SensorFusion: Localization and Tracking of Sensors Using RFID

Kenneth G. Bailey*
Georgia Institute of Technology

Brian J. Wiebe†
Oregon State University

Alejandro Uribe‡
Florida International University

Falko Kuester§
Department of Structural Engineering
University of California, San Diego

Abstract

This paper presents a new technique for the identification, spatial referencing and tracking of arbitrary sensors used in large-scale earthquake engineering experiments. Based on a triangulation scheme for radio frequency identification (RFID) tags attached to each sensor, it is possible to simplify and enhance the time-consuming record keeping process, needed to provide critical information about sensor capabilities, placement, wiring and calibration. Multi-plane RFID radar sweeps are used to identify all tags located within the swept out volume, derive their spatial location and to track them over time. A proof-of-concept system is presented in the context of a network centric approach towards information acquisition, processing, visualization, dissemination and curation. A set of test cases are described and discussed identifying the strengths and weaknesses of this approach.

Keywords: Sensor Networks, Large-scale Experiments, Earthquake Engineering, RFID, Tracking

1 Introduction

Earthquake experiments utilize large numbers of sensors to facilitate the collection of data. These sensors have to be placed, identified, logged, calibrated, monitored, and then collected once the experiment is over. During the course of an experiment, sensors might be moved, replaced, or recalibrated. The result is a large collection of sensors exceedingly difficult to track in space and time once hundreds to thousands of sensors are in place. SensorFusion aims at alleviating this problem by pairing a unique radio frequency identification (RFID) tag with each sensor before it is placed. The pairing of an RFID tag with a sensor subsequently allows for the swift identification of its capabilities, past performance and use, through a simple database query based on the unique 128-bit number associated with each RFID tag. In particular for passive tags that are powered remotely and do not need an onboard power source, this means that they can be embedded with the sensor, without requiring any maintenance during its lifetime. Through triangulation of the radio frequency signal emitted by the tag it is also possible to determine its spatial position as well as movement over time. Imagine having to document the sensor layout for a project, searching for a faulty sensor, re-configuring a sensor array or disassembling an experiment and restocking all of the sensors. With this approach each sensor can be uniquely identified and located via a real-time interactive map.

1.1 Background

RFID is a technology used to transmit information via electromagnetic waves also known as radio waves. Using an electromagnetic waveform, an RFID tag can transmit a unique ID and in some instances even auxiliary information associated with it. RFID systems commonly consist of an energizer and a reader component, in combination with an arbitrary number of RFID tags. The energizer emits an electromagnetic field into the target volume, which charges and/or triggers the RFID tag, forcing it to transmit its ID. The reader records this tag, processes it and subsequently relays it to SensorFusion's tracking client for processing, analysis and archival. Tags will communicate with any compatible reader within their range. Each of the tags has a unique ID that can be used by the reader to locate, track, or obtain information concerning an object associated with the tag.

1.2 History

RFID technology is built from a combination of radio broadcasting and radar technology. Radio and radar technology were developed in the 1930s and 1940s. The convergence of these two disciplines occurred on the heels of radar development, although RFID was not patented until 1983 [wikipedia.org]. In 1948, Harry Stockman discovered that radio waves can be used to power a remote transmitter [Stockman 1948]. His discovery became the principle for passive RFID (see Sec 1.3). Early technologies related to RFID were being explored for over 30 years prior to the first patent for RFID. One of the earliest technologies related to RFID was a long-range transponder system of "Identification Friend or Foe" (IFF), built for aircraft in WWII. In the 1960s, inventors used radio frequency to power remote devices and communication by radar beams. In the 1970s radio backscatter technology was developed and passive data transmission was successful through backscattering radio waves. In 1983, RFID was patented and its first commercial application was in Norway in 1987 as a toll collection system. Toll Highways in the United States quickly followed, and RFID was established in industry [Landt 2001]. Today, many organizations with complex asset management challenges are either considering, testing or readily using RFID-based inventory management technology, replacing line of sight techniques such as barcodes.

1.3 Tags

Conceptually, a RFID tag is an energy receptor. The tag will emit a series of radio pulses at a specific frequency when a compatible energizer is in range. These pulses contain encoded information, such as an unique ID and some error checking information. Other information can include industry specific information such as tag type or product cost. When the receiver reads this information, it breaks down the signal amplitude and extracts the information contained therein, much like a Frequency Modulated (FM) radio receiver in a stereo system. The RFID tags themselves can be broken down into two basic types: passive and active.

Passive RFID tags do not require a battery or any other internal source of power because they obtain their power from the electromagnetic energy transmitted from the reader. Low frequency tags (<100 MHz) are usually powered by magnetic induction. An alternating current in the reader coil induces a current in the tag's an-

*e-mail: kbailey7@mail.gatech.edu
†e-mail:wiebeb@onid.orst.edu
‡e-mail:aurib002@fiu.edu
§e-mail:fkuester@ucsd.edu

tenna coil, allowing a charge to be stored in a capacitor, which then can be used to power the tag [Want 2004]. While the electricity is passing through the coil of the tag, this electricity is altering the magnetic field of the energizer. This field change (called modulation) is how the low frequency tag transmits data. The energizer interprets the modulated magnetic field as data. The drawback is that the electric field will only alter the magnetic field of close objects, making the effective range of the tag limited to a few centimeters (See Fig.1).

High frequency tags (`>100 MHz`) obtain their energy using another method called electromagnetic capture, where the tag receives the signal and backscatters some of the incident RF energy to the reader. The reader, using a sensitive receiver, decodes the ID of the tag from the patterns of reflections, expressed as varying amplitude in the received signal [Want 2004]. Passive tags are usually small and portable but have a very limited range making them efficient only when used for short distances, more specifically, between a few centimeters and a few meters depending on the size and the design of the antenna, as well as the chosen radio frequency. Passive RFID operation requires very strong signals from the energizer, and the signal strength returned from the tag is constrained to very low levels by the limited energy [Zhang and Amin 2006].

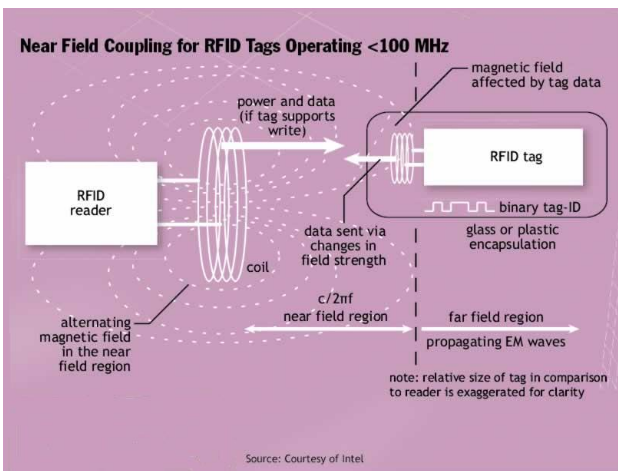

Figure 1: *Magnetic Induction in RFID*
[Want 2004]

Passive tags may have data storage capabilities if they are equipped with memory. The characteristics of this memory is the limiting factor for storing data and retrieving it from the tag. Two basic configurations are available: read-only and read-write. A read-only memory is programmed once to produce a desired data set. A read-write memory has the capabilities to be reprogrammed with different characteristics, if necessary.

One potential use of the passive RFID tag is in retail stores, replacing the printed barcode. The RFID tag will be either embedded in the label or printed on special label stock and will allow checkout to happen automatically by simply reading the RFID signal emitted by the tag when it is brought within range of a reader. In order to be competitive against barcode labels, simple passive RFID tags must cost less then 5 cents per unit. The cost savings in using the RFID tags might not be seen on the product shelves, but in saving merchandise "lost" in a complex transportation network [Santana 2004].

Active RFID tags can be further refined into semi-passive and active. They function under the same concept of RF communication but using their own internal power source. Active tags go beyond the basic functions of passive tags, moving into the capability range of small wireless network nodes. Some of the active tag manufacturers claim their tags can be read at an excess of 500 meters. This longer range is possible because they do not backscatter signals. Instead, they broadcast their own signals that could be extremely strong compared to passive tags . The cost for an active tag is around 4 dollars per unit (for simple tags), and costs are falling [Stanford 2003]. *Semi-Passive RFID tags* or battery-assisted passive tags are considered to be active because an internal source of power is used. However, the battery in semi-passive tags is only used to power the tag's electronics, not to broadcast a RF signal.

1.4 Triangulation

The term triangulation is used to define the method of locating something in relation to a defined reference frame [wikipedia.org]. If the distance from a point (x,y) in space to two reference points (X1,Y1) and (X2,Y2) is known, a triangle can be formed to find the location of point (x,y) in a 2D plane (see Fig. 2). The angles can be found using the law of sines:

$$\frac{a}{sin\,\alpha} = \frac{b}{sin\,\beta} = \frac{c}{sin\,\gamma} \tag{1}$$

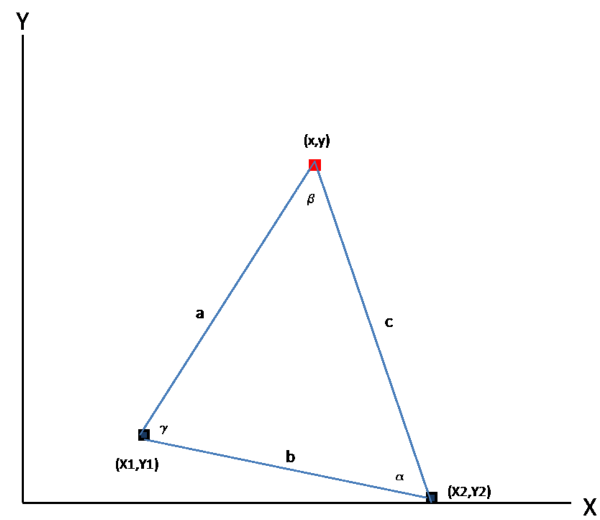

Figure 2: *Law of Sines*

Since (X1,Y1) and (X2,Y2) are points defined in a plane, simple trigonometry can be used to find the specific location of point (x,y) in that same plane.

In order to locate tags in a two-dimensional plane, some RFID systems triangulate the position of the tags using Received Signal Strength Indicators (RSSI). These indicators help a reader to estimate the distance from the tag to the antenna based on the tag's signal strength when it is sampled. One antenna will yield a distance value which can be used to construct a circular estimation of the tag's location. A second antenna is enough to construct a triangle, but will produce two different locations for the unknown distance. For true two-dimensional localization, three antennas are necessary. The third antenna localizes the triangle to one semicircle and provides a very good approximation of the location of the tag. SensorFusion uses the angle of incident and RSSI to locate tags. The angle of incident reduces the number of reader antennas to two, but has a limited detection area (see Fig. 9). Once the reader has acquired information from the target volume, tag beacons are further analyzed. Once processing is completed, the approximate location of identified tags can be visualized.

Another common method of triangulation uses three or more reference points (similar to how cell phone locations are determined). Each antenna samples the signal (using the RSSI) and from that sample, the radius of a circle is created. Each antenna in range samples the signal and creates a location circle. Where *all* these circles intersect is the location of the tag. A minimum of three readers is necessary for this form of triangulation. However, all RFID tags must be in range of at least three readers, possibly requiring the area to be "blanketed" with additional readers, increasing overall system complexity.

A major drawback with most triangulation systems is the RSSI. The signals emitted by passive and active tags are very sensitive to the surrounding environment. Metal items that reflect the signal, wooden items that absorb the signal, or other electromagnetic signals can have an extreme effect on the received signal (see Sec 4.1.1). Even the orientation of the tag's antenna (i.e. facing away from the reader) can have significant effects on the signal strength. These effects can create false readings and even make the tag appear to be very far away from one reader, while very close to another reader.

