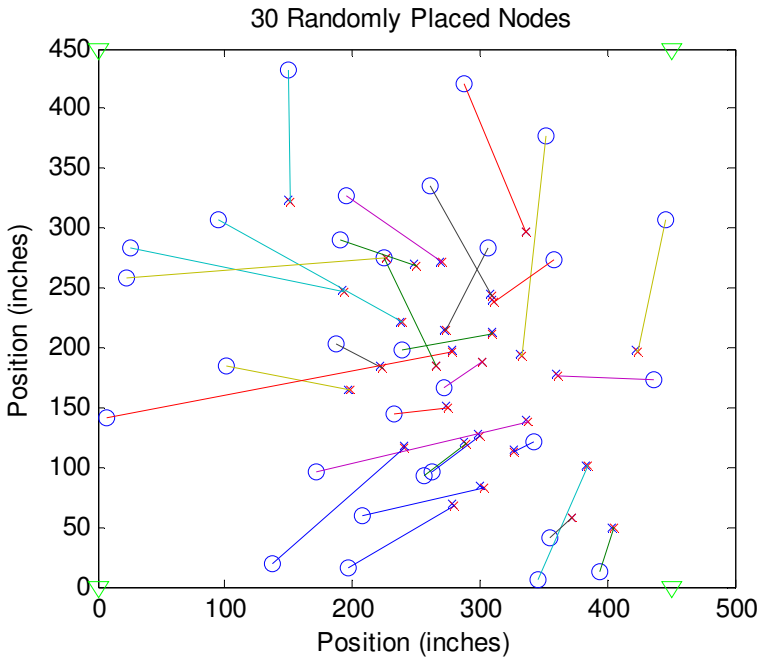

Fig. 22. A bias remains when neighbor selection is static. The position estimates are pulled towards the center due to the bias.

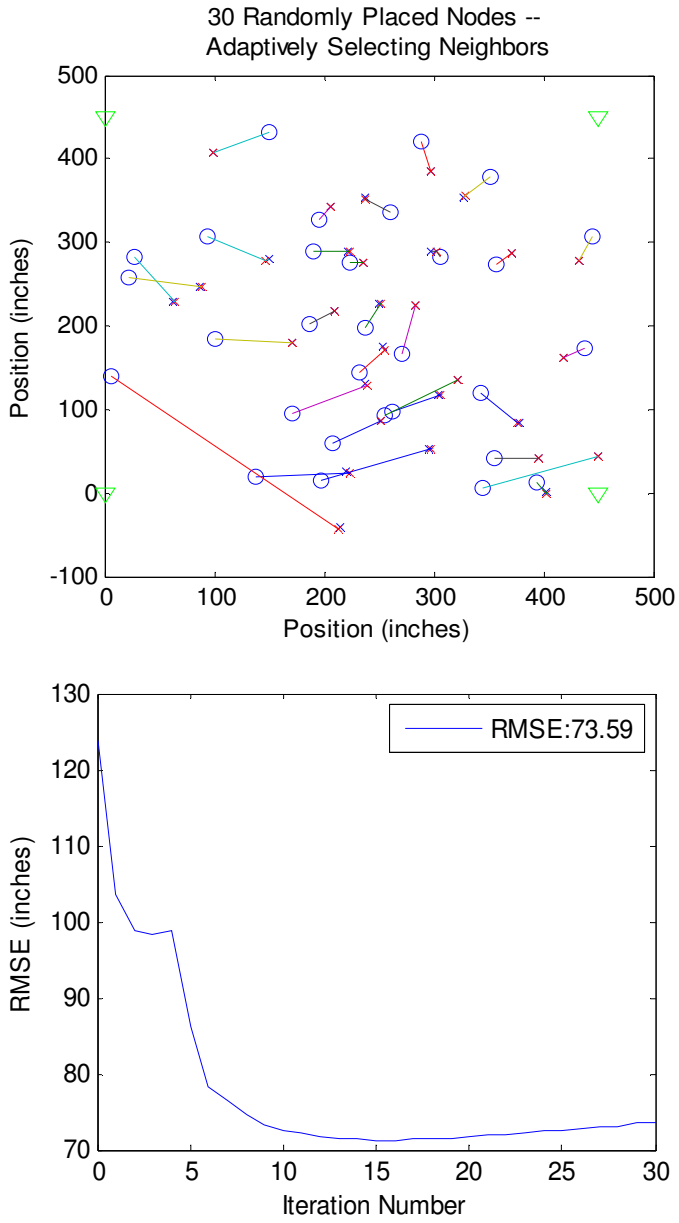

Fig. 23. Adaptive neighbor selection removes the biasing. (Top) The neighbors are static (based on power matrix) for the first five iterations, and are adaptively selected after that. (Bottom) The RMSE gets a "boost" from adaptive selection.

To overcome this bias, neighbors must be selected adaptively or, in other words, dynamically. A revised neighbor selection process was developed where initial

neighbors are chosen based on the power matrix satisfying the power threshold, however after a few iterations, neighbors are adaptively chosen each successive iteration based on current position estimate. Recall that the range estimator allows conversions from power to distance, so a power threshold can be thought of also as a distance threshold. The neighbors are adaptively chosen (each iteration) by comparing the distances between all the estimates against the distance threshold. Figure 23 shows the same nodes and data as Figure 22, but with adaptive neighbor selection after five iterations.

Another consideration is whether anchors should be subject to neighbor selection. It has been assumed up to this point that the anchors are most likely stationary XCVRs and every node can communicate to all anchors, however, in the real world, this may not be true. Therefore, anchors will be subject to neighbor selection. This brings about another problem of initialization. In the LC algorithm, each node obtains an initial position estimate from all anchors, however, if by chance, a node can not connect to *any* anchor, where should we place the initial estimate? It seems that if a node can not connect to any anchor, it is most likely in the center of the test area, and based on that, these initial estimates are placed in the center of the square (or line if 1-D). Other scenarios such as placing at the center with some randomness were investigated, but didn't seem to provide any more improvement than simply placing the initial estimate directly at center.

Finally, one last consideration is the possibility of two nodes not being neighbors at first, but after several iterations, their estimates grow closer together and the nodes are selected as neighbors. From a real world perspective, it could be that the nodes are not neighbors at first simply because the nodes physically can not communicate. It doesn't make much sense that after a few iterations, the nodes become neighbors even though they physically can not communicate. To mimic the real world, a second threshold was created called a "connection distance." In order for two nodes to be neighbors, they must satisfy both a neighbor distance *and* a connection distance. The connection distance should always be greater than or equal to the neighbor distance. One can think of the neighbors as the strongest signals from the pool of connections.

With the above proposed adaptive neighborhood selection, we can now look at how the LC algorithm compares with dwMDS when full connectivity is not assumed. The test setup and algorithm parameters are identical to section 4.1.2 (i.e. typical values, symmetric power matrices, averaging, etc.) except in this test, the nodes are allowed to choose neighbors and which connections will be used to estimate their positions. Figures 24-26 provide the 1-D results and Figures 27-29 provide the 2-D results. Table 3 summarizes the results for neighborhood selection.

