

# IMPLEMENTATION OF COLLABORATIVE RF LOCALIZATION USING A SOFTWARE-DEFINED RADIO NETWORK

Augustine A. Honoré, Ryan W. Thomas, Richard K. Martin, and Stuart H. Kurkowski

Air Force Institute of Technology  
Wright Patterson Air Force Base, OH 45433

## ABSTRACT

*Current U.S. military pursuits, such as the Joint Tactical Radio System (JTRS), are based on a software-defined radio (SDR) system. These kinds of systems use the flexibility of software to make old and new radio technology interoperable. Co-existence of different radio technologies has the potential to be a force enabler, but it still falls short, particularly for military operations in which failure to co-exist can cause interference fratricide between allied radios. We designed a collaborative network of cognitive radios derived from flexible, commercially-available SDRs. Using a network of SDRs as an experimental test bed, we implemented a detection and RF characterization algorithm, gathering signal data from multiple spatially diverse points in the network. This provided three main contributions: determining the process for pre-characterizing the SDRs; developing the measurement procedure for effective transmitter detection and estimation; and identifying effective real-world network topologies.*

## INTRODUCTION

Today, joint military access to the electromagnetic spectrum is coordinated through the Joint Frequency Management Office (JFMO). Its objective is to enable effective joint spectrum use for expected military operations. However, we know that missions change according to the situation at hand, and likewise spectrum demands change. Spectrum access becomes the limiting factor for rapid allied troop maneuvering and commanders must spend time renegotiating for their slice of the electromagnetic pie. Although spectrum approval authority can be delegated by the JFMO, determining how to do so at the device level while adhering to legal constraints is a difficult problem.

Practical access of the unused spectra (also called whitespace), whose availability shifts dynamically in space and time, requires an adaptive solution. In the last decade, a bold solution was proposed – an autonomous agent that proactively makes decisions to assist a user – in order to make spectrum access and other communication-based

This work is sponsored by AFRL and AFOSR. The views expressed in this paper are those of the authors, and do not reflect the official policy or position of the U.S. Air Force, Department of Defense, or the U.S. Government. This document has been approved for public release, distribution unlimited.

tasks feasible despite dynamic and constrained operating environments. The cognitive radio, as introduced by Mitola and Maguire in 1999 [1], is centered entirely on a sole user, acting as a personal assistant that delivers end-user capabilities through its ability to observe, adapt, and learn.

Dynamic spectrum access, however, is not a problem best solved through individual radio effort. In previous research, a cognitive radio's estimation of its current radio frequency (RF) environment has been shown to become significantly more accurate when performed in cooperation with other cognitive radios [2], [3]. A network of collaborating cognitive radios (a cognitive network) benefits from a shared representation of the RF environment by creating a more complete picture of their dynamic surroundings. Some essential components of understanding how to create this “map” of the RF environment include identifying the presence of primary users, determining their spatial positions, estimating their antenna patterns, and characterizing the propagation environment. In utilizing the RF environment, primary allied spectrum users may continue to operate unimpeded, and the collaborating cognitive radios (secondary users) can productively share the limited medium.

## BACKGROUND

Previous research has examined the process of characterizing the RF environment by mapping spectrum usage in space, time, frequency, and code. The 5.1 dimensional RF topography developed by Martin and Thomas [4] uses simulation results to demonstrate their transmitter characterization algorithm which identifies the presence, positions, and antenna patterns of primary users within a search space populated by CR nodes cooperating in a noisy environment. Using the received signal strength (RSS) obtained at each receiving sensor and known receiver positions, they have demonstrated how their algorithm can be used to improve decisions on spectrum availability in a dynamic spectrum access system.

RSS is a measure of how strong a signal is when it arrives at a sensor. It is commonly taken as a voltage measurement, or equivalently calculated as a signal power (e.g. the magnitude squared). Measurements can be made

from acoustic, RF or other types of signals without infringing on bandwidth or requiring complex hardware. However, RSS measurements are known to vary unpredictably usually because of operating environment conditions [5]. The most influential sources of error are due to multipath propagation and shadowing. Multipath is a phenomenon that destructively (or constructively) combines signals of differing amplitude and phase orientations that have traversed multiple paths prior to arriving at the receiver. Shadowing is the attenuation that results when a signal is forced to go through or bend around obstacles such as walls or trees. Despite these hazards, the relative simplicity and low cost of RSS-based techniques makes them attractive solutions for localization tasks.