2 Project Focus

In this proof-of-concept study, the feasibility of successfully pairing an RIFD tag with an engineering simulation sensor, detecting that RFID tag, locating it, and storing the information in a central database and subsequent visualization was explored. The core systems components can be characterized as (1) RFID identification and tracking, (2) data conversions and referencing, (3) centralized storage and (4) interactive visualization. Processing, storage and visualization, were realized in the context of the NEESCentral a cyberinfrastructure system, developed by the George E. Brown, Jr. Network for Earthquake Engineering Simulation Cyberinfrastructure Center (NEESit) (http://it.nees.org).

2.1 RFID Radar System

Different from Time Difference of Arrival algorithms, this system reader uses the two antenna to detect the angle of arrival of the tag's signal and the degradation of the signal to determine the tag's distance from the reader. This produces a set of Polar Coordinates to localize the tag in the X-Y plane (see Sec 1.4).

The operating frequency has been set to the preferred frequency for the U.S. and should not interfere with other radio system users such as cell phone networks. In the US the operating frequency is in the 902 to 930MHz band [Marsh 2007]. The system uses 1kHz of spectrum at this frequency. Due to its narrow operating frequency bandwidth, two radars can be set up within 4 meters of each other.

The test system included twelve (12) Claymore Long-range Tags (active), five (5) Stick Tags (active), and fifteen (15) EcochipTags (passive). Tag specifications are shown in Table 1.

2.1.1 Modifications

The highly likely three-dimensional distribution of sensors requires an RFID system with sufficient degrees-of-freedom to detect sensors in all three dimensions. In an enclosed space such as a small room, tags can be clustered to a specific location and can be easily viewed regardless of their vertical orientation. In a large, open space, such as a shake table, this would be ineffective. The first modification will be to sweep the radar through an arbitrary number of degrees. A benchmark of 180 degrees was chosen as the radar antenna could be mounted on a wall and the array could sweep an entire room. With a sixty-degree detection arc, it was decided that moving the array thirty degrees and then detecting the tags would cover any holes that subsequent, side-by-side sixty-degree cones would create.

Table 1: *TrolleyScan Tag Specs*

	Ecochip	Stick	Claymore
Type	Passive	Active	Active
Range(m)	10	30	40
Size(mm)	86X44X.5	160X12X3	180X30X30
Op. Power(μW)	200	5	0.6
Antenna Gain(dB)	2.1	2.1	5.0

[Marsh 2007]

The second modification to the array would be to rotate the horizontal array 90 degrees about the X-axis so that the array would be vertically positioned. This would convert the RFID system from detecting sensors on the X-Y plane to detecting sensors on the Y-Z plane (see Fig. 3). Combining this information would essentially provide three-dimensional tag locations. Further modifications to allow the radar antenna to sweep through the Y-Z plane would allow the RFID system to detect tags in the "gaps" of the coverage of the detection arc.

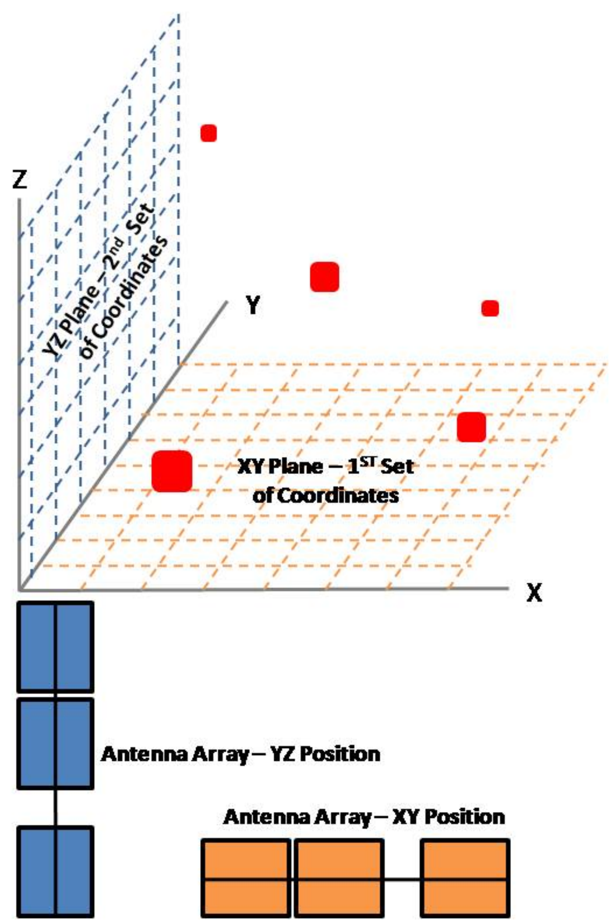

Figure 3: *Antenna rotation for 3D localization*

Ease of field deployment and use of the RFID system itself is a critical attribute. Long cables between experiments must be supported and an RS232 communications system will support this. The chaotic nature of the experiment environment also necessitates a robust communications protocol. This protocol should inform the user (or a connected program) if the reader system is not communicating with the software. The RFID system's RS232 cable will be connected to a Lantronix UDS-100-01 RS232-to-ethernet translator. The translator allows the radar system to be plugged into a simple Ethernet hub and controlled from any computer with the appropriate software (described below). For controlling the positioning electronics, a second Lantronix UDS-100-01 box will be connected to the serial interface of the positioning system for the same purpose. The Lantronix box acts as a miniature Ethernet card, supporting its own IP Address and Port. More importantly, if the RFID system loses connection or is shut off or unplugged from the network, the TCP/IP protocol immediately notifies the connecting applications that the device is not connected.

2.2 SensorFusion

The secondary tool of this project is the software designed to merge the information from the RFID reader into true 3-Dimensional data. In order to keep careful control of the data, the software needs to control the RFID system. Without actually having the RFID system, the software was difficult to design and implement. A software emulator was initially developed and used to provide controllable synthetic input to the processing and visualization stages. The primary obstacle in the software was determining how the radar array was positioned and whether the array was in horizontal or vertical position (flipped or not). It was decided then that the software, in addition to merging the sensor data, would also control the electronics and motors used to position the array.

The flow of the program is as follows:

1. Upon startup the program attempts to connect to the supplied IP address. (If a connection is not available the program aborts)
2. Pressing the start button starts the program.
3. Once started, the program sends an ON message to the emulator or the array instructing the array to energize and prepare to transmit data.
4. After turning on the array, the program sends the array the RADAR command, instructing the array to locate all the tags in its "field of vision".
5. The program waits for 30 seconds. During this time, the array is finding all tags in range and obtaining their information.
6. After 30 seconds have expired, the program sends a QUIET command, instructing the array to stop sending data and data processing begins.
7. At this time, the program repositions the antenna array and the process repeats from step 4.
8. A value of Sweep Angle and Sweep Distance (defaulting to 180 and 30 degrees respectively) determine how far the array rotates and how many times the array moves.
9. Once it rotates through Sweep Angle, the software instructs the hardware to "flip" the antenna array and start over, this time recording the information as if it were on the Y-Z plane.

The software itself uses the original position of the array as a starting point for determining where the tags are located. Keeping track of the pulses from the original 0 position, the software adds Sweep Distance times the Pulse value to the information return from the radar array, unless the array is "flipped", in which case the rotated value is ignored.

The data generated by the software will be stored in a central database repository alongside the sensor information and collected data. NEESit maintains such a centralized database in NEEScentral (https://central.nees.org). The information will be integrated directly with this organization's existing tools (see Fig.4 and 5).

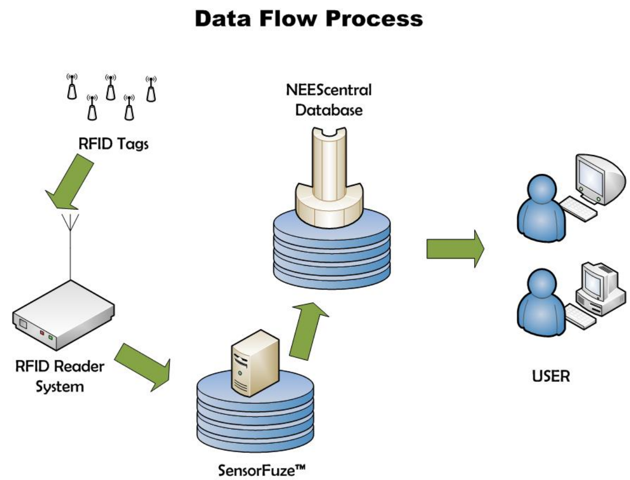

Figure 4: *SensorFusion Data Flow*

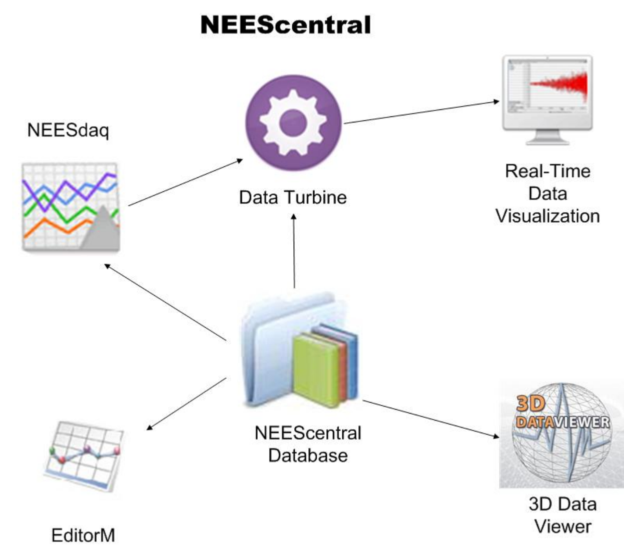

Figure 5: *NEEScentral Tools that can be integrated with SensorFusion*

3 Related Work

Similar RFID location and tracking concepts have been researched in the past. One approach was the 3D-iD system which is the equivalent of a GPS for a location fixed by boundaries (for example a building) but using RFID tags instead [Werb and Lanzi 1998]. The idea of mapping and localizing sets of RFID tags using a moving antenna in the form of a robot has also been approached [Hannel et al. 2004]. The LANDMARC RFID location prototype system uses reference active tags to improve the accuracy of the coordinates [Ni et al. 2005]. Another paper examines the applicability of direction-of-arrival(DOA) estimation methods to the localization and tracking problems of passive RFID tags. [Zhang and Amin 2006].

4 Technology

4.1 Challenges

4.1.1 RFID

Radio Waves have an ideal behavior when they travel through air. When radio waves encounter materials such as metals, concrete, or wood they behave in a different manner. The behavior of the waves depends on the dielectric constant (k) of the material involved. The dielectric constant of an object is the ratio of the ability of the material to carry alternating current to the ability of vacuum to carry alternating current. When a radio wave is traveling through air and encounters a different medium (object with a different dielectric constant) some of the energy is reflected. However, the object has to be of a certain size compared to the wavelength of the signal (32 cm at 915 MHz), say at least 1 cm^3, in order to have a notable impact on the behavior of a radio wave [Marsh 2004]. In the case of passive RFID systems, if some of the energy sent by the reader is reflected it means that the amount of energy received by the tag will be less than expected in optimal conditions (in air). In other words, the sensitivity of the tag will be smaller. Sensitivity is the received RF power necessary to turn on the RFID tag [Marsh 2004]. Smaller sensitivity means a longer range.

Some RFID tags used for SensorFusion will be attached to sensors that have enclosures made from aluminum. In theory, at any metallic surface, a signal is reflected from the metal back towards the reader, and this signal has the opposite phase angle to the incoming signal, which cancels the incoming signal at the surface (See Fig. 6) [Marsh 2004]. When a small separation exists between the tag and the metallic surface, the tag should be able to receive the signal but the signal will still be affected either negatively or positively by this situation. It has been found that a separation of approximately 8 mm between the tag and the surface will make the tag perform as if it was working in optimal conditions (free air). Any smaller separation will require transmitting more power from the reader the tag, but a bigger separation of up to approximately 24 mm will require transmitting less power from the reader, improving the range of the tag by about 40% [Marsh 2004]. In order to isolate the tags from the metal, a material with a dielectric constant similar to air `(k~1)` has to be used. Some of the common materials used for this purpose are bubble wrap, polyurethane or polystyrene. In addition, the piece of material used has to be more than 8 mm thick in order to optimize the energy consumption of the RFID tag.