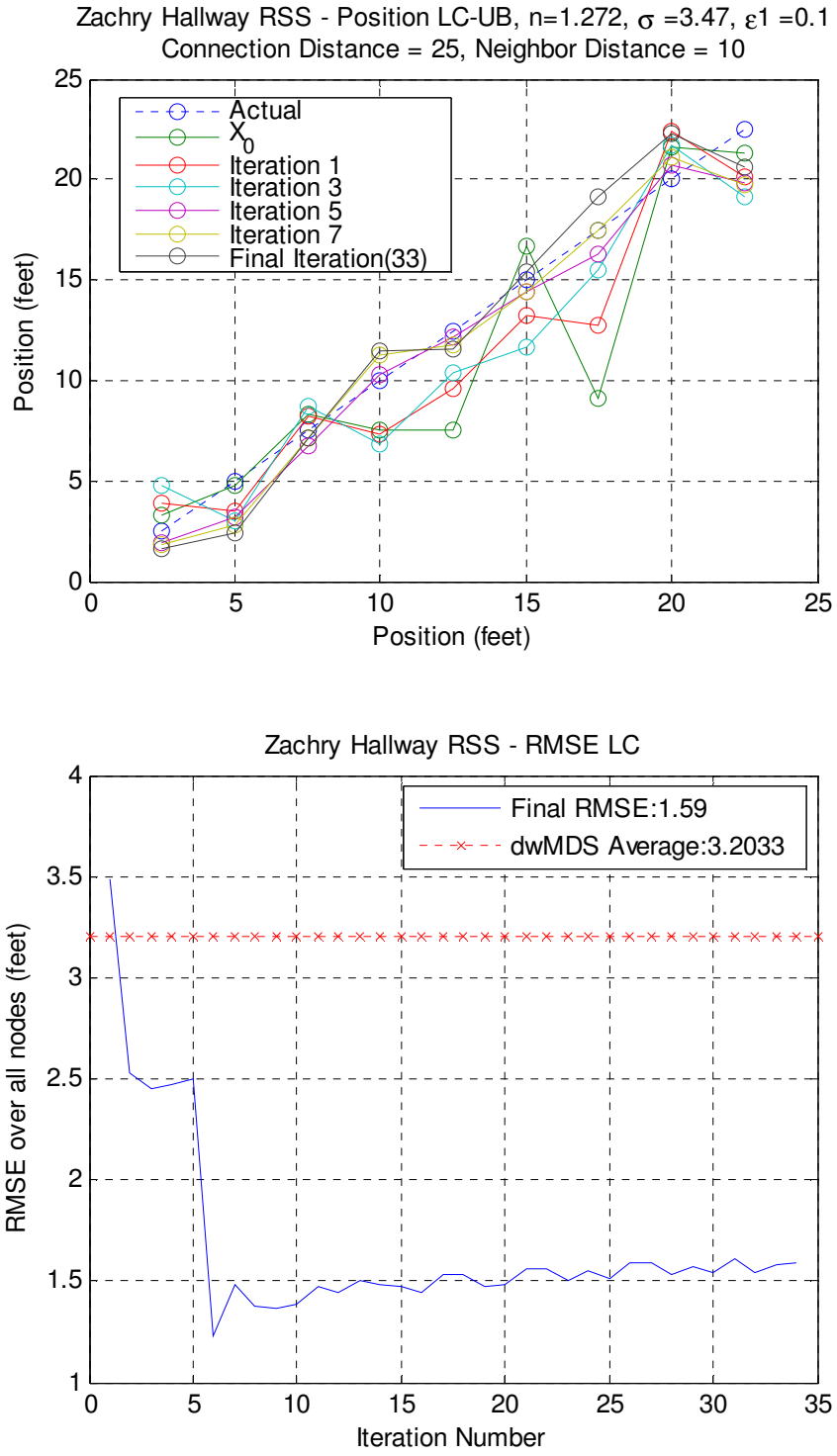

Fig. 24. 1-D hallway environment with neighbors selected: estimations can be seen across several iterations (top), and the corresponding RMSE (bottom).

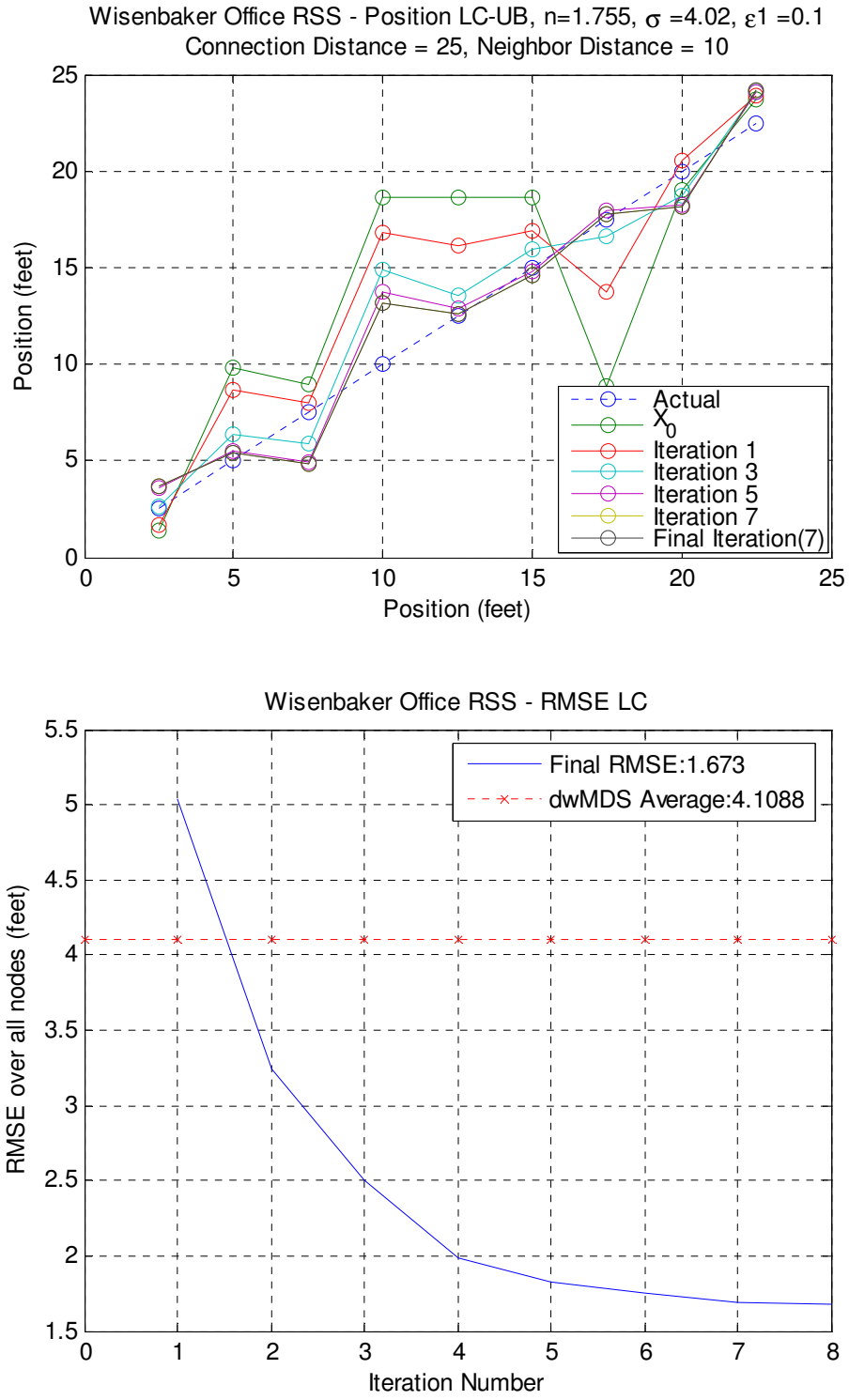

Fig. 25. 1-D office environment with neighbors selected.

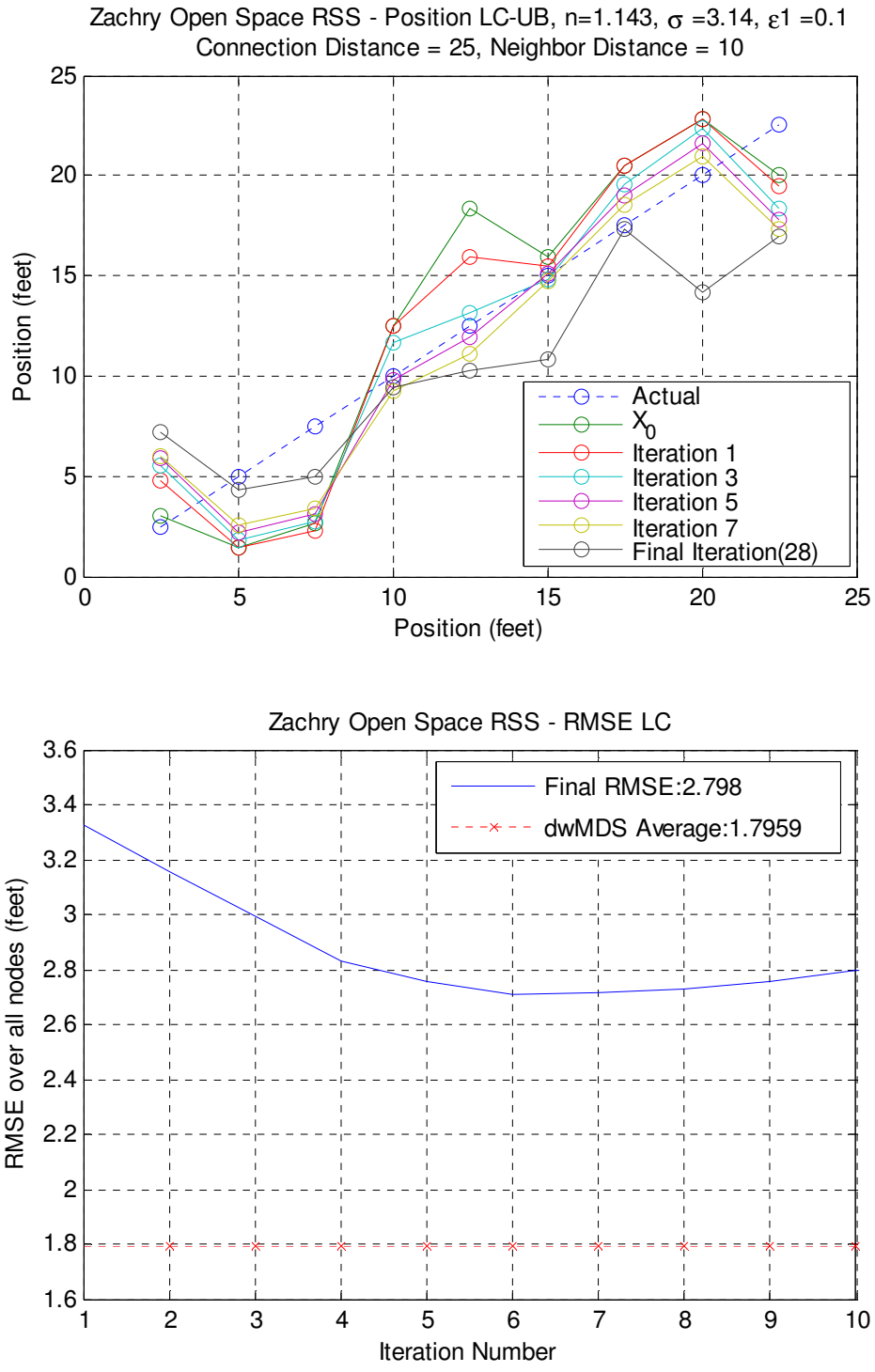

Fig. 26. 1-D open space environment with neighbors selected.