GNU Radio [6] is a free software development toolkit specializing in signal processing. Part of its success is derived from a flexible, process block abstraction which allows software developers to manipulate signals by appending a series of individual signal processing events. The GNU Radio project developed the Universal Software Radio Peripheral version 1 (USRPv1) as a relatively low cost (\$800) software radio. It too gained wide adoption through flexibility – offering a hardware platform that is easily reconfigured by adding or removing interchangeable daughterboards, each designed to operate within specific bands of the EM spectrum (DC to 5.9 GHz).

There is a large amount of existing work on localization using RSS measurements. In [7], a Maximum Likelihood (ML) estimator was derived for self-localization of a network of omnidirectional sensors, in which small subsets of the sensors were “anchor nodes” at known locations. In [8, 9], the Barankin and Cramer Rao Lower Bounds (respectively) for such location estimation techniques were derived. Outside of theory, there have been successful implementations of RSS localization schemes, but these focus on existing sensor platforms with RSS capabilities. The work here is the first to implement a ML RSS scheme on a cheap, off-the-shelf SDR, the USRPv1.

The USRP is a desirable platform for exploring transmitter characterization because of its flexibility. Unlike hardware radios and sensors, a USRP does not have to assume anything about the waveform or frequency of the transmit source. It can sense over a wide frequency band before detecting the transmitters in software. From an architectural point of view, USRPs can handle and use measured data themselves, instead of having to pass it off to a secondary computing device. Finally, since USRPs are increasingly being used as cognitive radios, developing this functionality on the USRP will speed up spatial RF environment characterization for cognitive radios.

## EXPERIMENTAL METHODOLOGY

Our approach organized the data collection and data reduction procedures as two distinct steps. First the USRP hardware was used to capture signal data using a GNU Radio Python script. Next, our MATLAB script was applied to convert, calculate, and extract RSS information from the signal data.

In the first step, a Python script for capturing raw signal data was developed, using the USRP hardware to iteratively capture 128  $\mu$ s of complex signal data, and storing the data on a binary-encoded file according to the measurement parameters: decimation rate, Fast Fourier Transform (FFT) size, tuning frequency, and file name. Once all signal data was collected by all participating sensing nodes, the data was then reduced using MATLAB.

In the second step, the data reduction script was designed to take the signal data files collected by the USRP and reduce them to a time-varying list of received power values given a station of interest. To do this, the binary files were first read and their complex signal data were extracted as two separate signals: the real (in-phase) and imaginary (quadrature phase) parts. Once the signal was converted to a frequency domain representation, a power spectral density (PSD) was calculated in order to determine the received power at each frequency bin.

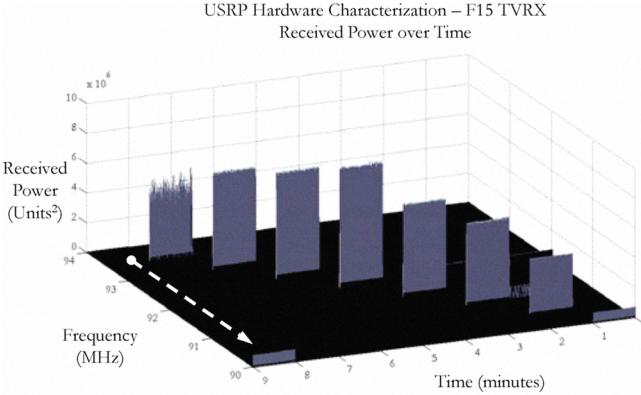
Using a PSD made it easy to identify some of an FM station's features such as peak power. When combined with the PSDs of all subsequent files it became possible to see how the signal strength of a station (henceforth called channel) fluctuated over time. Because of the previously identified problems with RSS measurements (fading, multipath) the captured PSD had a time-varying property, revealing a pervasive “noise” in the measurement. The additive effects of this “noise” made it difficult to precisely identify the lower and upper frequencies of the radio channels.