4.1.2 Software Design

As mentioned earlier, this project is broken down into many phases. The first phase of this project is to design and construct a working prototype of the SensorFuze computer program in the allotted 10 weeks of time. A large, complicated program would be detrimental to the project and would make the project much less feasible. Designing this software posed many problems, listed here:

1. No hardware was or would be available for testing until late in the project.
2. Which language would be best to use?
3. How should the control of the radar be implemented?

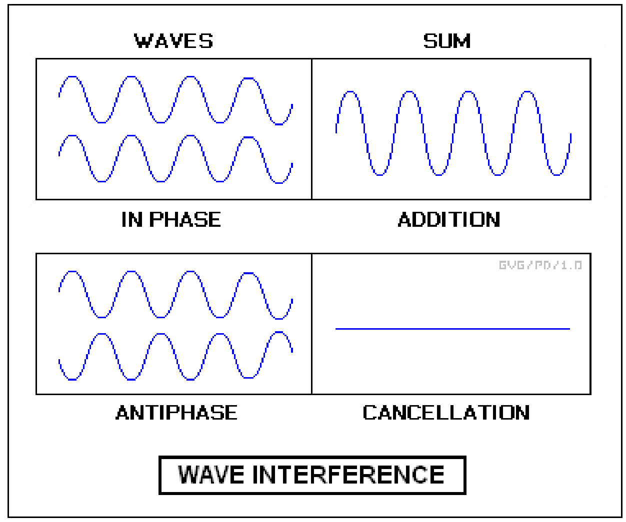

Figure 6: *Electromagnetic Wave Behavior: Addition and Cancelation*

[vectorsite.net]

4. How would we view (i.e. justify) the converted data?
   (a) Would we have time to integrate everyting into NEESit?
5. The radar system also has many flaws
   (a) How do we get 3-Dimensional data?
   (b) How do we overcome the 60 degree detection cone limitation?
   (c) Can we comunicate without using COM and RS232 protocols?

The lack of hardware was a major obstacle to overcome. Without hardware, there would be no way to test the new software program. Rather than wait for the hardware to be ordered and received, a plan was implemented using the TrolleyScan Radar System documentation. An emulator mimicking the TrolleyScan Radar System was created. Not only did this emulator provide a means of testing the software but it also provided a test platform for the programmers to tackle technical issues with the preferred compiler as well as relearn any lessons that had been forgotten. The emulator raised an additional issue, however. How robust would this emulator be? As a throw away testing tool, does it need to have advanced error detection and memory management? In the end, form and function was preferred over robustness. Where error checking provided a means to track down a bug, that code was put in place.

Design of the emulator introduced even more questions, but these were specific to the design process. Questions such as:

- Should the program be multithreaded?
- Should the program be a client or a server?
- What type of interface should the program have?
- How object oriented should the program be?
- Should the emulator emulate the changes to be made to the TrolleyScan Radar System?

All of these questions were answered. Multithreading was determined to be a necessity, otherwise the interface would hang while receiving data. The server sockets generated problems (such as waiting for data to be received) that were quickly resolved (and those solutions were implemented in the SensorFuze project). The interface chosen was a dialog box (see Fig. 7). and every major

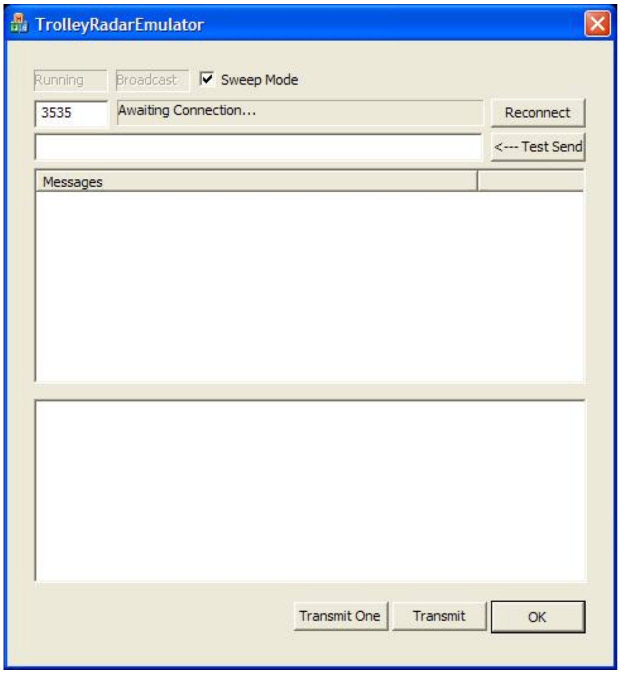

Figure 7: *TrolleyScan Radar Emulator Interface*

object was a class, including the threads for handling socket connections. This should allow these items to be removed and replaced with different objects if necessary. Finally, the initial emulator was designed to just generate data and work through multithreading and socket issues. After resolving the programming and architecture issues, the emulator was upgraded to perform like the modified radar array would. A prototype for the emulator took shape in a scant two weeks, allowing the SensorFuze program design to begin.

The design process for SensorFuze posed another major issue. How would the radar system be repositioned? As mentioned earlier, the RFID system is to be modified so that the radar will sweep through a 180 degree arc and flip over to cover the YZ plane as well as the XY plane. Initially, the design for this hardware incorporated a circular plate with holes drilled in it, a dc motor, an LED to tell the motor when to the stop, and a Integrated Circuit (IC) timer to determine when to move the radar. As the radar moved, a pulse would be transmitted to the computer indicating the array was in position. The problem with controlling the rotation through the IC was how to synchronize the data and what would happen if the system got out of sync.

The solution to this problem was in having SenzorFuze control the repositioning of the radar array. Instead of a plate and LED to control the turning radius, a servo motor was incorporated into the design, which can be controlled by pulses sent from the control program. Now, instead of having to synch with the pulses from the radar array, SensorFuze instructs the radar system to reposition itself, waits for confirmation of the new position, and then reactivates the array to collect data.

The final goal of this program is to integrate the sensor location data into NEEScentral for storage alongside the sensor information. When the sensor information is retrieved, the sensor's tag ID and location in three dimensional space would also be provided. Again, the lack of hardware was hurdle that was difficult to jump. Rather than scrap the idea of integration, the 3D DataViewer program, developed at the Rensselaer Polytechnic Institute (https://nees.rpi.edu/3dviewer) was chosen as a way to test SensorFuze's construction of the data and the feasibility of integrating this data into a third party tool. The third party tool also provided a nice visualization of the test data (see Fig. 8).

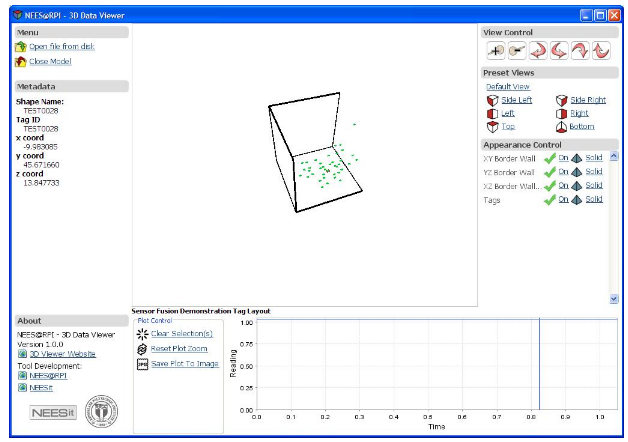

Figure 8: *RPI's 3D DataViewer reading a model file generated by SensorFuze*

Even though the radar system modifications might seem to be out of the scope of this section, these modifications provided key answers to the design and implementation of SensorFuze. The 3-Dimensional data problem also led to the design requirement that SensorFuze control the rotation of the radar array. If something happened and the array lost synchronization with the software, this would lead to all the tag information being incorrect and therefore useless. Since the idea was to make this system stand alone and automated, there might be a point where no one is watching the program and the array to determine if the array is out-of-sync with the program. The 60-degree cone limitation (see Fig. 9) raised issues concerning synchronization. This led to a pulse count design, allowing SensorFuze to keep track of exactly which face the radar system was oriented towards.

The last problem, the RS232 serial protocol led to another design issue. Serial communication is old and reliable, but programmitically communicating with a serial port is becoming more and more difficult as compilors move towards TCP/IP and socket programming. This protocol is more robust and able to detect disconnections much easier than a serial port. The design decision to pair the radar array with a Lantronix box to translate the RS232 data to ethernet packets allows SensorFuze to make use of Windows Sockets.

5 Testing

5.1 Software

Once SensorFuze was completed and all the minor issues (such as converting the sockets to non-blocking sockets and dealing with memory issues when passing messages) were taken care of, the software had to be tested. Originally, the software was tested with the emulator generating 50 random tags. The location of these tags was passed to SensorFuze and a location for each tag was calculated. Further tests were needed, however, to test the remaining functionality of the program. To conduct these tests, the emulator was further modified to emulate the sweeping and flipping motion of the radar and test data was generated so that the emulator would

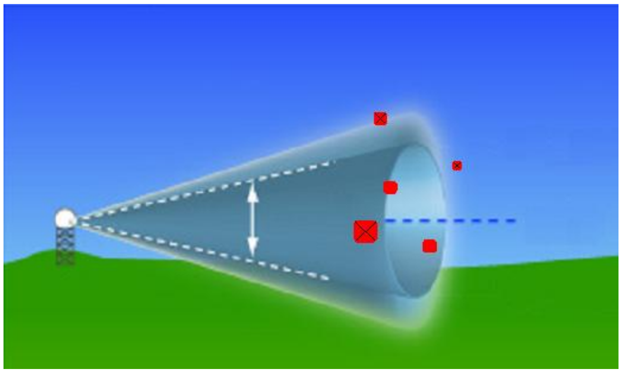

Figure 9: *Detection Cone - Sensors with an X through them cannot be reliably located.*

broadcast “known” locations of tags in polar coordinates. SensorFuze would read this data and calculate the tag’s cartesian coordinates.

The following table (Table 2) shows a snippet of the original data that was created to test the array. The data was separated into 6 zones that the emulator would “sweep” through. If the tag was in the current zone, that tag information would be sent to SensorFuze for processing.

Table 2: *Sample of test data for the emulator*

Dis	Hor Angle	Ver Angle	X	Y	Z
30	-17.5	-10	28.612	-9.022	-5.209
50	-5	36.5	49.810	-4.358	29.741
28.5	7.25	13.6	28.272	3.597	6.702
47.5	14.95	-13.3	45.892	12.254	-10.927
12.75	42.35	-10	9.423	8.589	-2.214
23.27	40.29	11.5	17.750	15.048	4.639
27.93	54.33	-27.93	16.286	22.690	-13.082

The first test highlighted a flaw in SensorFuze’s ability to calculate cartesian coordinate locations. The program was not taking into account the position of the radar array. As a result, all the tags were located in the first arc of detection. The program was modified to include the sweep angle and pulse count when calculating the tag’s coordinates.

Second and subsequent tests have shown the most of the tags appear to be located properly. As shown in Figure 8, the data even closely resembles the original, hand-drawn data. In order to spot check the tested data, a test file was generated with all the tag information. This test file was generated at the same time as the xml file and a small portion of the file (showing only the id, x, y, and z values) is shown in Table 3.