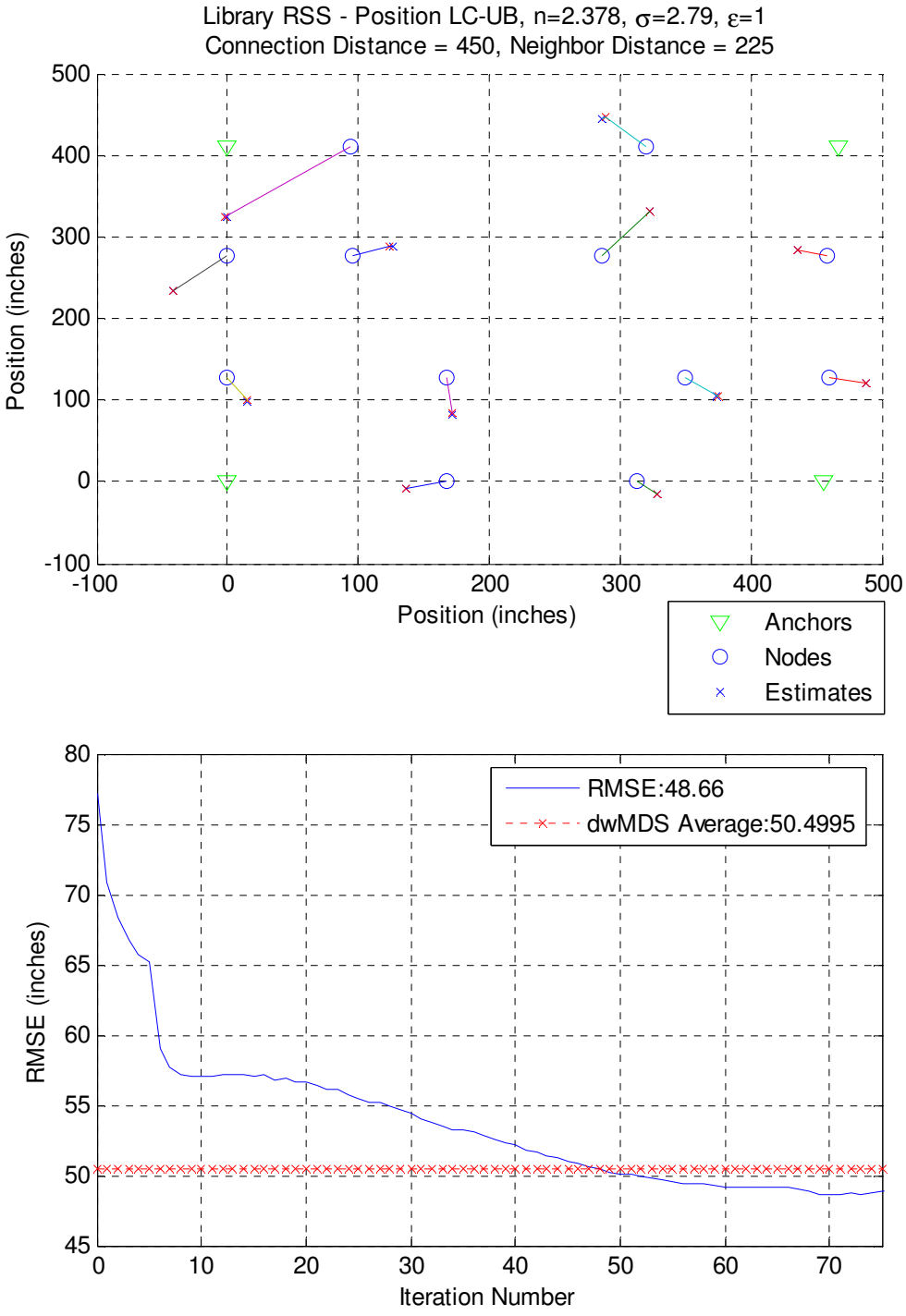

Fig. 27. 2-D Library (shelving) environment with neighbor selection: final estimation is shown (top) and the corresponding RMSE (bottom).

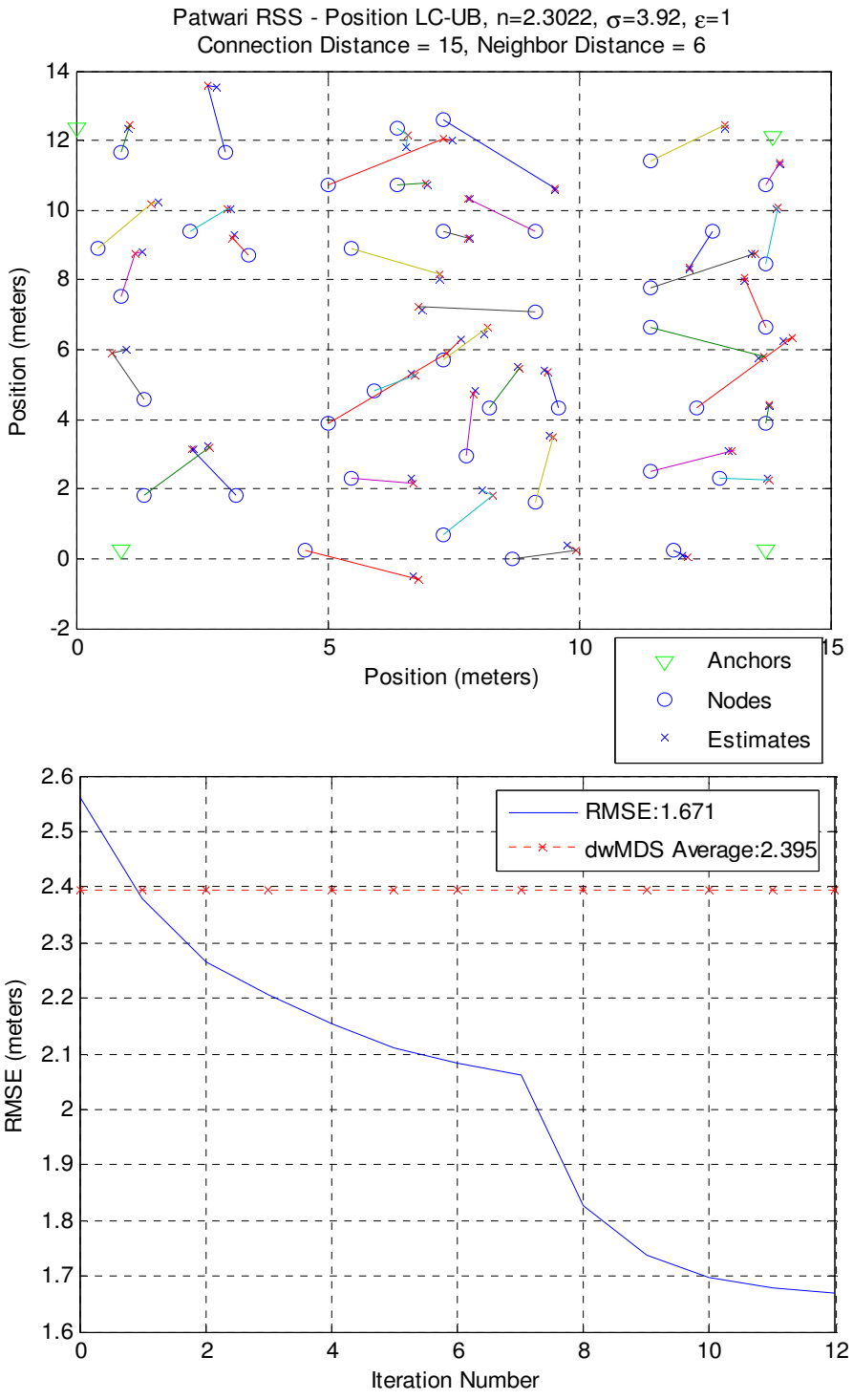

Fig. 28. 2-D results of cubicle environment with neighbor selection. Data courtesy of Dr. Neal Patwari [7].