In order to minimize these effects the script took the average of the PSDs over a sliding time window. By using the time-averaging process

$$PSD_{avg}[i] = \frac{1}{M} \sum_{j=t+1}^{t+M} PSD_j[i] \quad (1)$$

where  $M$  is the total number of PSDs calculated during the desired time interval and  $PSD_j[i]$  is the  $i$ th element of the  $j$ th PSD calculated. Under this scheme, the time varying aspects of the noise in PSD shape were cancelled, allowing more readily identifiable features by our automated station detection algorithm.

This algorithm identified occupied FM channels using another sliding window that compared the power contained in clusters of frequencies (a local average power) against the average power contained within the entire viewable bandwidth of the time-averaged PSD (a global average power). If the local average power was greater than or equal to the global average power, we estimated that a channel had a transmitter present in it.



**Figure 1:** Time-varying plot of the characterization signal's PSD for a single USRP

$$\frac{1}{w} \sum_{i=n+1}^{n+w} PSD_{avg}[i] \begin{cases} \geq \frac{1}{N} \sum_{i=1}^N PSD_{avg}[i] & tx \\ < \frac{1}{N} \sum_{i=1}^N PSD_{avg}[i] & no\ tx \end{cases} \quad (2)$$

In Equation 2,  $n+1$  and  $n+w$  are proportional to the starting and ending frequencies of the channel being tested and  $N$  is proportional to the size of the bandwidth of the entire PSD. For a series of windows that test positive for an existing transmitter, the center frequency of the channel is discerned to be a function of the average center frequency bin of those windows' value, and the bandwidth a function of the total number of windows and window sizes.

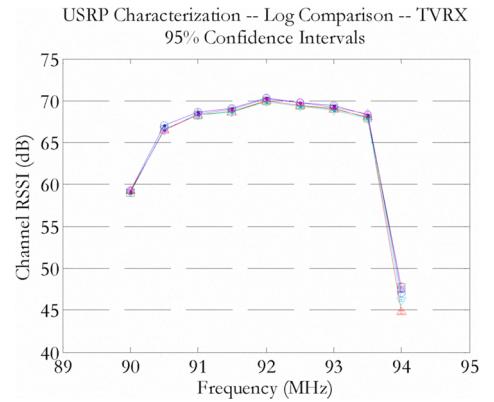
Aside from external sources of measurement error, particularly multipath fading and shadowing for RSS-based applications, it is also important to recognize internal sources of error. Energy-based localization techniques greatly depend on how closely a sensor set responds given the same input conditions [5]. The RSS technique we implemented is of no exception. Thus, we developed a procedure that helped to determine the uniformity of our sensing nodes. Using the same input signal applied to each sensor, we were able to calibrate the USRPs.

Each sensor was given a -3 dBm (158.3 mV<sub>0</sub>-peak) sine wave from a signal generator into the receive port – a daughter board capable of receiving FM radio and analog television signals (50 - 860 MHz). The -3 dBm signal was swept through nine evenly spaced frequencies covering the bandwidth of our experiment (90.0-94.0 MHz in steps of 0.5 MHz) and dwelled at each frequency for one minute before advancing to the next. Figure 1 depicts the power received by one USRP over time (right to left), and frequencies (front to back).

Once all of the radios finished sampling the characterization signal, the sample data was reduced by extracting only those areas where the swept signal was present (the gray pillars of data in Figure 1). On average, each pillar within the sweep band formed a channel

approximately 70 kHz wide. The channels occurring at 90 MHz and 94 MHz were not relied on because they were subject to clipping effects. Additionally, the signal energy intended for 94 MHz was displaced to the other side of the viewable band. This is true for all sample data sets, and is highlighted in Figure 1 with an arrow. Subsequent activities were designed to avoid the bounding frequencies.