Table 3: *Portion of test data generated by sensor fuze*

ID	X	Y	Z
TEST0035	-28.002	32.961	7.510
TEST0024	-11.207	38.398	-0.175
TEST0013	5.024	37.011	-8.573
TEST0002	28.272	3.597	6.702
TEST0036	-36.459	23.451	-9.102
TEST0025	-12.689	43.476	-0.198
TEST0014	9.730	36.288	8.048
TEST0003	45.892	12.254	-10.927

The conversion of the polar coordinates to cartesian coordinates was done with simple geometric equations for completing triangles. The polar coordinates for each two dimensional representation can be thought of as an angle and hypothenuse of a triangle (see Fig. 10). With the angle and the hypothenuse length, both X and Y can be found using the trigonometric ratios shown in equations (2) and (3).

$$sin\ \theta = \frac{opp}{hyp} \tag{2}$$

$$cos\ \theta = \frac{adj}{hyp} \tag{3}$$

$$Y = R * sin\ \theta \tag{4}$$

$$X = R * cos\ \theta \tag{5}$$

Substituting X for adj, Y for opp, and R for hyp, equations (2) and (3) are solved for the unknown variable yielding equations (4) and (5). These equations produce a 2-dimensional location in the cartesian coordinate system. With a supplied vertical angle, the same formulas are used to produce a Z component, except in this instance, only the sin component needs to be calculated.

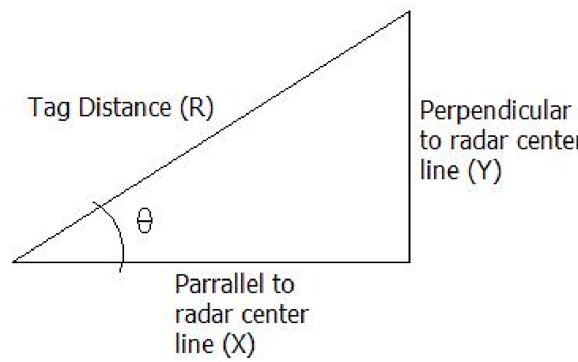

Figure 10: *Each set of polar coordinates can form a triangle, with X and Y being unknown.*

5.2 TrolleyScan Radar System

5.2.1 Assembly

The arrival of the TrolleyScan Radar System heralded a change in the focus of the project. With hardware now available, extensive testing of the software and the limits of this RFID system could now begin. The first step was the assembly of the radar array. Three antennas were attached to six aluminum cross bars that were bolted together to form a horizontal array. The energizer was placed on the right side (when facing the broadcast direction) while the two detectors were placed on the left side of the crossbars (see Fig. 11).

Each antenna is appropriately marked with a “left”, “center”, or “right” label, signifying the antenna’s location on the crossbeam assembly. These labels also signify where the antennas are plugged into the back of the box. Three cables, also marked, plug into the antennas and the box providing communication between the array and the Range Scanner. Care was taken to make sure the Range Scanner box was not powered on or plugged into a power outlet. The Range Scanner requires the energizer array (the three antennas) to be plugged in to dissipate any energy caused by turning on the box [Marsh 2007].

Figure 11: *The assembled array. Note the asymmetrical assembly of the antennas.*

Once the antenna array was assembled and connected to the Range Scanner box, the radar system was connected to the assembled XPC Shuttle computer (see Fig. 12) via serial cable. At this point, the goal was to check the array and calibrate it using the supplied program (as shown in Fig. 13). Once the radar system was attached to the computer, the device was powered on and it immediately started picking up the few passive tags that were set beside the antenna array. This initially confused the group as the provided program had not been used to turn "on" the scanner. This detection was due to the initial energy discharge the radar makes when powering up.

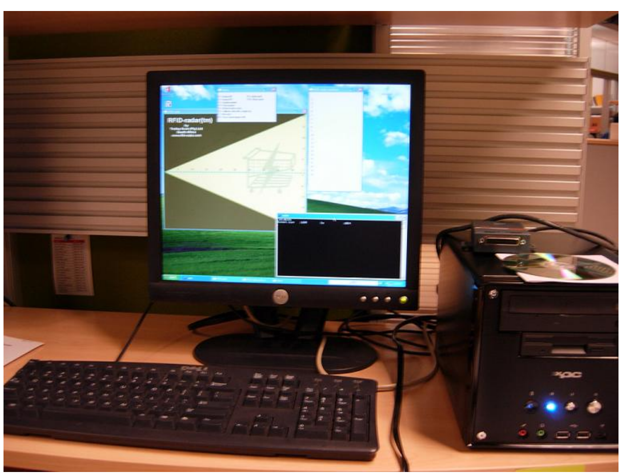

Figure 12: *The XPC Shuttle computer to run the software that controls the radar.*

5.2.2 Calibration

The environment used to test the system was an office where a little interference could exist. The final goal of the project is to be able to use RFID tags in complex environments where interference can

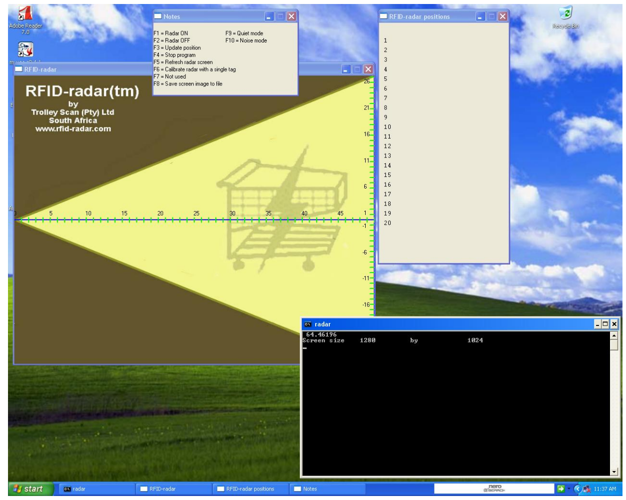

Figure 13: *Screen shot of the supplied software used to calibrate the array*

be an issue. By choosing this place for testing, the real limitations of the system can be observed closely.

Once assembled, the radar system had to be calibrated. The first attempt of calibration was a dismal failure as the radar system had trouble initially even detecting a tag. Calibration requires that an active tag be placed 9 meters away and directly in front of the array, about 4 feet off the ground. Once in place, the radar system needs to detect the tag. When the tag is first detected, the range is set to the default of 60 meters and displayed to the screen [Marsh 2007]. The calibrate command was sent to the radar system and after several seconds, the system returned a range value with the tag. Only the range was 17.5 meters, much farther away than the measured 9 meters where the tag was located. To further complicate matters, after a few broadcasts, the range of the tag returned to 60 meters.

Several different active tags were placed at the 9 meter position, to discard the possibility of a malfunctioning tag, and the machine was reset and recalibrated several times. The recalibration results were always different but never at the 9 meter range. The closest calibration value was 9.25 meters and after several minutes of broadcasting, the radar system started reporting the tag at closer range, even though the tag had not been moved.

The next step was to refer to the documentation of the radar system to try to find recommendations for calibration. The documentation provided helpful information on how to correctly calibrate the system. The inaccuracy was caused by the antenna array being resting on a surface. The solution to this problem was to design and build a mount that will hold the antenna 4-5 ft above the ground. The design and fabrication of the mount would serve an additional purpose as well. The various planned control electronics could be attached to this mount when that portion of the project was implemented.

The mount for the antenna was designed in such a way that interference could be avoided as much as possible. A camera tripod was modified to hold the antenna array with the center of the array being approximately 5 feet above the ground. The mount also has the required degrees-of-freedom necessary for the system to sweep through a detection zone. The antenna can be rotated horizontally and vertically as necessary. However, design and manufacture of this piece of equipment delayed the process a few days.

In order to test the RFID system on a different setting (while the mount was fabricated), the antenna was mounted on a vertical surface (similar to hanging it on a wall) and calibrated. Even though better results were expected, the first calibration attempt after the new positioning of the antenna array wasn't successful. The data expected from the system was a 0-degree angle and a 9-meter range. The angle obtained was exact but the range was found to be 13.5 m. A second calibration attempt was made obtaining more acceptable results: 8.67 m and 0° (see Sec. 6.1).

5.3 Multiple Tag Test 1

After successfully calibrating one tag, the next test was to test the TrolleyScan Radar System with several tags of different types. The goal of this test was to determine how accurately the different tags were reported and whether or not the addition of new tags would affect the speed of acquisition. For this test, the initial calibration tag was left in place to provide a base comparison for newly acquired tags. Initially, only active claymore tags were placed in the energizer field, but an ecochip and stick tag were added to complete the test. Three additional claymore tags, one stick tag, and one ecochip tag were placed in the energizing field. Once these tags were placed, the radar system was given enough time to acquire the tags and the data was taken from the demonstration program (supplied with the hardware) and graphed (see Sec. 6.2).

5.4 Multiple, Single Type Tag Tests

After the results from the calibration and the previous test were tabulated, several problems with the array seemed to surface. In an attempt to isolate these issues, the next three tests were conducted in a different direction using only a single type of tag each time. Five tags were placed at measured distances and a test was run on each set. Also, as a result of the previous tests, more than one minute of data (the total time stored by the demonstration program) needed to be collected. The theory was that the longer the system had to run, the more accurate the readings would become. For longer readings, SensorFuze would have to connected to the radar system and the data would have to be written to disk. Each test was run for one hour and uncovered several bugs in the program leading to the data from the first set of tests being completly inaccurate. A second set of tests were run after the software was corrected. With the wild swings in distance reported from the previous tests, it was decided that the following tests would focus on the reported distance, with angle data being tabulated when and if the distances could accurately reported. The results for these tests are shown in Section 6.3.

5.4.1 Claymore (Active)

The first test conducted was with the active, Claymore tags. These tags were the most accurate from the first tests and would probably provide a decent benchmark for the remaining tags. Each tag was placed no more than five meters away from the array, facing the general direction of the antenna array. The distance from the tag to the array was carefully measured and labled at the tag location, so the other tags could be placed accurately. Once the tags were placed, the antenna was turned on, the data was recorded to disk, and the data was plotted.

5.4.2 Stick (Active)

The second test involved removing and deactivating the Claymore tags and replacing the Claymore tags with the active Stick tags. These tags were placed as close to the marked locations as possible to make this an accurate test. Once the tags were activated, the software recorded the data and the distance values were plotted.

5.4.3 Ecochip (Passive)

The final portion of this test was to use the passive, Ecochip tags. These tags are supposed to function best when placed closest to the reader. Each stick tag from the previous test was removed and deactivated and then replaced with a passive tag. The radar array was then turned on and allowed to acquire the tags before the data was written to disk and plotted.

5.5 Outside Test

With the results of the last several tests showing a lack of distance accuracy, the group decided to contact the manufacturer of the RFID system. His response seemed to focus on the fact that the radar system might be having trouble distinguishing true signals from reflected rignals. Armed with this information, the TrolleyScan Radar System was taken outside for testing in as much of an interference free zone as possible. Due to power plug constraints, the TrolleyScan Radar System was again hooked to the shuttle computer and the demonstration program was run. The system was recalibrated and several tags were introduced one at a time, with the calibration tag remaing as a benchmark. First, an active tag was introduced to the energizer field and then removed. Then two stick tags were introduced and removed. After testing the stick tags, one group member was given an active Claymore tag and sent through the field, never stopping at one point. After the motion test, two more Claymore tags were placed as far away as they could be detected in the energizor field. Finally, two passive tags and one stick tag were placed near the array at different angles.

5.6 Line-of-Sight, Multiple-Tag Test

As a last test, the group chose to test the TrolleyScan Radar System's ability to detect tags that were hidden or not directly visible by the TrolleyScan Radar System. One of the primary requirements of the project is to detect tags in earthquake simulations and sometimes these tags might be buried in concrete or placed behind obstacles. This test would determine if these obstacles would prevent the radar from detecting hidden tags. After the results of the Outside Test, the stick tags were dropped from the testing regime as being too unreliable. Seven tags were used for this test, three active and four passive. Three passive tags were placed behind obstacles and one was placed in sight of the array. Two of the active tags were activated and "hidden" from the radar while a third was activated and placed nearly nine meters away from the radar. Thirty minutes of data was collected and the test was cut short when it was discovered that the insight passive and active tags were hardly being detected and hidden passive tags were never detected at all.

6 Results

6.1 Calibration Test

The following figures (see Fig. 14 and Fig. 15) represent the results from the first *successful* calibration. The original calibration data was not recorded. Figure 14 shows the reported distance of the tag from the antenna along with the actual distance of the tag from the reader. After several attempts, the readings became stable with little fluctuations in the reported distance. The data set in Figure 14 shows the information gathered for 80 seconds, roughly the maximum amount of data able to be copied from the demonstration program.