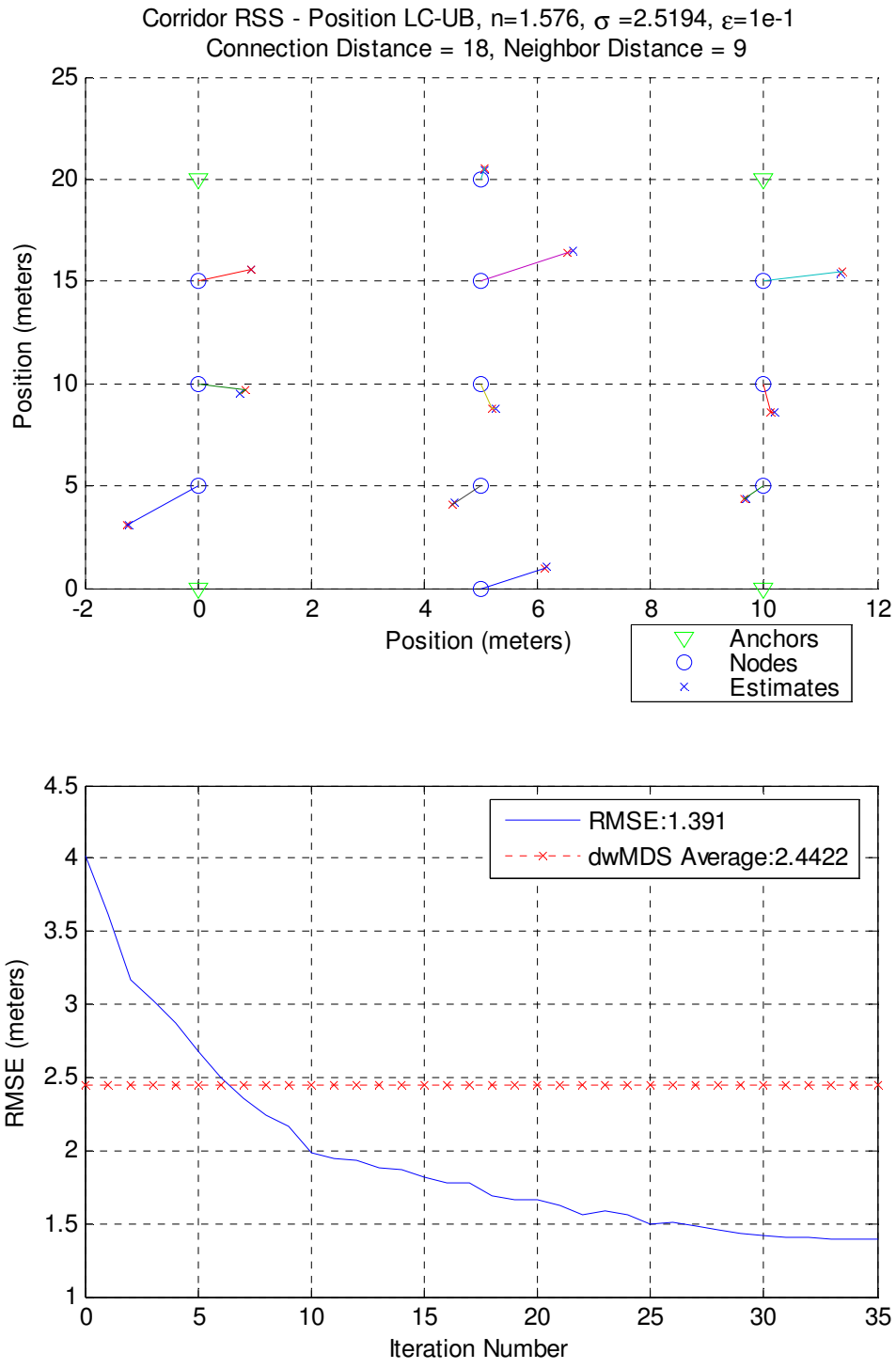

Fig. 29. 2-D results of corridor environment with neighbor selection. Data courtesy of Rajukumar Samuel. The raw data can be obtained from Table 8.

Table 3

RMSE and probability results for the 1-D and 2-D environments with neighbor selection.

1-D Environments (Neighbors Selected)	Min. LC RMSE (feet)	Average Min. dwMDS RMSE (feet)	Probability that LC had lower RMSE
Zachry Hallway	1.2326	3.2033	.69
WERC Offices	1.6728	4.1088	1
Zachry Open Space	2.7065	1.7959	.09
2-D Environments (Neighbors Selected)	**Min. LC RMSE**	**Average Min. dwMDS RMSE**	**Probability that LC had lower RMSE**
Library (shelving)	48.66 in.	50.5 in.	.58
Cubicle	1.671 m.	2.395 m.	1
Corridor	1.391 m.	2.442 m.	.99

Compared to Table 1, there is a slight improvement in RMSE when nodes are able to select neighbors. For Zachry Open Space, dwMDS saw a dramatic improvement due to neighbor selection (from 3.17ft. to 1.79ft.). Likewise, for the 2-D Corridor, LC saw a dramatic improvement as well (from 3.119m. to 1.391m.). It is quite evident that neighbor selection can improve the final estimates of these algorithms. Despite the improvements, however, dwMDS still seems unable (on average) to perform better than LC in most of the environments.

4.2 Conclusions

The focus of this research was to investigate the performance of two position locating algorithms in a wireless environment. These two algorithms rely solely on an

easily obtainable source of information, received signal strength (RSS). Rather than using computer simulations to test these algorithms, real world power measurements in the field were collected and used as the input data. The two algorithms tested were a linear combination (LC) algorithm [2] and the distributed weighted-multidimensional scaling (dwMDS) [1] algorithm. Since the same data was used for each algorithm, this provides a fair comparison and does not give any advantage to any particular algorithm. In addition, since the data is taken directly from the field, this provides more confidence of a successful real world implementation of these algorithms.

Results showed that in most environments with typical settings, the LC algorithm provides an estimate that is closer to the true location compared to dwMDS for both the 1-D and 2-D scenario. As seen in 4.1.3, the algorithms seem to perform the best when the power matrices are symmetric. Finally, when the assumption of full connectivity is removed and a more realistic system is used (not every node is able to connect to all other nodes), all estimates seem to improve and again the LC algorithm performed better on average in most environments compared to dwMDS.

REFERENCES

[1] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. on Sensor Networks*, vol. 2, no. 1, pp. 39-64, February 2006.

[2] W. Chen, S. Miller, "Distributed Linear Combination Estimators for Localization Based on Received Signal Strength in Wireless Networks," technical document, Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, 2008.

[3] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., Prentice Hall PTR, Upper Saddle River, NJ, 2002.

[4] D-Link, "D-Link *Air* DI-514 2.4 GHz Wireless Router Manual," ftp://ftp.dlink.com/ Gateway/di514_revB/Manual/DI-514_manual_07062005.zip (current 10 Jul. 2008).

[5] "Model: ZEW1501 Wireless CardBus Adapter," technical document, Zonet USA Corporation, Walnut, CA, 2008.

[6] "Passmark WirelessMon – Wireless 802.11 WiFi Monitoring Software," Passmark Software Pty Ltd, http://www.passmark.com/products/wirelessmonitor.htm (current 17 Jul. 2008).

[7] N. Patwari, "Wireless Sensor Network Localization Measurement Repository," http://www.eecs.umich.edu/~hero/localize/ (current 17 Jul. 2008).

[8] "Intel PRO/Wireless Network Connection for Mobile – Technical Documents," technical document, Intel Corporation, http://support.intel.com/network/connectivity/products/wireless/tech_docs_mobile.htm (current 17 Jul. 2008).

[9] N. Patwari, A. O. Hero III, et al., "Relative Location Estimation in Wireless Sensor Networks," *IEEE Trans. on Signal Processing*. vol. 51, no. 8, August 2003.