After computing the average received signal power within all channels we were able to compare the channel RSS values, illustrated in Figure 2. The fact that the received power is not the same for all radios is due to the loose manufacturing tolerances for the analog front end components of the USRP. In particular, gains are not applied identically between the TVRX daughterboards we used, causing the slight variation in the TVRX linear-scale plot. However, Figure 2 reveals that, in general, the TVRX daughterboards respond within 0.46 dB of one another for a -3 dB input signal. The fall off at 90.0 and 94.0 MHz is due to the filter characteristics on the TVRX card.



**Figure 2:** USRP hardware characterization results in the log domain

Only after we began the data reduction process in MATLAB were we able to identify a problem in the data collection process. There were some instances in which the complex signals were subject to an abnormal “ringing” effect within the first 23 samples (or 5.75  $\mu$ s) of data. In the frequency domain, these large, narrow impulses transformed into broad spectral densities that dwarfed all other PSDs. Given the transient nature of the ringing and their presence only at the beginning of some sampling iterations, we suspected that somewhere in the USRP a power surge occurs when it is commanded to start sampling. Our solution was to extend the number of samples we would normally collect by specifying an intermediate FFT size of  $512 + 23 = 535$  points in the data collection code, and then we removed the first 23 samples for all signal data files as they were imported into MATLAB.

In [4] Martin and Thomas derived a new sensor localization algorithm that applies a ML approach to

estimate a transmitter's position, orientation, beam width, and transmit power using RSS measurements. A large portion of their paper's focus was centered on determining additional transmitter parameters such as power, path loss, transmitter directionality and beamwidth. Of particular interest was the fact that their ML algorithm that could determine directionality parameters, as previous research generally ignored non-uniform antenna gain patterns. To demonstrate their algorithm, they created a MATLAB simulation which modeled sensing and transmitting nodes that operated within a log-normal fading environment under various topologies of receivers. Because we required the amplification capabilities of the USRP TVRX daughterboard, practical considerations from the low VHF/UHF frequencies of this board (50-870 MHz) prevented deployment of a directional antenna. Instead, we experimentally performed omnidirectional localization – an important subset of the ML algorithm capabilities.

In general, the localization algorithm for omnidirectional nodes uses the same approach as for directional nodes. Beforehand, all receiving nodes are located within a Cartesian coordinate plane. The sensing nodes (whose positions are known) observe the received power from a transmitter located at some unknown point in the plane. The ML algorithm takes the following arguments: each sensing node's x and y coordinates, the RSS value observed by each sensing node, and two vectors that define the boundaries and number of points within a search space (one vector for each orthogonal dimension).

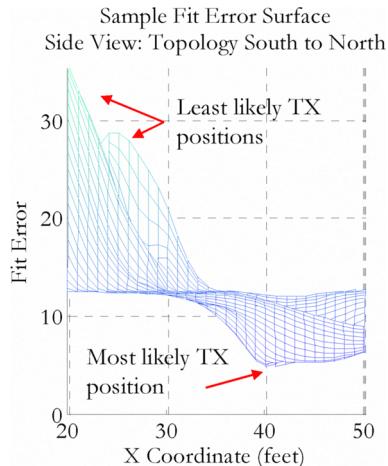
Using the observed RSS values and the locations of the nodes, every point in the grid is evaluated to identify the likelihood that a transmitter resides there. The preliminary calculations identify the mean distances (and mean squared-distances) between all sensing nodes and the current search point, the variance of the sensors' distances to the current search point, the average RSS value received by the sensor network, and the mean power-distance product - an average of the power received by each sensor, scaled by their separation distance to the current search point.