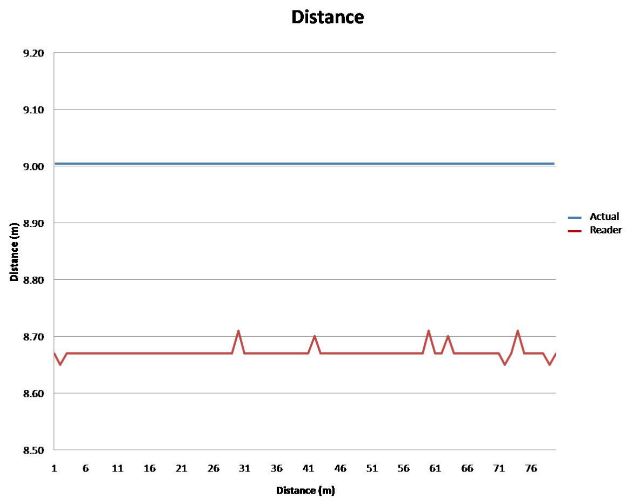

Figure 14: *Actual and recorded distance*

Figure 15 shows the angle as reported by the demonstration software. This information is somewhat more volatile than the distance readings, especially early in the recorded cycle. After several seconds, the reported position began to ocillate between the actual value (in blue) and 0.5 degrees. Remember that the actual position of this tag was directly in front of the radar at 0 degrees.

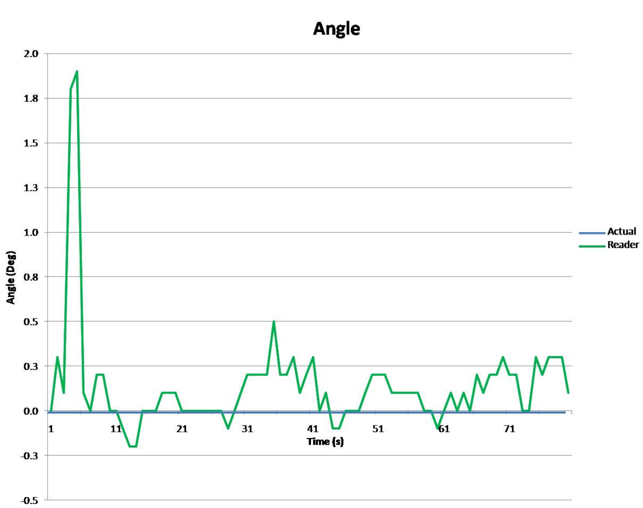

Figure 15: *Actual and recorded angle*

From this recorded data, the average distance and angle were calculated along with a percent error. The average distance obtained from the reader in 80 seconds was 8.6715 m for a 3.65% error. The average angle recorded was 0.1314 for a 0.8% error. Table 4 illustrates these results.

Table 4: *Calibration Test Results*

	Actual	Radar	% error
Distance (m)	9	8.6715	3.65
Angle (°)	0	0.1314	0.8

6.2 Multiple Tag Test 1

The results of this test began to show some possibility of problems with the TrolleyScan Radar System. As shown in Figure 16, the Ecochip tag (id BBBF73425), was initially reported close to the actual value of 1.91 meters. However, the tags position seemed to jump up to 60 meters and stay there. Towards the end of the test data, the tag was reported at point blank range or 0 meters in front of the array.

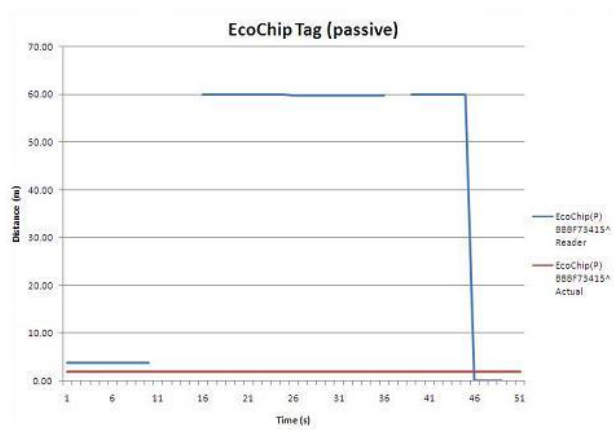

Figure 16: *Distance values reported over 60 seconds*

The second graph (see Fig. 17) shows the results of the distance recorded for the stick tag (id BCBBB4645). The results show that the distance recorded remained relatively consistent (as opposed to the the previous ecochip tag), but the recorded value was almost two meters away from the actual value. Also notable was the jump towards the end of the graphed data, where the tag seemed to "drift" even farther away from the antenna array.

The next four graphs show the plotted values of distances recorded for the Claymore active tags. As can be seen in Figures 18, 19, and 20, the active tags appear to be reported much more accuratly. The tag in Figure 21 is reported much farther away. Again, note the distance change toward the end of the graph, where the tag appears to "drift". In two of the three cases, this drift made the readings more accurate, but in the third case (Fig. 21), the reported values were not that accurate and the new values are just as inaccurate.

As can be seen in the graphs, the active tag distances appear to be consistently reported, but as can be seen in Table 5, the error percentages of all but one tag are more than 5%. While more than good enough for a warehouse where products can be several meters in size, a distance difference of 0.5 meters can include several sensors. Angle values were not recorded as the group focused on the distance values.

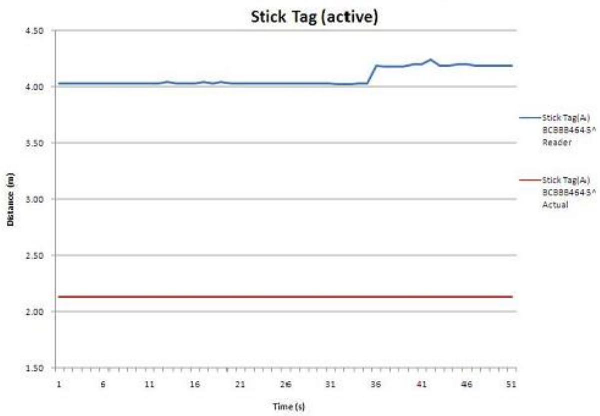

Figure 17: *Distance values for the stick tag. Note the jump towards the end of the graph*

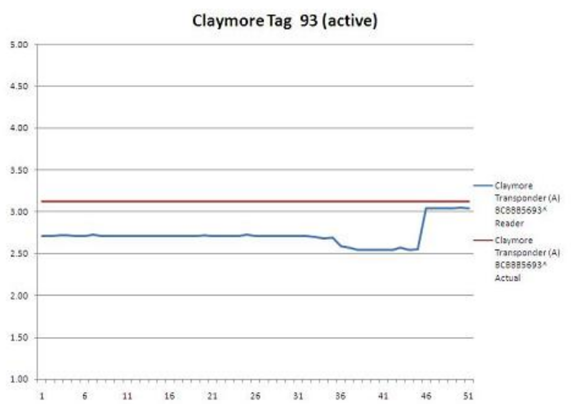

Figure 18: *Distance value sfor Active Tag 93*

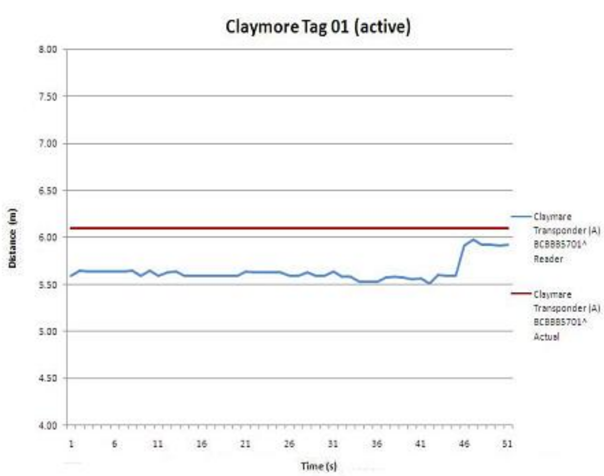

Figure 19: *Distance values for Active Tag 01*

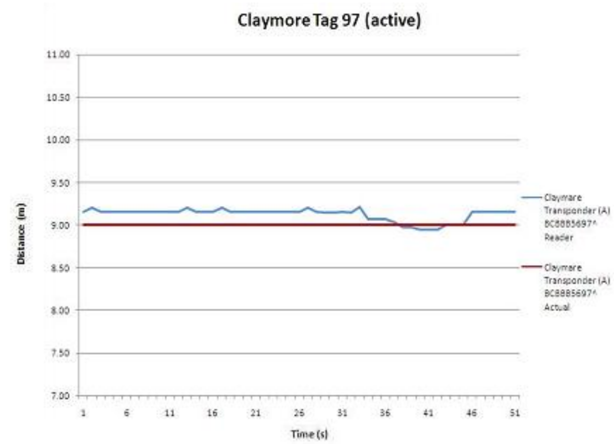

Figure 20: *Distance values for Active Tag 97 (the calibration tag)*

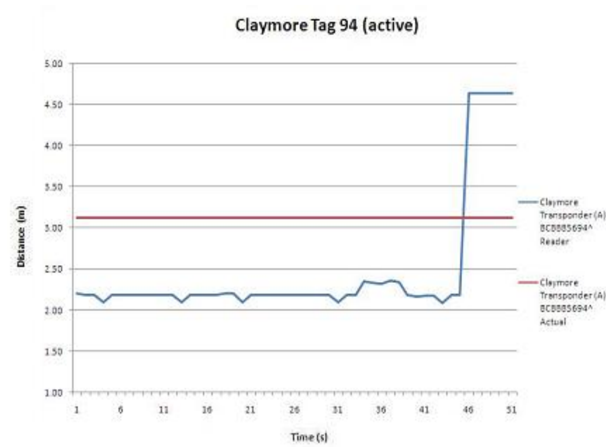

Figure 21: *Distance values for Acitve Tag 94*

Table 5: *Mulitple Tag Test 1 Results*

Tag	Actual Distance(m)	Avg Distance(m)	% error
Ecochip	1.91	40.84	2038.22
Stick	2.13	4.82	126.29
Claymore 93	3.12	2.72	12.82
Claymore 94	3.12	2.48	20.51
Claymore 97	9.00	9.13	1.44
Claymore 01	6.10	5.64	7.54

6.3 Multiple, Single Type Tag Test

6.3.1 Claymore (Active)

The following graphs (Fig. 22, 23, 24, 25, and 26) are the plotted results of the five Claymore Active tags. Note that in this test, the sample time has increaed from 1 minute to 1 hour of data. The first Active Tag (see Fig. 22) clearly occilates from values being to close to the array to values being to far from the array. For several minutes, the data seems to be accurately reported, but then the tag is reported closer to the array that it actually is. The spike near the 12-minute mark is a point when the tag was lost repeatedly for several minutes. From the documentation included with the TrolleyScan Radar System, when a tag is lost, it is reacquired and the range has to be calculated again. Since the calculation time is only 20 seconds, this spike clearly shows that the tag was lost and found repeatedly.

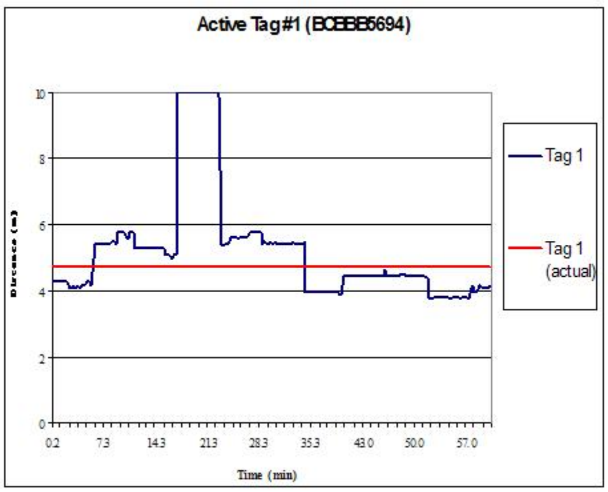

Figure 22: *Plotted values for the first active tag. Note the spike where the tag was lost for several minutes.*

The results for the second tag are shown in Figure 23. The plotted data shows the tag to be inaccurately reported almost the entire time, save for a brief instance. In this case, the longer reported time did not result in more accurate results. The most notable of this plotted data is the small spike at the 21-minute mark where the tag seemed to be accurately reported.