APPENDIX A

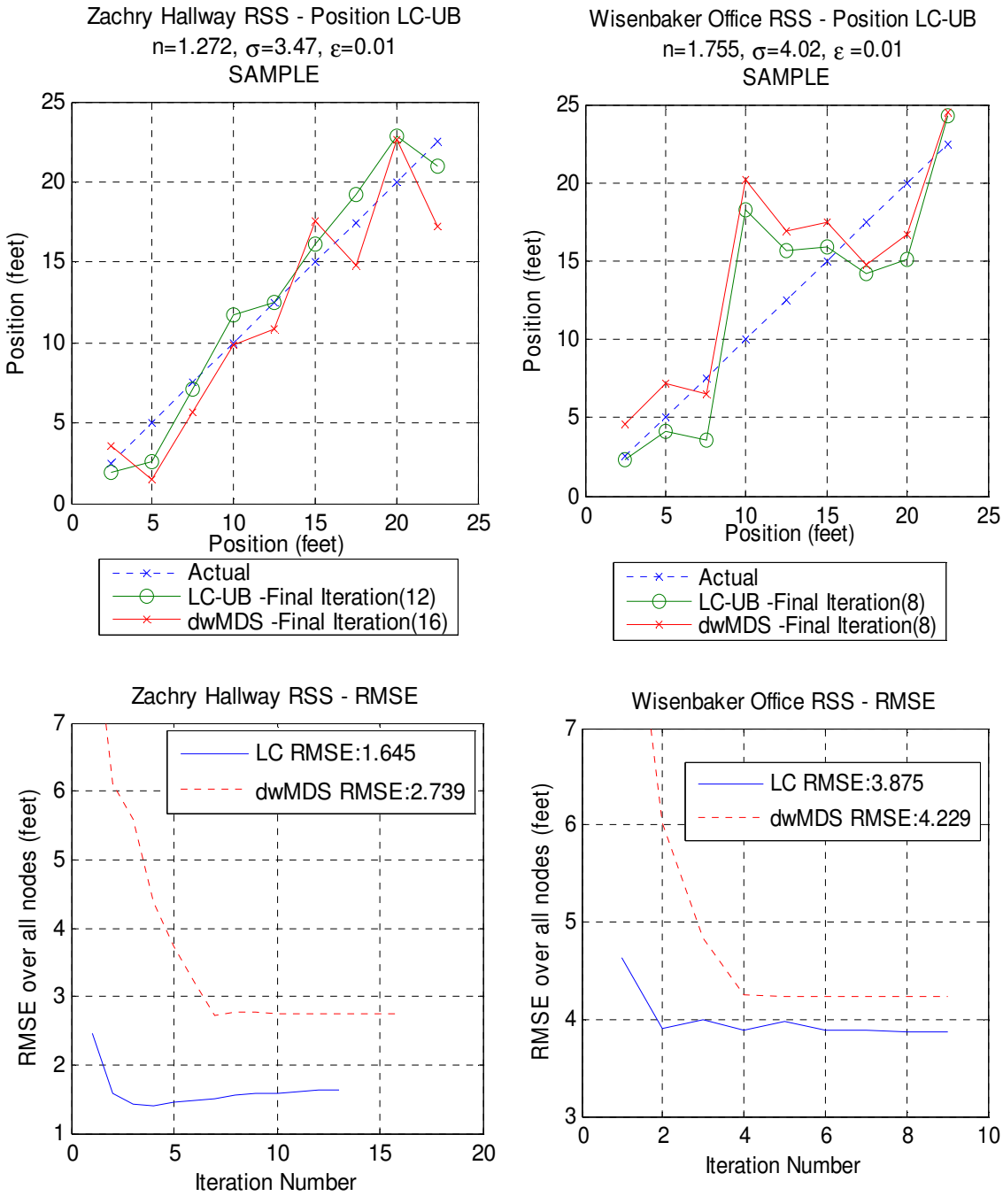

Fig. 30. Sample dwMDS estimates for the 1-D hallway and office (top). The corresponding RMSE for each environment (bottom).

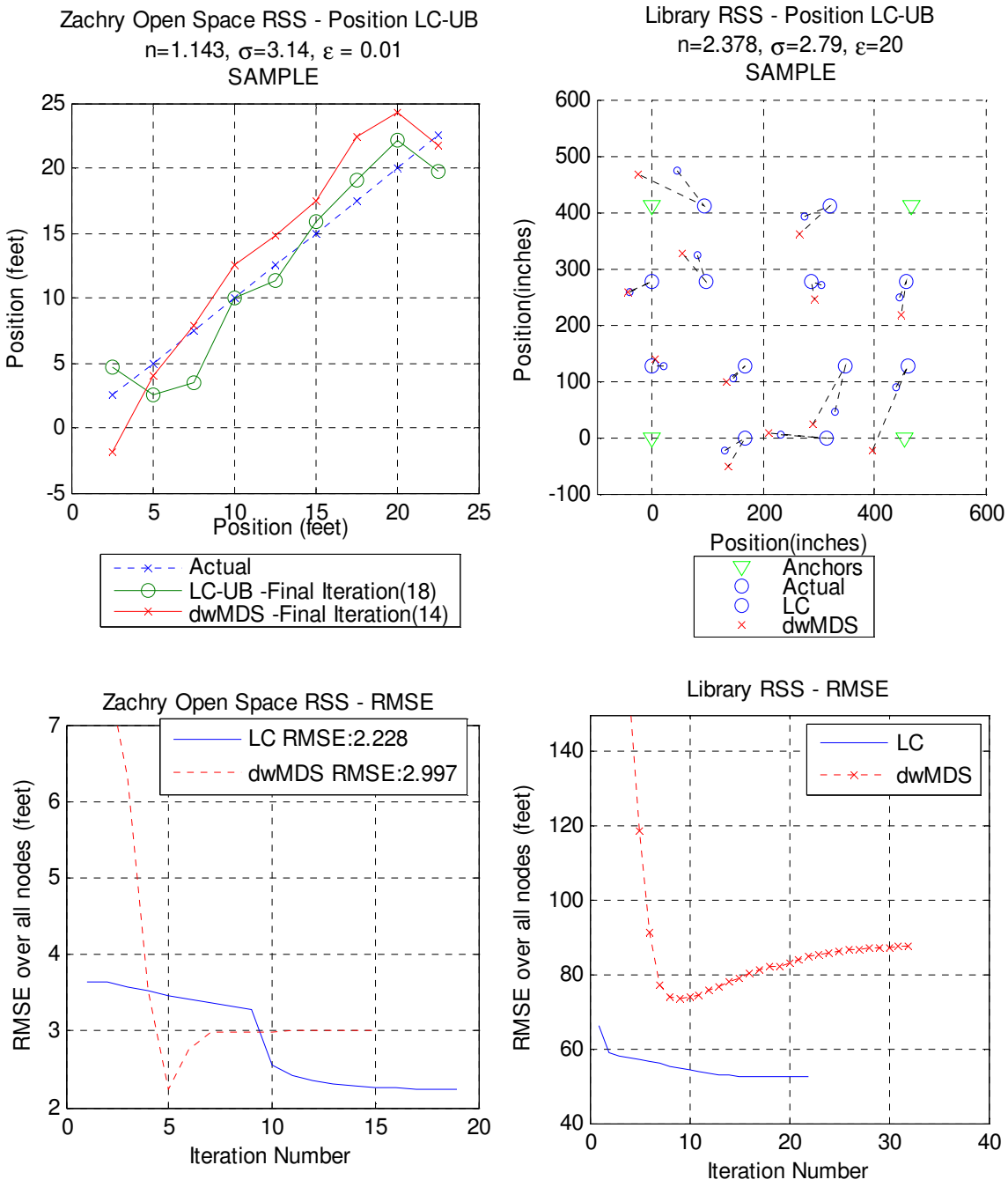

Fig. 31. Sample dwMDS estimates for the 1-D open space and 2-D library (top). The corresponding RMSE for each environment (bottom).

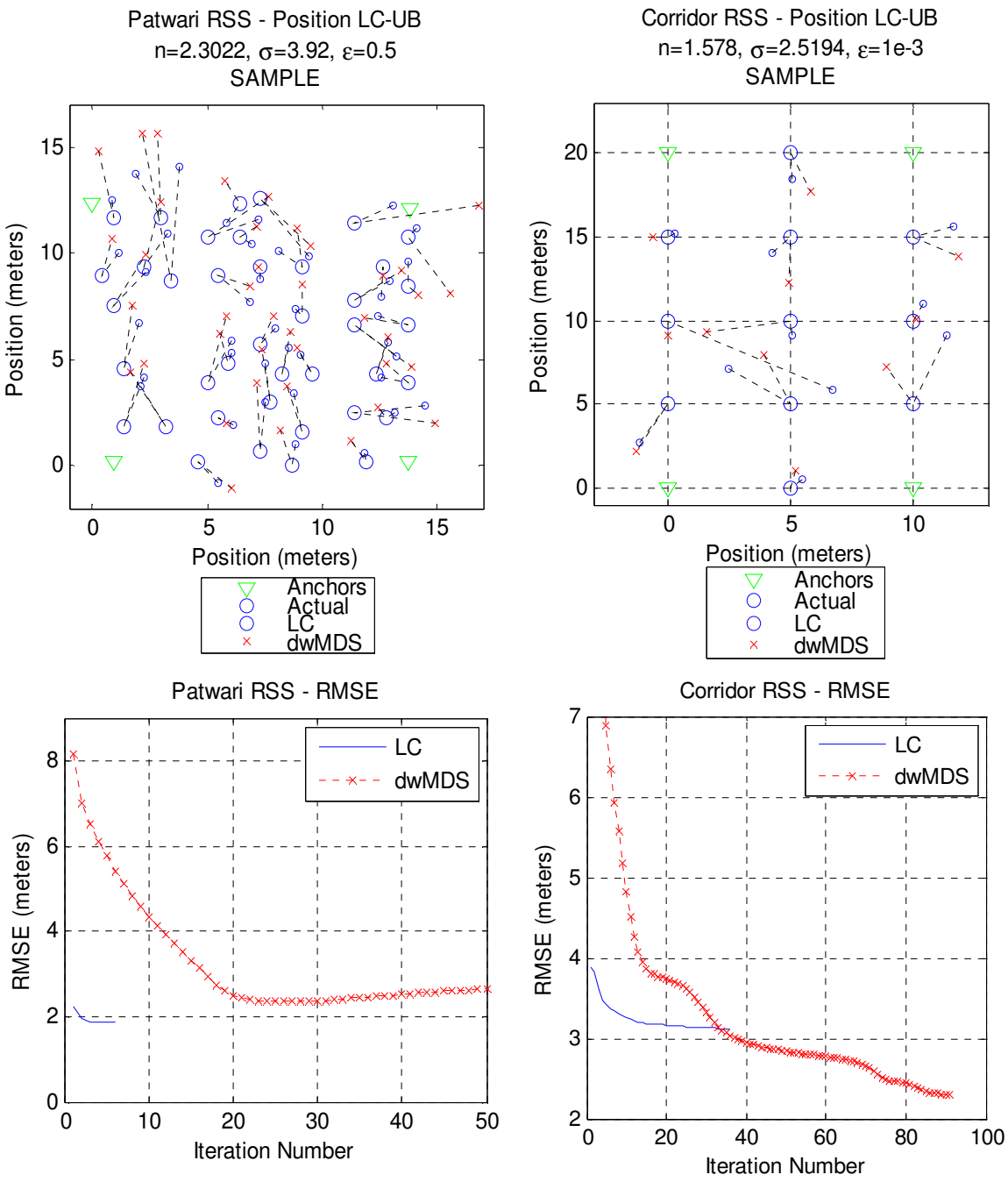

Fig. 32. Sample dwMDS estimates for the 2-D cubicle and 2-D corridor (top). The corresponding RMSE for each environment (bottom).