The likelihood calculations are performed in two phases. First, estimates of the most likely transmitter characteristics are calculated, which assumes that a transmitter exists at the current search point. Then, an error is calculated between the supposed transmitter characteristics (what was observed) and the best possible transmitter characteristics (what would have been observed). These calculations (as derived in [4]) are expressed as:

$$\begin{aligned} P0_{best} &= \frac{\langle p \rangle \langle d^2 \rangle - \langle p \cdot d \rangle}{var(d)} \\ np_{best} &= \max\left(\frac{\langle p \rangle \langle d \rangle - \langle p \cdot d \rangle}{var(d)}, 0\right) \\ fit_{error} &= \text{norm}(p - P0_{best} - (np_{best} \cdot d)) \end{aligned} \quad (3)$$

where  $P0_{best}$  is the transmitter power that would have been observed by receivers had the transmitter been at the search point,  $np_{best}$  is the calculated path loss exponent,  $fit_{error}$  is the normalized difference between the power observed and the power that would have been observed,  $p$  is the power received by the sensing nodes (arranged as a vector),  $d$  is the distances between each node and the current search point (arranged as a vector),  $\langle \rangle$  represents a vector mean, and  $var()$  is the variance of the vector values.

The fit error is a direct representation of the likelihood that a transmitter (with similar observed properties) exists at the current search point. After all search points have been evaluated, the computed error values form a matrix whose entries coincide with the search grid. Therefore, the search point that bears the lowest error value represents the most likely position of the transmitter. Figure 3 illustrates a sample fit error matrix taken as a surface and viewed from the side.



**Figure 3:** Example fit error surface viewed from the side

Before integrating the localization algorithm into our experiments, we made note of some usage considerations. Our first concern regarded how fine the search grid would be “drawn.” Having fewer points meant having fewer cumulative calculations but would yield coarser, and thus poorer, position estimate resolution. For the scope of the experiment we performed, we determined a 1 foot by 1 foot resolution was a reasonable resolution to start with, requiring 2500 calculations (for our 50'x50' experimental tests) and operating at an acceptable resolution.

Our second concern regarded sensor placement within the search space. We designed our experiment such that the transmitters would be placed on top of a search grid point. In this way, we could accurately determine whether a position estimate was correct, and if not, calculate a position error with a floor of 0 feet.

We had six USRP software-defined radios, divided into two roles; five of the radios were declared sensing nodes, and one radio was declared a transmitting node. The

sensing nodes ran the Python script to collect signal data and the transmitting node broadcasted audio signals using an FM radio transmission program. Having too few sensing nodes would severely limit the number of shapes and sizes of our sensing topologies. Also, sparse topologies have been analytically shown to degrade the performance of the algorithm [4]. Thus, we developed a sensible way to expand our design options so that we could increase the likelihood of achieving meaningful results.

Our post-processing approach separates measurement and processing into two distinct phases. This distinction provided an opportunity well-suited for experimental analysis. Since data collection and data reduction did not occur concurrently, we were able to conduct multiple small-topology experiments at different points in time. We then combined the data from the experiments as if they occurred concurrently. Finally, we applied our data reduction process to the accumulated data.

The sensing nodes were arranged in positions one through five as shown in Figure 4 and collected signal data from transmitters located at nearby positions (represented by the squares), with locations unknown to them. Then, the sensing nodes were rearranged to positions six through ten to observe the same transmitters from different locations. By combining the signal data from our five sensing nodes (that sensed the transmitter from two independent locations), we essentially emulated a ten node topology.

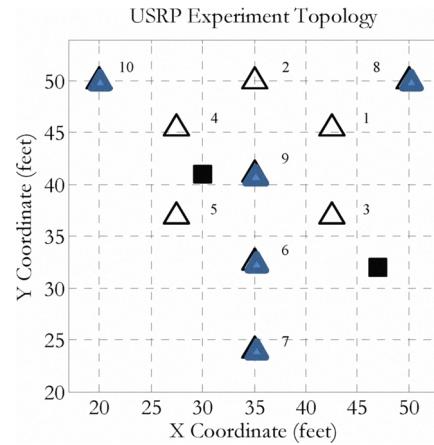
The flexibility of the post-processing approach allowed us to capitalize further on the time shift concept. By excluding the data collected by some sensors, we could select a number of sub-topologies from our combined set of 10 nodes. For example, excluding the outermost sensors in Figure 4 forms a hexagon topology with a node at its center. Repeating this process, we were able to identify 21 sub-topologies that varied by the number of nodes, shape, and size (as in perimeter). The sub-topologies took the following forms: Triangle, Rectangle, Hexagon, Line, and one topology that included all of the sensors. With the exception of the Line and All-Sensors topologies, every other topology type was varied by excluding, and then including, the node located in its center.