Of all five tags recorded during this test, the third tag (shown in Fig. 24) was the most accurately detected. Although the detected values ocillated between 4 and 6 meters, the average detected location was only 0.2 meters away from the actual location. However, the fact that the tag's position was still ocillating was cause for concern.

Tag number four was the most consistent of the five tags plotted. The average detected distance value seemed to hover around the 4 meter mark. This value, however, was still almost a full meter away from the tag's actual position. The plotted data is shown in Figure 25.

Finally, tag number five seemed to have the greatest variation in reported distances. The tag was rarely even closely reported to its actual position and was lost at least twice for several minutes. Like the ecochip tag from the first multiple tag test, this tag was reported (not once, but twice) as being right next to the radar array. The distance, plotted versus time, is shown in Figure 26.

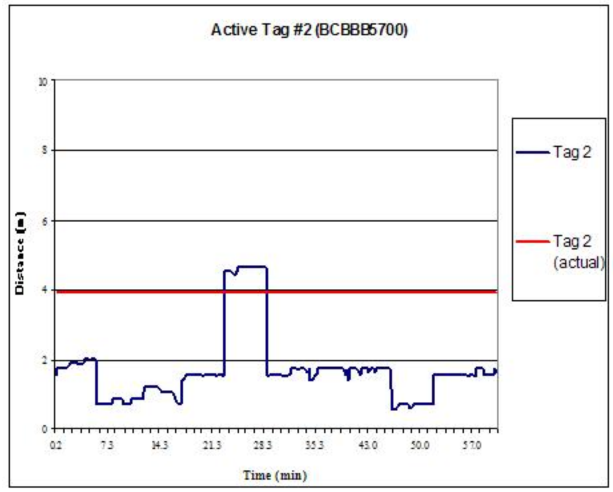

Figure 23: *Plotted values for the second active tag. Note the spike where the tag is reported close to its actual position.*

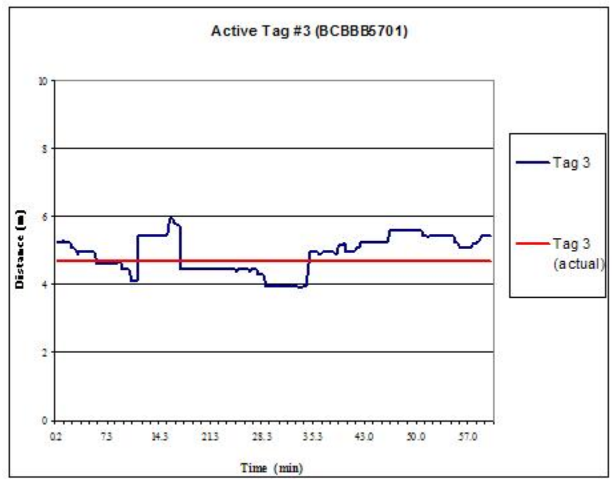

Figure 24: *Plotted values for the third tag. This was the most accurately recorded tag.*

The percent error values, along with the average distance from the actual location, of the recorded tags is shown in Table 6. These tags, while supposed to be the most accurate becuase of the stronger signal, are still not being reliably detected. However, the results from the stick or Ecochip tags may prove more reliable.

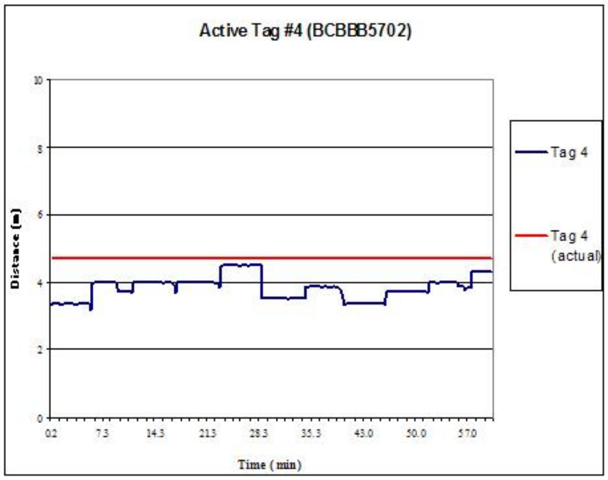

Figure 25: *Plotted values for the fourth tag. This data was the most consistent.*

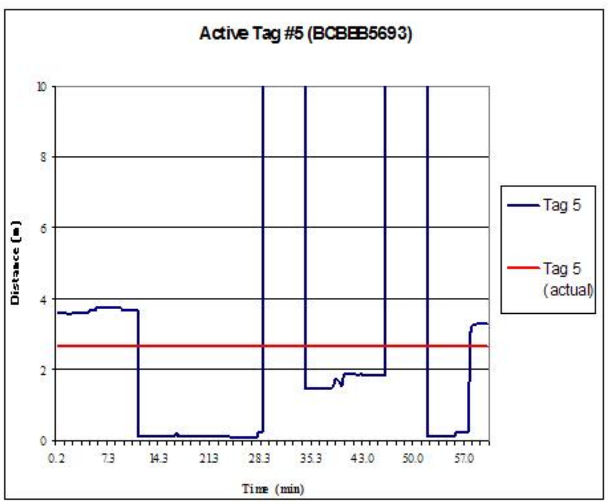

Figure 26: *Plotted values for the fifth tag. Note the wildly ocillating values.*

Table 6: *Multiple, Single Type Tag Test 1 Percent Error*

Tag #	Tag ID	% Error	Avg Error (m)
1	BCBBB5694	11%	0.5
2	BCBBB5700	57%	2.2
3	BCBBB5701	4%	0.2
4	BCBBB5702	19%	0.9
5	BCBBB5693	114%	3.0

6.3.2 Stick (Active)

If the results from the Claymore tag test were considered unacceptable, the results from the stick tag test were deplorable. As can be seen from each table, the stick tag's reported location showed severe ocillation, rarely even coming close to the actual distance the tag was from the reader. This test was alarming in that it showed that the stick tags were very unreliable.

As can be seen in Figure 27 and Figure 28, the first two stick tags have a widely ocillating reported position, ranging from 4.5 meters to 8.5 meters (stick 1) and 4 meters to 6 meters (stick 2). Neither of these tags show the consistency that the claymore tags had shown in the previous test.

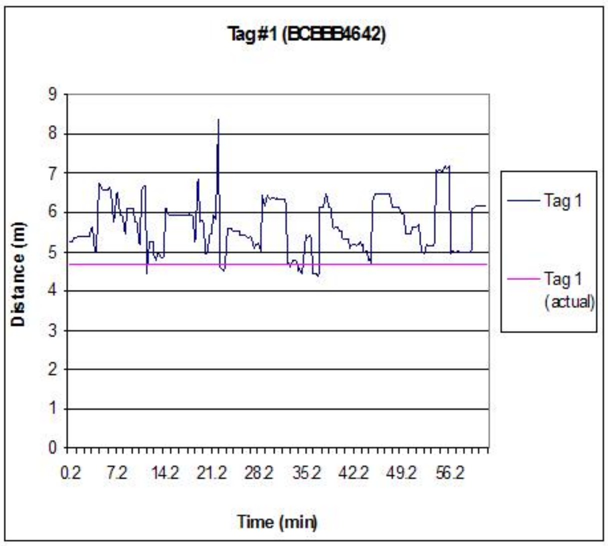

Figure 27: *Plotted values for the first stick tag. Note the wildly ocillating values.*

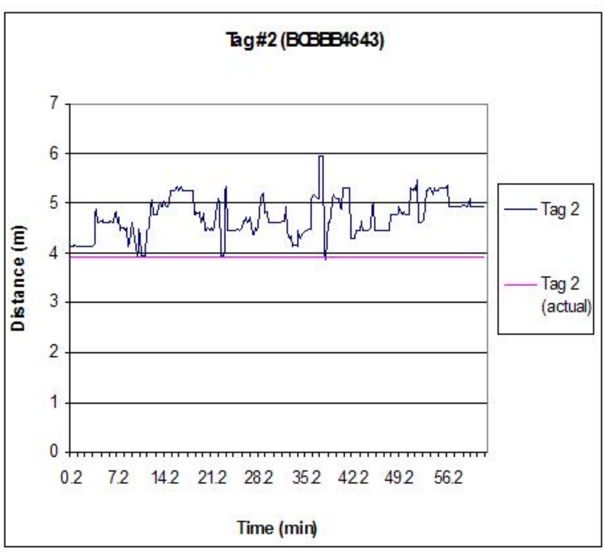

Figure 28: *Plotted values for the second stick tag*

The data plotted in Figures 29 and 30 show that these two tags were constantly being lost and reaqcuired. Rarely did these tags even report a distance less than 60 meters. And several times they were reported at 0 meters (or directly in front of the array). The fourth tag was constantly lost for several minutes at a time (verified by watching SensorFuze run) and while the plotted data looks much cleaner than Figure 29, the fact that the tag is constantly being dropped is cause for concern.

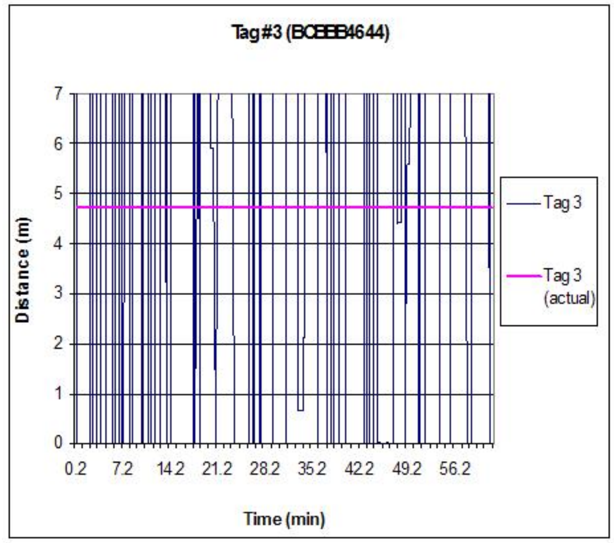

Figure 29: *Plotted values for the third stick tag*

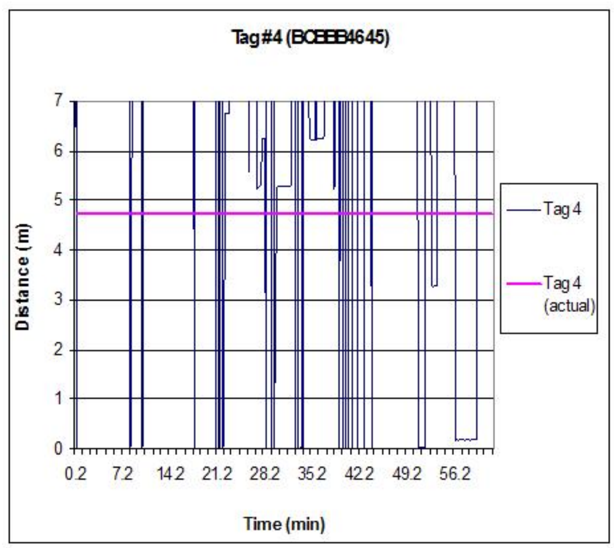

Figure 30: *Plotted values for the fourth stick tag. Note the long periods where the tag was consistently recorded at 60 meters. This is characteristic of constant dropping and reacquiring.*

The fifth tag behaved much like the first two and from the data shown in Figure 31. It is clear that this tag was the most accurately reported. Even with the small distance spikes, the majority of the data seems to be recorded near the tag's actual distance value. But as can be seen from Table 7, these tags are extremely unreliable and prone to being "lost" by TrolleyScan Radar System.