Table 4

Zachry hallway RSS and actual distance.

RSS (dBm)	Transmitter Location $d_0 = 2.5$ $p_0 = -23.29$										
Receiver Location		1	2	3	4	5	6	7	8	9	10	11
	1	0	-18	-20	-32	-28	-25	-35	-31	-34	-29	-30
	2	-31	0	-16	-18	-36	-35	-37	-25	-37	-29	-37
	3	-36	-22	0	-26	-19	-31	-33	-42	-37	-35	-37
	4	-30	-22	-37	0	-25	-21	-39	-26	-34	-37	-41
	5	-33	-33	-25	-26	0	-19	-19	-23	-34	-36	-39
	6	-36	-36	-33	-24	-23	0	-23	-22	-34	-32	-35
	7	-31	-30	-42	-31	-22	-21	0	-22	-27	-32	-26
	8	-32	-35	-28	-37	-24	-32	-20	0	-26	-19	-35
	9	-36	-37	-37	-32	-33	-33	-31	-20	0	-31	-30
	10	-32	-31	-31	-37	-30	-36	-32	-18	-18	0	-27
	11	-32	-27	-32	-30	-28	-30	-36	-30	-22	-21	0
Distance (feet)	Transmitter Location											
Receiver Location		1	2	3	4	5	6	7	8	9	10	11
	1	0	2.5	5	7.5	10	12.5	15	17.5	20	22.5	25
	2	2.5	0	2.5	5	7.5	10	12.5	15	17.5	20	22.5
	3	5	2.5	0	2.5	5	7.5	10	12.5	15	17.5	20
	4	7.5	5	2.5	0	2.5	5	7.5	10	12.5	15	17.5
	5	10	7.5	5	2.5	0	2.5	5	7.5	10	12.5	15
	6	12.5	10	7.5	5	2.5	0	2.5	5	7.5	10	12.5
	7	15	12.5	10	7.5	5	2.5	0	2.5	5	7.5	10
	8	17.5	15	12.5	10	7.5	5	2.5	0	2.5	5	7.5
	9	20	17.5	15	12.5	10	7.5	5	2.5	0	2.5	5
	10	22.5	20	17.5	15	12.5	10	7.5	5	2.5	0	2.5
	11	25	22.5	20	17.5	15	12.5	10	7.5	5	2.5	0

Table 5

RSS for Wisenbaker (WERC) office environment. True distance is same as Table 5.

RSS (dBm)	Transmitter Location $d_0 = 2.5\ p_0 = -18.8$										
Receiver Location		1	2	3	4	5	6	7	8	9	10	11
	1	0	-19	-33	-26	-36	-34	-29	-29	-30	-33	-35
	2	-12	0	-22	-20	-23	-22	-33	-43	-35	-39	-30
	3	-22	-12	0	-12	-33	-32	-29	-31	-39	-37	-27
	4	-25	-15	-13	0	-38	-27	-25	-34	-40	-40	-30
	5	-38	-34	-30	-30	0	-12	-15	-29	-27	-30	-30
	6	-31	-32	-24	-33	-14	0	-12	-26	-37	-38	-27
	7	-40	-32	-29	-25	-16	-12	0	-22	-28	-28	-27
	8	-30	-28	-35	-33	-34	-17	-10	0	-17	-31	-33
	9	-33	-31	-32	-36	-25	-21	-15	-11	0	-23	-25
	10	-31	-33	-34	-32	-26	-28	-30	-25	-34	0	-13
	11	-39	-36	-33	-27	-24	-27	-27	-29	-28	-17	0

Table 6

RSS for Zachry open space environment. True distance is same as Table 5.

RSS (dBm)	Transmitter Location $d_0 = 2.5\ p_0 = -29.12$										
Receiver Location		1	2	3	4	5	6	7	8	9	10	11
	1	0	-27	-27	-32	-37	-37	-43	-44	-49	-37	-37
	2	-35	0	-31	-36	-34	-34	-39	-43	-43	-40	-39
	3	-28	-33	0	-25	-27	-35	-30	-43	-42	-43	-40
	4	-25	-25	-25	0	-27	-30	-31	-33	-37	-42	-38
	5	-41	-30	-32	-44	0	-25	-30	-32	-31	-39	-41
	6	-43	-33	-35	-28	-32	0	-29	-28	-29	-39	-33
	7	-39	-46	-32	-31	-35	-32	0	-32	-33	-30	-34
	8	-38	-40	-42	-33	-32	-39	-40	0	-30	-34	-32
	9	-35	-44	-36	-40	-34	-30	-25	-33	0	-37	-24
	10	-37	-46	-38	-44	-36	-33	-29	-24	-30	0	-33
	11	-40	-42	-41	-33	-36	-37	-39	-34	-35	-34	0

Table 7

RSS and node location for 2-D library environment.

RSS (dBm)		Transmitter Location $d_0 = 94$ $p_0 = -44.95$															
Receiver Location (upper triangle – Zonet, lower triangle – Intel)		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	1	0	-52.3	-49.1	-55.7	-57.5	-51.3	-50.2	-47.2	-53.1	-58.2	-57.8	-59.9	-67.3	-57.1	-63.4	-59.7
	2	-43.5	0	-49.7	-51.4	-57.5	-53.6	-51	-49.9	-58.8	-61.3	-60.8	-59.5	-66.9	-59.2	-62	-64.6
	3	-50.3	-45	0	-51.7	-49.2	-51.7	-51	-54	-56	-57.9	-58.7	-52.2	-63.9	-57.7	-59.7	-55.9
	4	-53.9	-54.6	-40.6	0	-50.9	-51.1	-51	-58.7	-58.5	-67.1	-53.4	-55.4	-54.1	-58.4	-70.2	-61.4
	5	-63.9	-55.7	-53.6	-46.3	0	-49.3	-52.3	-58.1	-63.6	-64.6	-56	-51.9	-59.5	-63.3	-69.3	-60.9
	6	-53.4	-52.9	-49.6	-42.7	-48	0	-52.7	-51.1	-61.1	-55.8	-53.9	-53.8	-59.5	-58.4	-61.7	-57.6
	7	-48.2	-43	-51.3	-50.8	-58.3	-49.5	0	-51.1	-53.1	-51.4	-54.2	-63.1	-61.7	-59.2	-55.8	-57.3
	8	-43.3	-48.9	-56.6	-62.3	-61.8	-58.8	-50.9	0	-49.9	-52.2	-54.7	-62.3	-64.1	-60.2	-56.2	-52.2
	9	-50	-59.7	-60.8	-67.1	-65.8	-64.9	-52.7	-43.9	0	-51.2	-55	-60.8	-59.2	-62.3	-49	-50.5
	10	-55.2	-61.9	-56.6	-61.4	-63	-64.1	-52.9	-53.4	-55.5	0	-55.5	-59.1	-59.3	-54.4	-49.1	-50.5
	11	-55.8	-59.3	-55.3	-55.1	-59	-54.2	-52.7	-56.9	-61.1	-52.4	0	-49.6	-48.5	-50.6	-64.4	-51.8
	12	-63.7	-65.6	-57.4	-51.3	-50.9	-49.4	-58.5	-61.2	-63.1	-56.3	-44.2	0	-46.2	-55.5	-60.5	-66.4
	13	-66.6	-61.4	-64.7	-54.9	-60	-63.8	-66	-65.2	-63.3	-61.9	-55.8	-49.5	0	-50.6	-63.5	-54.9
	14	-63	-58.9	-61.8	-56	-62	-59.4	-56.7	-59.7	-56.4	-57.8	-41.9	-55.6	-53.5	0	-55.2	-51.4
	15	-63.8	-63.1	-62.4	-62.2	-69.1	-62.9	-55.5	-55.4	-45.6	-52.8	-58.4	-64.5	-59.1	-55.4	0	-48.1
	16	-63.9	-60.1	-61.8	-64.4	-60.9	-64.9	-54.5	-49.3	-44.7	-44.1	-60.6	-62.6	-55.1	-54.7	-41.8	0
Inches	x	0	167	313	456	460	349	167	0	0	96	286	459	467	320	94	0
	y	0	0	0	0	128	128	128	128	277	277	277	277	410	410	410	410

Table 8

RSS and node locations for 2-D corridor environment.