Although time shifting added a great deal of flexibility to our post-processing approach, its benefit would be moot unless our experiment design included an accurate data-alignment process, due to the time varying nature of the signal being transmitted (an FM modulated audio sample). When we first introduced our data collection process, we explained that each signal data file was time stamped with the current system time. This timestamp was associated with the system time immediately before the signal capture function call and each signal capture could be identified according to when it was collected. However, in order to align the separate sets of data, they needed to be time

stamped with respect to a common reference. Thus, we established a common time reference by creating a wired local area network. The host computers for all sensing nodes were joined to the same subnet as the transmitter's host computer, which acted as the network time protocol (NTP) server. Under this architecture, the sensors' host computers would synchronize their system clocks by polling the time server upon system boot-up and periodically thereafter.

Our final design measure in support of time shift regarded how we implemented timestamps during the data reduction process. As stated before, the timestamps were given as Epoch time, which denotes the number of seconds since midnight of January 1, 1970. To help simplify our alignment and time-averaging processes, we ignored partial-second increments by truncating the timestamps to whole numbers of seconds. Doing this changed the "resolution" of contiguous data captures to be relative to the nearest second. Thus, when we applied our time-averaging process to datasets collected by any given node, we specified an averaging interval of one second.

As given in Figure 5, our collaboration experiment was conducted outdoors in an uncovered parking lot. Our equipment setup was no less than 50 feet away from vehicles or other large RF reflective objects. Weather conditions were more accommodating than usual for a typical Ohio winter: a high temperature in the low 40s, clear skies, 70% humidity, and winds from the South-



**Figure 4:** Complete sensor topology as a result of applying time shift

southwest averaging 10 mph. At the beginning of every phase, each sensing node was remotely started using a remote desktop application. Once all of the nodes had begun collecting signal data, the transmitter node was remotely activated, and a digital timer was started. After 61 seconds had elapsed, the transmitter was turned off and then the sensing nodes were commanded to stop signal collection. Upon completion of the final phase, all signal

data were retrieved from the host computers, and organized in preparation for data reduction and analysis.



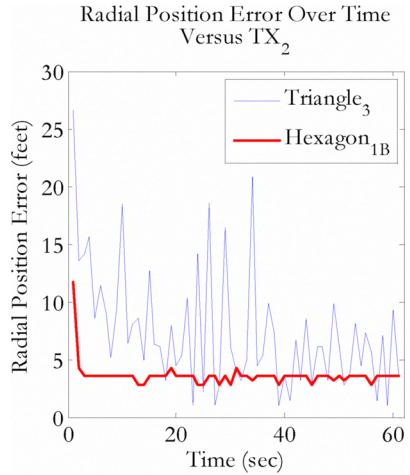
**Figure 5:** Sensing and transmitting nodes arranged outdoors with power and network cabling

## DATA ANALYSIS

Our performance metrics were derived from the position estimates that were generated by our localizing sub-topologies. Using the known transmitter locations, we calculated position errors by evaluating

$$\text{error}_{\text{radial}} = \sqrt{(x_{\text{guess}} - x_{\text{true}})^2 + (y_{\text{guess}} - y_{\text{true}})^2} \quad (6)$$

where  $x_{\text{guess}}$  and  $y_{\text{guess}}$  are the estimated transmitter coordinates; and  $x_{\text{true}}$  and  $y_{\text{true}}$  are the transmitter's actual coordinates. By performing this calculation for all position estimates, we created time-varying error vectors for each sub-topology. Since the radial position errors varied over time, we calculated the mean position errors and position error variance for our topologies (as given in Figure 6).