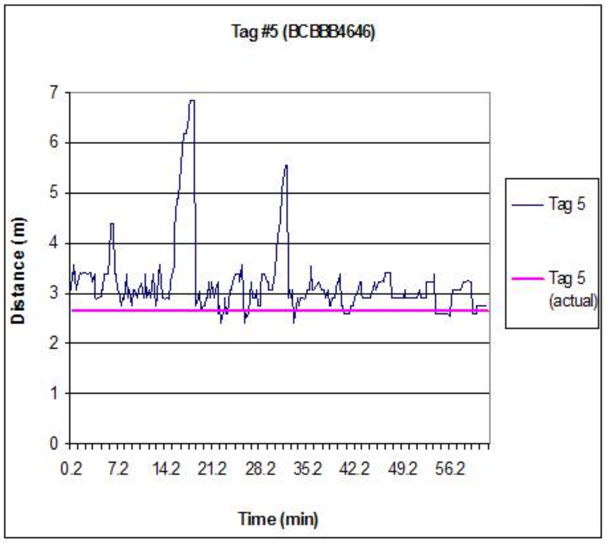

Figure 31: *Plotted values for the fifth stick tag*

Table 7: *Multiple, Single Type Tag Test 2 Percent Error*

Tag #	Tag ID	% Error	Avg Error (m)
1	BCBBB4642	21%	1.0
2	BCBBB4643	20%	0.8
3	BCBBB4644	569%	26.9
4	BCBBB4645	254%	12.0
5	BCBBB4646	17%	0.6

6.3.3 Ecochip (Passive)

The passive tag test results were much better than the Active Stick tag test, but still not what was expected. The first tag plotted (shown in Fig. 32) showed fairly consistent results around the 4 meter mark. The one exception was near the 10-minute mark where the tag was lost and reacquired for almost 4 minutes.

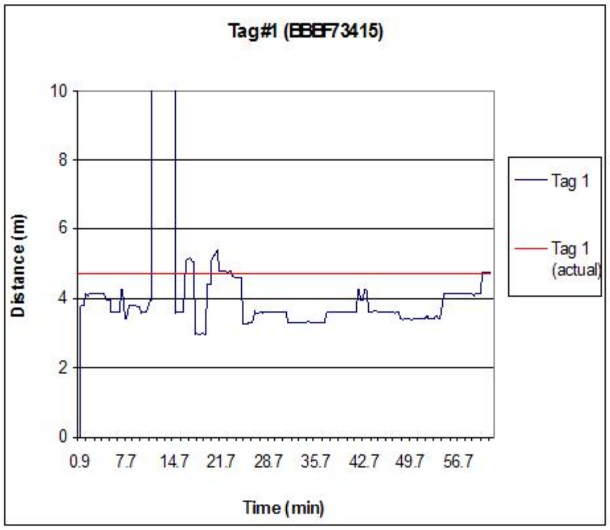

Figure 32: *Plotted values for the first passive tag*

The second plotted tag (shown in Fig. 33) shows even more consistent data, with reported values hovering around the 3.5 meter mark. The spike at the beginning is most likely due to the radar having yet to calculate the distance of the tag based on the received signal. These results were encouraging, possibly meaning that the tags could be accurately acquired within 4 meters. Not quite up to the specifications of the manufacturer, but encouraging none-the-less. The values plotted for the third tag (shown in Fig. 34) are similar in consistency to the second tag.

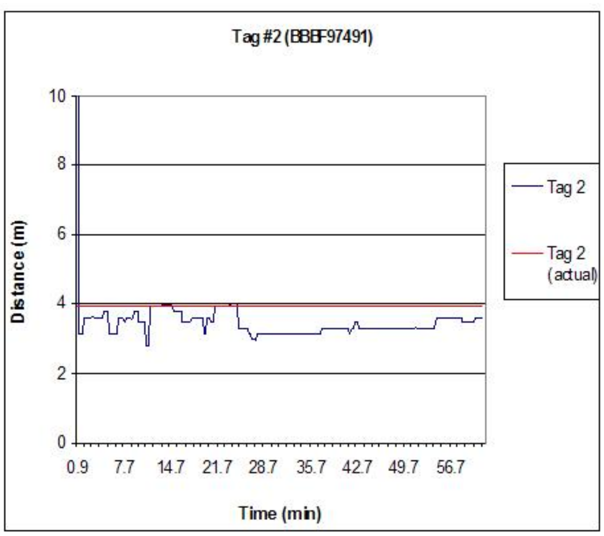

Figure 33: *Plotted values for the second passive tag*

The fourth tag, however, was the closest tag to the array and, although the data was somewhat consistent, the reported distance was

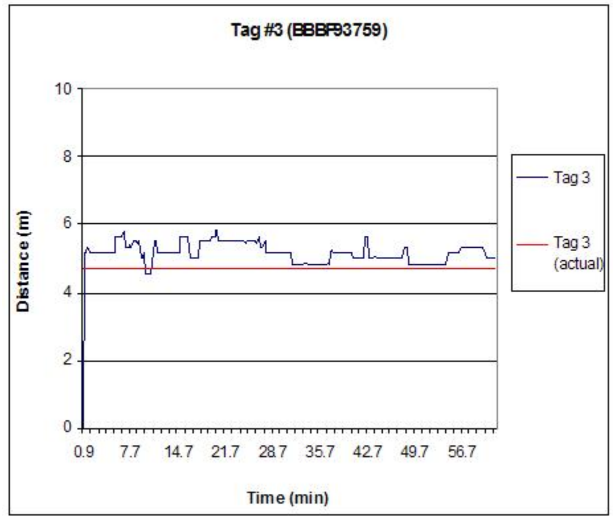

Figure 34: *Plotted values for the third passive tag*

well above the tag's two meter actual distance. The detected range of this tag ocillated much more than the previous two tags, even though it was much closer than them. The distance is plotted in Figure 35.

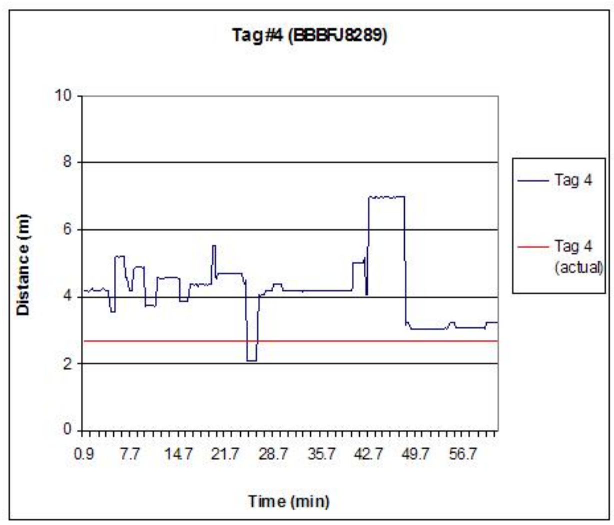

Figure 35: *Plotted values for the fourth passive tag*

The fifth tag was almost dropped from the results section. Obviously this tag was having some trouble. It was rarely acquired and when the tag was picked up by the radar, the tag would rarely remain detected long enough for the distance to be calculated. As shown in Figure 36, the tag was repeatedly dropped. This drop was visually verified by watching the program and comparing it to the written files. The percent error for this test is shown Table 8. Note that even though some of the tags are reported constistently, the most accurately detected tags are still half a meter off on average. Even at close range, the passive tags seem to be unreliable.

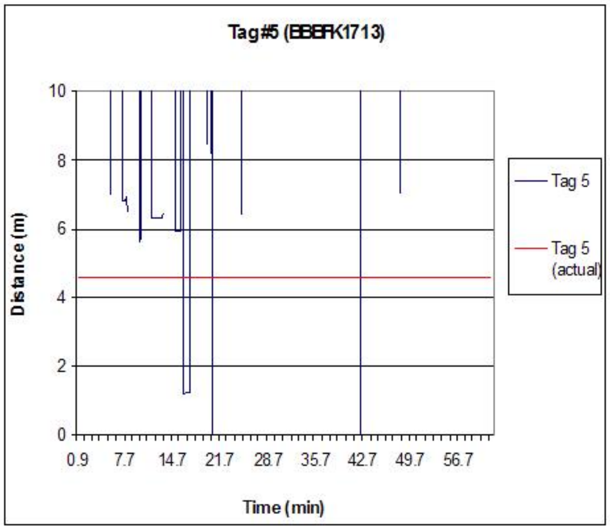

Figure 36: *Plotted values for the fifth passive tag*

Table 8: *Multiple, Single Type Tag Test 3 Percent Error*

Tag #	Tag ID	% Error	Avg Error (m)
1	BBBF73415	27%	1.3
2	BBBF97941	10%	0.4
3	BBBF93759	9%	0.4
4	BBBFJ8289	60%	1.6
5	BBBFK1713	670%	30.9

6.4 Outside Test

The results of this test really drove home the inaccuracy of the TrolleyScan Radar System. Free from the computers, metal girders, ethernet cables, desks, and cabinets, the TrolleyScan Radar System should have performed admirably. The results tell another tale. The maximum detectable distance of the Claymore tag proved to be 33 meters, well shy of the 50 meters the tag should read at. The passive tag was detectable up to 8 meters away, closer to its specification, but still short of the 10 meters that was expected. The stick tag provided no reliable distance measurements. When they were detected, rarely would they stay detected for long.

The different tags did not have consistent distance measurements. When some tags were introduced, the new tags would affect the range of the tags already in the field. The most notable of these effects was when an introduced stick tag caused the calibration tag (previously consistent) to start "drifting" forward and backward. None of the tags were reliably detected. The moving tag showed no distance change, even as it was displaced nearly 20 meters. The "optimal" environment, free from obstructions, seemed to have no noticable improvement on the results.

6.5 Line-of-Site, Mutliple-Tag Test

As previously noted, the out-of-site (or hidden) passive tags were never detected. As shown in Figures 37 and 38, the hidden active tags were detected, but the recorded positions were very erratic. Most of the data positions were recorded as being 0 meters in front of the array while only a few positions were even close to being accurate. The results for Tag 5 are shown in Figure 40. The results for Tag 6 (shown in Figure 39) are even worse. This tag was rarely detected, if at all and when it was detected, the RFID system detected it 0 meters in front of the array. These tags were in direct line-of-sight but were hardly ever detected. It is possible that number of objects in place to hide some of the tags created a form electronic camoflage that "hid" even the tags in plain sight. Another probable explanation is a data collision problem noticed when the RFID system was attempting to detect more than five tags. This problem is discussed further in section 7. The data for this so was erratic that percent error was not even calculated.