RSS (dBm)		Transmitter Location $d_0 = 5\ \ p_0 = -30.895$														
Receiver Location		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	1	0	-27.3	-39.5	-36.3	-40.6	-42.5	-36.2	-36.9	-35.7	-30.8	-36.0	-35.4	-39.0	-41.2	-40.9
	2	-27.3	0	-30.3	-39.5	-40.1	-36.6	-40.7	-36.6	-29.4	-37.5	-35.5	-35.2	-38.0	-41.4	-40.1
	3	-39.5	-30.3	0	-28.3	-35.2	-37.9	-38.9	-27.7	-34.1	-33.4	-33.2	-34.8	-34.8	-39.3	-37.5
	4	-36.3	-39.5	-28.3	0	-29.4	-33.9	-30.4	-35.8	-35.8	-37.2	-41.1	-33.6	-37.5	-36.7	-36.7
	5	-40.6	-40.1	-35.2	-29.4	0	-29.7	-36.5	-37.3	-36.2	-41.1	-45.9	-39.3	-37.2	-36.0	-36.0
	6	-42.5	-36.6	-37.9	-33.9	-29.7	0	-29.3	-36.0	-37.2	-38.0	-38.2	-36.2	-33.3	-36.4	-29.3
	7	-36.2	-40.7	-38.9	-30.4	-36.5	-29.3	0	-28.6	-35.0	-33.9	-37.1	-36.1	-37.4	-28.0	-37.7
	8	-36.9	-36.6	-27.7	-35.8	-37.3	-36.0	-28.6	0	-28.6	-34.8	-38.9	-33.4	-27.6	-39.1	-39.1
	9	-35.7	-29.4	-34.1	-35.8	-36.2	-37.2	-35.0	-28.6	0	-27.9	-37.7	-30.5	-33.8	-34.7	-36.1
	10	-30.8	-37.5	-33.4	-37.2	-41.1	-38.0	-33.9	-34.8	-27.9	0	-30.3	-41.8	-35.9	-36.4	-41.4
	11	-36.0	-35.5	-33.2	-41.1	-45.9	-38.2	-37.1	-38.9	-37.7	-30.3	0	-29.9	-35.1	-35.4	-43.9
	12	-35.4	-35.2	-34.8	-33.6	-39.3	-36.2	-36.1	-33.4	-30.5	-41.8	-29.9	0	-28.2	-34.6	-35.1
	13	-39.0	-38.0	-34.8	-37.5	-37.2	-33.3	-37.4	-27.6	-33.8	-35.9	-35.1	-28.2	0	-29.0	-36.5
	14	-41.2	-41.4	-39.3	-36.7	-36.0	-36.4	-28.0	-39.1	-34.7	-36.4	-35.4	-34.6	-29.0	0	-30.5
	15	-40.9	-40.1	-37.5	-36.7	-36.0	-29.3	-37.7	-39.1	-36.1	-41.4	-43.9	-35.1	-36.5	-30.5	0
Meters	x	0	0	0	0	0	5	5	5	5	5	10	10	10	10	10
	y	0	5	10	15	20	20	15	10	5	0	0	5	10	15	20

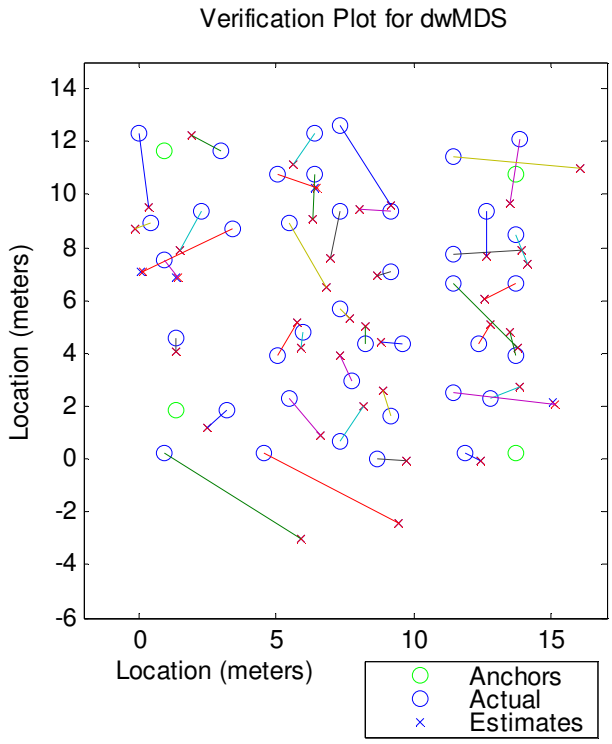

Fig. 33. This plot verifies correct implementation of the dwMDS algorithm. The results are nearly identical to [1] Figure 7d. The RMSE for this plot was 2.27 m as compared to 2.48 m in [1, p. 60].

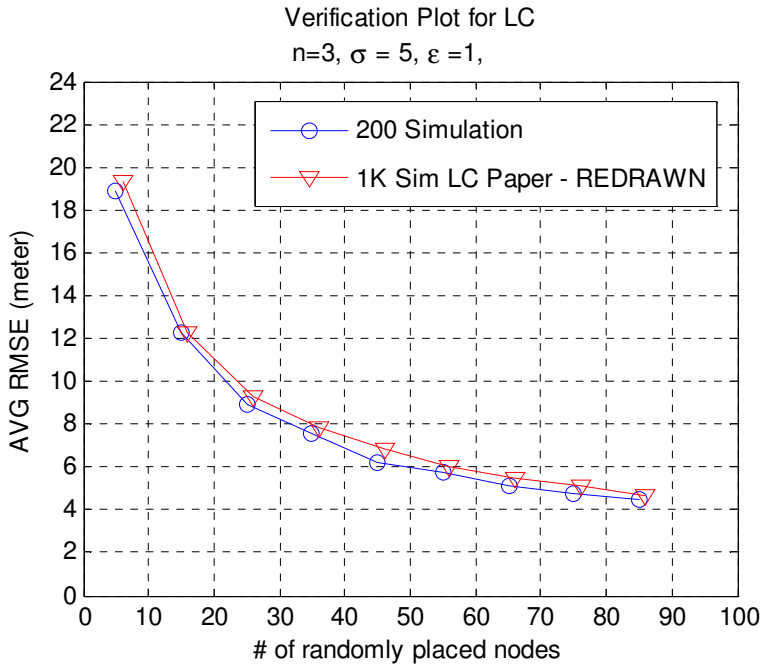

Fig. 34. This plot verifies correct implementation of the LC algorithm. The graph was redrawn from [2].

APPENDIX B

This appendix will give some details about the implemented algorithms such as initialization and weights assigned.

B.1 dwMDS Details

For a more complete analysis of dwMDS, please consult [1]. dwMDS tries to minimize a cost function labeled as, $\mathcal{S}$. This cost function can be reduced into local cost functions, $\mathcal{S}_i$, for each of the n blindfolded nodes.

$$\mathcal{S} = \sum_{i=1}^{n} \mathcal{S}_i \tag{1}$$

$$\mathcal{S}_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij}(\delta_{ij} - d_{ij}(X))^2 + \sum_{j=n+1}^{n+m} 2w_{ij}(\delta_{ij} - d_{ij}(X))^2 \tag{2}$$

where w_{ij} is a weight assigned for the node i to node j connection (more on this later). δ_{ij} is the range estimate based upon the power matrix between node i and j. m is the number of anchors (two for 1-D and four for 2-D). X is a coordinate matrix containing the estimate locations, so $d_{ij}(X)$ is the distance between the estimates of node i and node j. X is arranged such that each column holds the coordinates of each node. The first n columns are node coordinates followed by m columns for the anchors coordinates. Since dwMDS is iterative-based, the X matrix and S will change each iteration, k, as the estimates move, so it is common to denote X as $X^{(k)}$ where $X^{(0)}$ is

the initial estimate of all nodes, and S as $S^{(k)}$. w_{ij} and δ_{ij} will remain constant throughout iterations.

The algorithm begins with the user providing the weights, range estimates, and initial estimations. $S^{(0)}$ is computed along with another set of weights denoted as a.

$$a_i^{-1} = \sum_{\substack{j=1 \\ j \neq i}}^{n} w_{ij} + \sum_{j=n+1}^{n+m} 2w_{ij} \qquad i = 1,....,n \tag{3}$$

The algorithm is a nested loop. The outer loop stops when $S^{(k-1)} - S^{(k)} < \varepsilon$ (we begin with $k = 1$) where ε is a user-specified number. The inner loop iterates across all n blindfolded nodes. During each iteration of the inner loop, $S^{(k)}$ is being modified as each node updates its current estimation (i.e. $S^{(k)} = S^{(k)} - S_i^{(k-1)} + S_i^{(k)}$). The nodes calculate their new position with a third set of weights for each node i, denoted as $\vec{b}_i$. $\vec{b}_i$ depends on the current estimates ($X^{(k)}$) and will change across iterations.

$$\vec{b}_i^{(k)} = [b_1,...,b_{n+m}]^T \qquad i = 1,....,n \tag{4}$$

$\vec{b}_i$ will be the weighting for each of the nodes that node i can establish a connection.

$$\begin{aligned} b_j &= w_{ij}\left[1 - \delta_{ij} / d_{ij}(X^{(k)})\right] && \text{for } j \leq n,\ j \neq i \\ b_j &= \sum_{j=1}^{n} w_{ij}\delta_{ij} / d_{ij}(X^{(k)}) + \sum_{j=n+1}^{n+m} 2w_{ij}\delta_{ij} / d_{ij}(X^{(k)}) && \text{for } j = i \\ b_j &= 2w_{ij}\left[1 - \delta_{ij} / d_{ij}(X^{(k)})\right] && \text{for } j > n \end{aligned} \tag{5}$$

The new estimate, $x_i^{(k)}$, of each node i is calculated by:

$$\hat{x}_i^{(k)} = a_i\left(X^{(k-1)}\vec{b}_i^{(k-1)}\right) \qquad i = 1,....,n \tag{6}$$

Once $x_i^{(k)}$ is found, $S^{(k)}$ is adjusted and the new position estimate is sent to all the neighbors nodes.