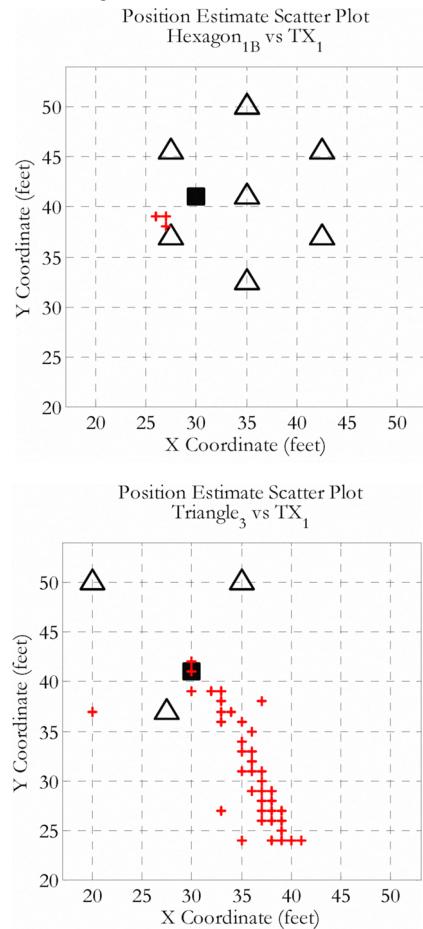


**Figure 6:** Time-varying radial error comparison

By combining these two metrics we were able to make comparisons between our topologies according to how accurate and how consistent they were with their

localization attempts and thus determine the effectiveness of the localization algorithm.

Our data analysis revealed that topologies such as the hexagon (radios 1, 2, 3, 4, 5, 6, 9) and rectangle (radios 1, 3, 4, 5) topologies produced the most accurate and most stable position estimates (best mean: 3.5 ft for Hex<sub>1B</sub> vs TX<sub>2</sub>, best variance: 0.11 ft<sup>2</sup> for Rect<sub>1B</sub> vs TX<sub>2</sub>). Other topologies such as the triangle and lines yielded very inaccurate and unstable position estimates (worst case was the 2, 5, 10 topology with mean: 12 ft, variance: 35 ft<sup>2</sup>). The trends of the topologies' performances were reflected in the number of nodes used to create them as well as their spatial diversity with respect to the transmitter. When viewed as scatter plots, the position estimates for any pair of topologies makes these trends more apparent. Figure 7 offers a comparison between the best and the worst topologies taken against the same transmitter location.

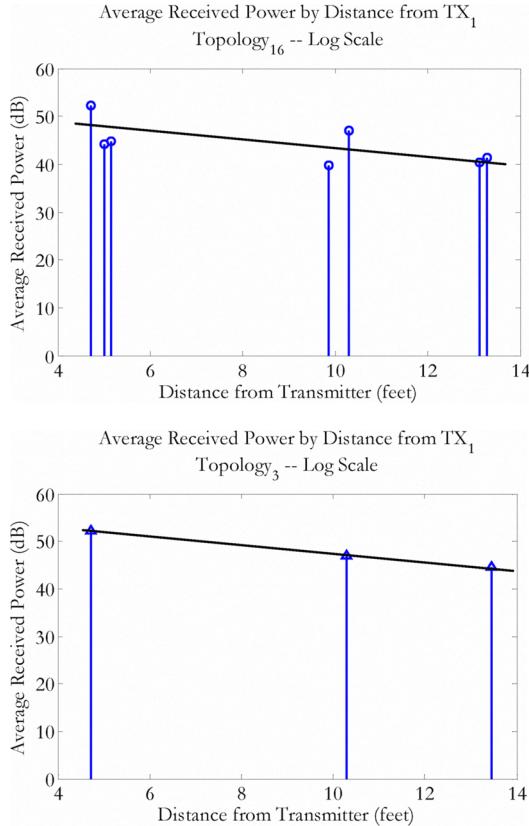


**Figure 7:** Scatter plots of position estimates (shown as +) for the Hexagon<sub>1B</sub> (top) and Triangle<sub>3</sub> (bottom) topologies against TX<sub>1</sub> (61 estimates each)

Figure 7 shows that the Hexagon<sub>1B</sub> topology (top) has a low variance of estimates and the Triangle<sub>3</sub> topology (bottom) has a high variance. The high variance is expected due to the sparseness of the topology. However, two additional things stand out from the triangle's scatter

plot: the position estimates appear to be biases, pointing towards the upper-left sensor (radio 10), and there are at least two position estimate outliers. Investigating the data set revealed that the outliers occurred at the very first and last estimates. We removed these estimates from all data sets when making our topology comparisons.