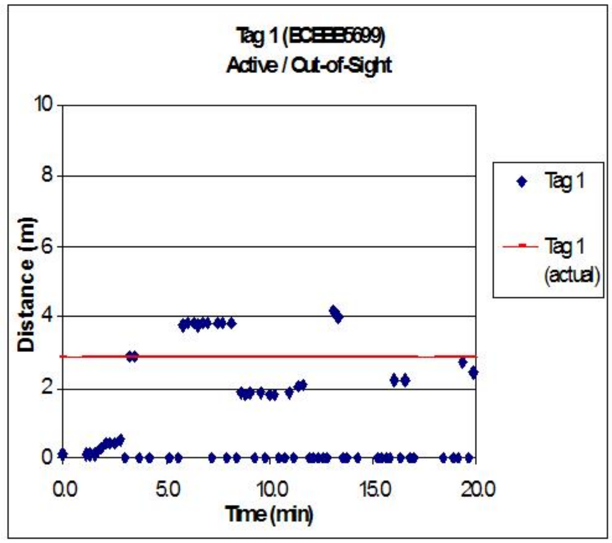

Figure 37: *Plotted values for the first out-of-sight active tag*

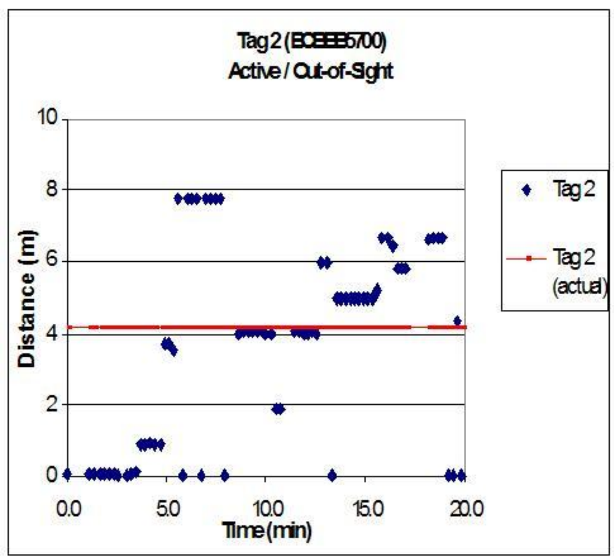

Figure 38: *Plotted values for the second out-of-sight active tag*

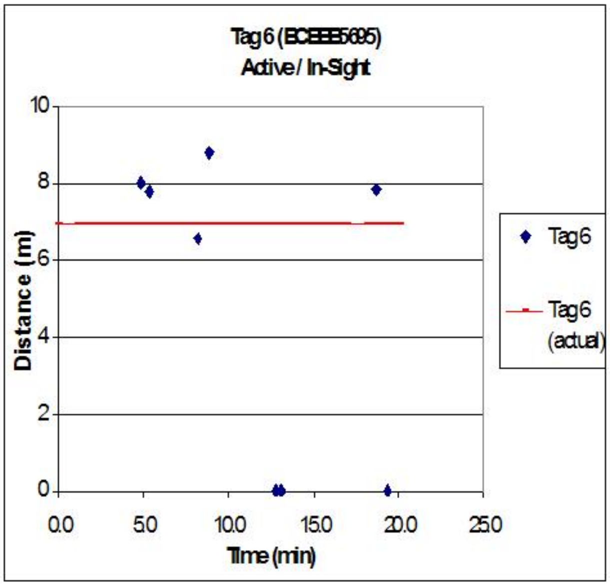

Figure 39: *Plotted values for the third in-sight active tag*

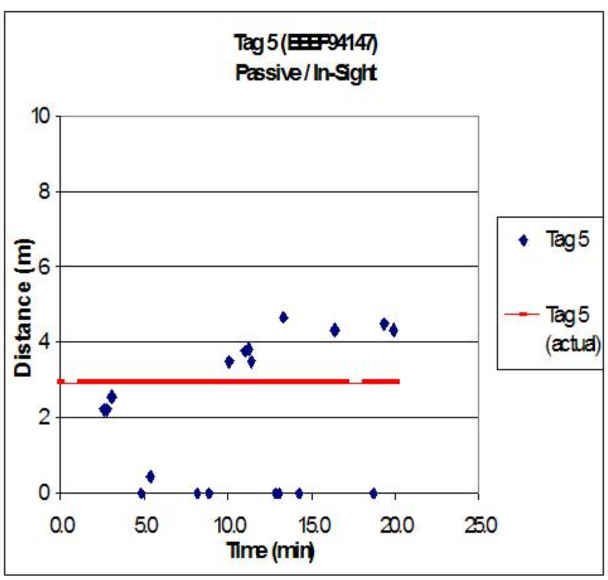

Figure 40: *Plotted values for the fourth in-sight (tag 5) passive tag*

7 Discussion

The test results seem to clearly point out that the TrolleyScan Radar System is inaccurate and unreliable. But what could be causing these results?

As with every experiment ever conducted, the introduction of the imperfect human also results in the introduction of human error. Nowhere was this more clear than in the written software and the TrolleyScan Radar System documentation. The documentation of the commands had obviously not kept up with the evolution of the TrolleyScan Radar System itself and several commands had to be changed to allow SensorFuze to effectively communitcate with the device. Additionally, different data, such as command repeating was being sent back by the radar system resulting in several tags being recorded as "Quiet" or "Noise". Obviously, this data had to be ignored when tags were being processed.

But the most problematic bug noted, however, was created in response to the transmitted tag data. The tag information was transmitted from the array using an undocumented control character, the carot `(^)`. This, along with the different way the data was transmitted resulted in several test runs have 0 distance and angle. When the commands and data were dealt with, a bug in the emulator caused even more problems.

The initial tag data was supposed to include a P to indicate a moving tag and a − when the tag was not moving. The TrolleyScan Radar System instead sent a whitespace character. The emulator sent the necesarry dash and when looking for the dash to parse, instead of finding no note, the programming was finding the negative symbol being sent to report negative angles. This caused the angles and distance to be set incorrectly and as result, the program threw out that data. This caused nearly every test run with SensorFuze to be invalid. Each test had to be repeated once the problem was corrected.

Another source of human error comes in the measurements made during the placement of the tags. Even using a tape measure, the values in inches (or less) were visually estimated. That data was then placed in a calculator to convert the english values into metric values, which were then rounded to the nearest hundredsth digit (.00). All of this introduces some potential error, especially if some values are mistyped when being converted. To overcome this form of potential error, values were constantly rechecked and remeasured.

Other sources of human error were introduced during calibration. Again, visual estimation was used to place the tag directly in front of the radar array. But this tag might have been slightly off by a degree or two. As mentioned above, initially the TrolleyScan Radar System was placed on a table. This table began interfering with the reception of the signals. The documentation said specifically, do not place the device near a shelf. This is another form of human error, only this was a necessity while a different form of mounting the array was developed. Eventually, the table being used as a temporary mount was tilted over and the array was hung from the table instead of being rested on it, allowing the group to get around this potential form of error.

One of the major sources of the inaccurate results could be blamed on the cluttered environment. Signals bouncing off the walls, metallic studs, computers, and desks were clearly creating false signals that the radar system was happily willing to process. This is apparent when using the demonstration software as it plotted the estimated locations of the tags on the screen. After several minutes, each tag produced a 180-degree circle on the screen. The only reason the data was a half circle was because the software did not plot values outside the range of -90 and 90 degrees. However, several times the radar array would return a degree position of 225 degrees, even though this system was not even supposed to be able to see a tag at that angle.

Other sources of error also come from the TrolleyScan Radar System. Several times, especially when dealing with more than nine tags, the data would start to collide. Where the format of the data should be similar to the following:

```
BCBBB4546^P 12.2 -025.6
BCBBB4547^ 10.2 004.0
BCBBB4548^ 9.7 017.3
BCBBB4549^ 3.1 -002.5
BCBBB4546^P 12.2 -025.6
BCBBB4547^ 10.2 004.0
BCBBB4548^ 9.7 017.3
```

the data with nine or more tags would start looking like the following:

```
BCBBB4546^P 12.2 -025.6
BCBBB4547^ 10.2 004.0
BCBBB4548BCBBB4549^ 3.1^ 9.7 017.3
 -002.5
BCBBB4546^P 12.2 -025.6
BCBBB4547^ 10.+ 004.0
BCBBB4548^ 9.( 017.3
```

This form of data simply cannot be processed accurately. The demonstration program, which only displays data, would display this information as is, but would be unable to plot it. SensorFuze, which converts the information to numbers would either lose this tag or convert the value "`9.(`" to a zero. This garbage data would happen often, sometimes with no apparent reason. The data generated as a result would often be skewed. In order to mitigate some this skewed data, SensorFuze was modified to only accept a nine character id string. Anything more or less would cause that data to be thrown out. However, the garbage numbers were not seen until several tests had been run and not all of the data was visually checked. However, few of the values generated during tests resulted in pure 0 values being recorded (with the expection of Fig **??**). Data with garbage in it seemed to appear at random, but was especially noticable with more than nine tags.

8 Conclusion and Future Work

SensorFusion is a project that in theory, will greatly simplify the process of earthquake simulation data acquisition. However, as can be seen from the results of several tests, the TrolleyScan Radar System does not appear to be robust enough or discriminating enough to handle this acquisition. Even with the number of bugs found and corrected in SensorFuze, this piece of software appears to be ready to accept a corrected version of the TrolleyScan Radar System or a different product for RFID detection and can easily be integrated into the suite of tools provided by NEESit. The integration tests were successful (SensorFuze was even modified to create data suitable for Realtime Data Viewer produced by RPI) but the detection tests were not.

Based on the results obtained, the project can be directed onto several different paths in the future. Perhaps the easiest reccomendation to implement would be to acquire a different detection product. A new product would require further research into a different localization algorithm (such Time-Delay on Acquisition algorithms) and some modification to SensorFuze. Another recommedation would take advatage of the existing TrolleyScan Radar System. Obviously there are some flaws in the design. Future work could be directed to detect and eliminate the TrolleyScan Radar System flaws. Flaw correction would require knowledge of the specific hardware used in the design, assembly langauge programming, and some basic soldering skills. Finally, a more radical approach would be to dispense with the RFID altogether and research into alternative forms of localization.

9 Acknowledgements

This paper is based upon work supported by the National Science Foundation under Grant No. CMS-0402490. The authors would like to thank Lelli Van Den Einde, Tess Lacuesta, Joe Cribari, and the NEESreu program for organizing this REU research experience, as well as the entire NEESit team for their support throughout this research project. The above support is greatly appreciated. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

ASSAD, M. 2007. *A Real-Time Laboratory Testbed For Evaluating Localization Performance Of WIFI RFID Technologies*. Master's thesis, Worcester Polytechnic Institute.

GEORGE E. BROWN, J. N. F. E. E. S. C. C. Neesit. Website: http://it.nees.org.

GOUTHAMAN, P. 2007. Sensing future opportunities for rfid. *RFID Journal* (Apr.).

HANNEL, D., BURGARD, W., FOX, D., FISHKIN, K., AND PHILIPOSE, M. 2004. Mapping and localization with rfid technology. *Robotics and Automation, 2004* (Apr.).

JUELS, A., RIVEST, R. L., AND SZYDLO, M. 2003. The blocker tag: selective blocking of rfid tags for consumer privacy. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, ACM Press, New York, NY, USA, 103–111.

LANDT, J. 2001. Shrouds of time - the history of rfid.

MARSH, M. 2004. The impact on rf operating power needed for passive rfid transponders that are attached to metal, wood, paper, glass, and human/animal bodies.

MARSH, M. 2007. Rfid-radar development model handbook.

MICROCHIP TECHNOLOGY INC. 1998. *Passive RFID Basics*.

NI, L., LIU, Y., LAU, C., AND PATIL, A. 2005. Landmarc: Indoor location sensing using rfid. *Wireless Networks* (Jan.).

SANTANA, J. 2004. Finding rfid's break-even point. *RFID Journal* (Dec).

STANFORD, V. 2003. Pervasive computing goes the last hundred feet with rfid systems. *Pervasive Computing, IEEE 2*, 2, 9–14.

STOCKMAN, H. 1948. Communication by means of reflected power. *Proceedings of the IRE 36*, 10 (Oct.), 1196–1204.

VECTORSITE.NET. Wave interference. Available at www.vectorsite.net.

VOGT, H. 2002. Efficient object identification with passive rfid tags. *Pervasing Computing: First International Conference*.

WANG, F., AND LIU, P. 2005. Temporal management of rfid data. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, VLDB Endowment, 1128–1139.

WANT, R. 2004. The magic of rfid. *ACM Queue 2*, 7 (Oct.).

WERB, J., AND LANZI, C. 1998. Designing a positioning system for finding things and people indoors. *Spectrum, IEEE 35*, 9, 71–78.

WIKIPEDIA.ORG. Radio-frequency identification. Available at www.wikipedia.com.

ZHANG, Y., AND AMIN, M. G. 2006. Localization and tracking of passive RFID tags. In *Wireless Sensing and Processing. Edited by Rao, Raghuveer M.; Dianat, Sohail A.; Zoltowski, Michael D.. Proceedings of the SPIE, Volume 6248, pp. 624809 (2006).*, vol. 6248 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*.