The first set of weights, w_{ij}, was chosen based upon [1, p. 55].

$$w_{ij} = \exp\left(-\delta_{ij}^2 / h_{ij}^2\right) \quad \text{for valid } \delta_{ij}\text{, else 0.} \qquad (7)$$
$$h_{ij} = \max\left\{\max_k \delta_{ik}, \max_k \delta_{kj}\right\}$$

δ_{ij} was calculated from the power matrix using a maximum likelihood estimator of the form:

$$\delta_{ij} = d_o\left(\frac{P}{p_o}\right)^{-1/n} \qquad (8)$$

where d_o is a reference distance and p_o is the power at that distance. Both P and p_o are measured in watts. n in this particular equation is the path loss exponent (not to be confused with the number of nodes).

The initialization used was a 2-stage process according to [1, p. 53]. dwMDS was run twice with the first run having a random initialization and a fairly large ε. The w_{ij} was assigned based on (7) with δ_{ij} being valid if it was less than a user-specified "connection distance" (section 4.1.4). When finished, these rough estimates were then used as the initial estimates for the second run. The weights were re-assigned in the second run based on (7), with a valid weight being determined not by δ_{ij}, but by $\left\|\hat{x}_i - \hat{x}_j\right\|$ being less than a user-specified "neighbor distance". The second run had a smaller ε.

B.2 LC Details

This section will go into further detail of the LC algorithm. Like dwMDS, the LC algorithm is distributive and also relies on weighting to calculate a nodes position estimate. The algorithm is also iterative-based and terminates when the cumulative squared difference of estimates from one iteration to the next becomes less than a user-specified ε.

First, the unbiased range estimator converts the power matrix into distances.

$$\hat{d}_{ij} = d_o \left(\mathcal{P} / p_o \right)^{-1/n} \exp(-0.5 b^2) \tag{9}$$

where $b = (\sigma \ln 10)/(10n)$ and power is measured in a linear scale (i.e. watts). Note that $\hat{d}_{ij}$ represents the distance to node i based upon node j's perspective (i.e. i is the TX, and j is the RX). This notation will be used throughout B.2. Next is the initialization of each node based upon anchors. For 1-D, we estimate an initial position by

$$\hat{x} = a_1 \hat{d}_1 + a_2 (\mathcal{R} - \hat{d}_2) \tag{10}$$

$$a_1 = \frac{\hat{d}_2}{\hat{d}_1 + \hat{d}_2} \text{ and } a_2 = \frac{\hat{d}_1}{\hat{d}_1 + \hat{d}_2} \tag{11}$$

where $\hat{d}_1$ is the estimated distance to the first anchor at coordinate 0 and $\hat{d}_2$ is the estimated distance to the 2nd anchor at coordinate $\mathcal{R}$. For 2-D, it is not as simple. The 2-D initialization is iterative and terminates when the squared difference of estimates between one iteration and the next is less than an ε. We begin with $k = 1$ and by

placing nodes in the center of the area $\left[\hat{x}^{(0)}, \hat{y}^{(0)}\right] = [\mathcal{R}/2, \mathcal{R}/2]$. Normalized vectors are created from the node to each of the four anchors.

$$v_i^{(k)} = \frac{\left[\hat{x}^{(k-1)} - x_i, \hat{y}^{(k-1)} - y_i\right]}{\left\|\left[\hat{x}^{(k-1)} - x_i, \hat{y}^{(k-1)} - y_i\right]\right\|} \qquad \text{for } i = 1,...,4 \quad (12)$$

where $\|\cdot\|$ is the Euclidean norm. Weights are computed by

$$a_i = \frac{1/\hat{d}_i^{\,2}}{\sum_{j=1}^{4} 1/\hat{d}_j^{\,2}} \qquad \text{for } i = 1,...,4 \quad (13)$$

and an estimate is found by

$$\left[\hat{x}^{(k)}, \hat{y}^{(k)}\right] = \sum_{i=1}^{4} a_i \left(\left[x_i, y_i\right] + \hat{d}_i v_i^{(k)}\right) \quad . \qquad (14)$$

The iterations stop when

$$\left(\hat{x}^{(k)} - \hat{x}^{(k-1)}\right)^2 + \left(\hat{y}^{(k)} - \hat{y}^{(k-1)}\right)^2 < \varepsilon \, . \qquad (15)$$

If a node is not able to connect to all four anchors (section 4.1.4), then it will estimate its position based upon whichever anchors are available. If no anchors are available, the node is placed at the center.

With an initial estimate, cooperative locating can begin. $\mathcal{M}$ is the number of blindfolded nodes and $[\mathcal{M}+1, \mathcal{M}+4]$ are the four anchors. For the 1-D case, we create a set of distances denoted as "estimated effective distances" for all connecting nodes

$$\left(\delta_{ij}^{(k)}\right)^2 = \frac{\left(\hat{x}_{jj}^{(k-1)}\right)^2\left(\mathcal{R}-\hat{x}_{jj}^{(k-1)}\right)^2}{\left(\hat{x}_{jj}^{(k-1)}\right)^2+\left(\mathcal{R}-\hat{x}_{jj}^{(k-1)}\right)^2}+\left(\hat{x}_{ii}^{(k-1)}-\hat{x}_{jj}^{(k-1)}\right)^2 \quad \text{for } i,j=1,\ldots,\mathcal{M}. \tag{16}$$

The new node estimate is computed by

$$\hat{x}_{ii}^{(k)} = \sum_{j=1}^{\mathcal{M}} a_{ij}^{(k)}\left(\hat{x}_{jj}^{(k-1)}+\operatorname{sgn}\left(\hat{x}_{ii}^{(k-1)}-\hat{x}_{jj}^{(k-1)}\right)\hat{d}_{ij}\right) \quad \text{for } i,j=1,\ldots,\mathcal{M} \tag{17}$$

$$a_{ij}^{(k)} = \frac{1/\left(\delta_{ij}^{(k)}\right)^2}{\sum_{l=1}^{\mathcal{M}} 1/\left(\delta_{il}^{(k)}\right)^2} \quad \text{for } i,j=1,\ldots,\mathcal{M} \tag{18}$$

and the iterations stop when

$$\sum_{i=1}^{\mathcal{M}}\left(\hat{x}_{ii}^{(k)}-\hat{x}_{ii}^{(k-1)}\right)^2 < \varepsilon \tag{19}$$

For the 2-D case, we compute the normalized vectors between connecting nodes.

$$v_{ij}^{(k)} = \frac{\left[\hat{x}_{ii}^{(k-1)}-\hat{x}_{jj}^{(k-1)},\,\hat{y}_{ii}^{(k-1)}-\hat{y}_{jj}^{(k-1)}\right]}{\left\|\left[\hat{x}_{ii}^{(k-1)}-\hat{x}_{jj}^{(k-1)},\,\hat{y}_{ii}^{(k-1)}-\hat{y}_{jj}^{(k-1)}\right]\right\|} \quad \text{for } i\neq j,\ v_{ii}^{(k-1)}=0 \tag{20}$$

The "estimated effective distances" are computed for all connecting nodes.

$$\left(\delta_{ij}^{(k)}\right)^2 = \frac{1}{\sum_{l=\mathcal{M}+1}^{\mathcal{M}+4} 1/\left(\left(\hat{x}_{jj}^{(k-1)}-\hat{x}_{l}\right)^2+\left(\hat{y}_{jj}^{(k-1)}-\hat{y}_{l}\right)^2\right)} + \left(\hat{x}_{ii}^{(k-1)}-\hat{x}_{jj}^{(k-1)}\right)^2+\left(\hat{y}_{ii}^{(k-1)}-\hat{y}_{jj}^{(k-1)}\right)^2 \tag{21}$$

The new node estimate is calculated by

$$\left[\hat{x}_{ii}^{(k)}, \hat{y}_{ii}^{(k)}\right] = \sum_{j=1}^{\mathcal{M}} a_{ij}^{(k)} \left(\left[x_{jj}^{(k-1)}, y_{jj}^{(k-1)}\right] + \hat{d}_{ij} v_{ij}^{(k)}\right) \tag{22}$$

$$a_{ij}^{(k)} = \frac{1/\left(\delta_{ij}^{(k)}\right)^2}{\sum_{l=1}^{\mathcal{M}} 1/\left(\delta_{il}^{(k)}\right)^2} \qquad \text{for } i, j = 1,...,\mathcal{M}. \tag{23}$$

The iterations stop when

$$\sum_{i=1}^{\mathcal{M}} \left(\hat{x}_{ii}^{(k)} - \hat{x}_{ii}^{(k-1)}\right)^2 + \left(\hat{y}_{ii}^{(k)} - \hat{y}_{ii}^{(k-1)}\right)^2 < \varepsilon. \tag{24}$$

VITA

Name:	Felix Gutierrez, Jr.
Address:	Department of Electrical and Computer Engineering, Texas A&M University, 214 Zachry Engineering Center, TAMU 3128 College Station, Texas 77843-3128
Email Address:	felixgutierrez06@gmail.com
Education:	B.S., Electrical Engineering, The University of Texas at Austin, 2006 M.S., Electrical Engineering, Texas A&M University, 2008