In order to investigate the scatter plots' shapes we graphed the average received power for the two topologies according to the distance between each of their nodes and the transmitter. Figure 8 illustrates a set of measurements used for an estimate in Figure 7.



**Figure 8:** Received power by distance for Hexagon<sub>1B</sub> (top) and Triangle<sub>3</sub> (bottom) against TX<sub>1</sub>

As a sanity check, we calculated the path loss exponent for the two measurements shown in Figure 8. According to standard path loss equation

$$P_r = P_t - \alpha \log(d/d_0) \quad (7)$$

where  $P_t$  and  $P_r$  are power transmitted and received in dB,  $d$  is the distance between the radios,  $d_0$  is a reference distance and  $\alpha$  is the path loss exponent, we calculated the path loss to be approximately 2.6, a reasonable value. Investigating further, we found that Radio 2 consistently had a larger average received power value than would be expected from either path loss or variation from fading. While the cause for this discrepancy is still undetermined (most likely manufacturing tolerances on the USRP and

our antennas), it does help explain the bias in the Triangle<sub>3</sub> variance shown in Figure 7.

Nonetheless, given the varied shapes and sizes of our topologies and the consistency of their general performance with the simulation results of [4], we were convinced that the ML localization algorithm performed as it was designed to, and that the algorithm could be feasibly implemented using off-the-shelf SDR equipment.

## CONCLUSIONS

The research presented here was unique for two reasons. First, it relied on data collected from both actual hardware experiments and simulations. Unlike typical research in dynamic spectrum access which relies mostly on simulation data, our experiments provided a means of validating our localization process. Second, it demonstrated that collaboration between nodes can feasibly be used to create a more complete map of the RF environment, thereby allowing more effective automated decision-making at the device level.

Despite our success, the research presented here can be extended in several ways. First, directional antennas could be implemented to allow evaluation of the algorithm's performance under more complex RF arrangements. Next, a unified, extensible language could be created to allow the radios to share a common resource map and thereby facilitate more productive collaboration by allowing sensor-to-sensor queries. Finally, additional experiments could be designed to allow the success or failure of a network-centric task (such as sharing time-sensitive data) to be carried out based on the level of collaboration conducted by a sensing network.

## REFERENCES

- [1] J. Mitola, Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio. PhD thesis, Royal Institute of Technology (KTH), 2000.
- [2] C. R. C. M. da Silva, W. C. Headley, J. D. Reed, and Z. Y., "The application of distributed spectrum sensing and available resource maps to cognitive radio systems," 2008.
- [3] S. M. Mishra A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. Of ICC 2006*, 2006, pp. 1658-1663.
- [4] R. K. Martin and R. W. Thomas, "Algorithms and Bounds for Estimating Location, Directionality, and Environmental Parameters of Primary Spectrum Users," submitted to *IEEE Trans. Wireless Comm.*, April 8, 2009.
- [5] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses and N. S. Correal, Locating the nodes: Cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE* 22(4), pp. 54-69.
- [6] Ettus Research. (2008, Nov). Ettus research order page. [Online]. 2009(01/10), pp. 1. Available: <http://www.ettus.com/orderpage.html>
- [7] N. Patwari, A. O. Hero, III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. on Signal Processing*, vol. 51, no. 8, pp. 2137-2148, Aug. 2003.
- [8] Weiss, A. J. "On the Accuracy of a Cellular Location System Based on RSS Measurements," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 6, pp. 1508-1518, Nov. 2003.
- [9] H. Koopapaty, "Barankin Bounds for Position Estimation Using Received Signal Strength Measurements," in *Proc. IEEE Vehicular Technology Conf.*, vol. 5, Milan, Italy, May 2004, pp. 2686-2690.