Object Localization using Passive RFID Tags

by

RAMPRABHU JAYARAMAN

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

written under the direction of

Professor. Ivan Marsic

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2009

ABSTRACT OF THE THESIS

Object Localization using Passive RFID Tags

By Ramprabhu Jayaraman

Thesis Director:

Professor. Ivan Marsic

Passive radio frequency identification (RFID) systems are revolutionizing the indoor positioning and tracking applications. There has been substantial research on practical applications of this technology and hospitals especially trauma care units are one such area where this capability can lead to improved workflow. Our system uses the Alien RFID reader and the "Squiggle" passive RFID tags to create an effective solution for tracking various medical items. Based on the Received Signal Strength Indication (RSSI) value of the tags, we developed a localization algorithm which uses a neural network estimator to estimate the distances of the tags. To reduce the effect of noise in the RSSI values received from the reader, we accumulate data over a period of time, remove the outliers and the average the remaining RSSI values. The RSSI based estimation algorithm provides very accurate estimation when the spatial density of tags is low (about 25 tags per square meter). To improve the localization accuracy at higher spatial densities we augmented the RSSI method of estimating distances by using the number of times the tags were read or the "read-count". We also investigated how different types of occluding materials affect the localization accuracy. Metal and Humans also can cause complete

occlusion when the positioned in direct line of sight between the antenna and the tag. To overcome human based occlusion, we placed an additional ceiling mounted antenna per 10 m^2. This intervention makes possible the detection (but not localization) of tags when the vertical field of view is not occluded. We also studied the effect of the material to which the tags are attached and determined the effects on localization accuracy.

The software system developed using Java is designed in a modular fashion and provides interfaces to tools like Matlab so that it is easy to experiment to various other localization algorithms. We also developed an intuitive User Interface to display the locations of tags and the associated items. Once a tag is identified its associated description can be looked up in a computer database and this also can be displayed in the user interface.

Dedication

To YOU

Acknowledgement

I would like to thank Prof. Marsic for giving me this opportunity to work on something practical and interesting, for his continuous guidance throughout my research, his support in setting up the system and his belief in me to experiment with things in the way I wanted. I would like to thank Y for being a great support in setting up the experiments in the lab and A.S. for being non-intrusive during the whole time. I would also like to thank RIM for holding back my job for a year and let me complete my thesis in relative peace. I would like to thank my mom for the constant yelling to complete my Masters which in turn came from the inquiries of nosy neighbors and relatives. I also would like to thank RFIDSupplyChain for being a great vendor and providing my numerous quotes and all the other help in the equipment purchases. Thanks to Ruya for well pretty much everything.

Contents

List of illustrations

Chapter 1

1. Introduction

1.1 Motivation

Automatic identification (Auto ID) of objects enables the organizations that manage global supply chains and vast trading partner networks to operate more efficiently and save cost. Auto ID includes a host of technologies like bar codes, smart cards, voice recognition, biometric technologies, optical character recognition, radio frequency identification (RFID), and others. Bar codes have been the primary means of identifying products since late 1960s. RFID offers many compelling advantages over bar-codes, including non-line-of-sight operation. Some of the large retailers and government agencies have realized the promise that RFID offers to businesses and have released mandates and recommendations to their suppliers to use RFID.

RFID is an Auto ID technology that enables products to be uniquely identified without the need for line of sight. RFID enables computers to sense objects and collect the identification codes that are assigned to objects. In combination with the Internet and associated infrastructure, RFID will enable companies to track and trace individual items through the supply chain, i.e. from the manufacturer, through the distributor, to the retailer, and finally to the consumer. RFID aims to provide retailers a near-perfect supply chain visibility. That is, companies would be able to know exactly where every item in their supply chain is at any moment of time. In essence, RFID is revolutionizing the way products and goods are tracked and traced in the supply chain. Retailers consider RFID as an investment for the future providing advantages like cost reduction by maintaining correct amount of stock levels, increase in revenue by reducing the out-of-stocks, counterfeit protection, shrinkage protection, and real-time tracking of supplies. These benefits are pervasive throughout the supply chain. In a highly competitive business environment, RFID represents the next level of supply chain efficiency that many companies are striving to attain.

The use of RFID has been quite extensive in a supply chain environment and has thus been studied quite extensively. One of the avenues of expansion of this technology are Hospitals, which draws lots of parallels to supply chain industries.

Hospitals manage a huge inventory of drugs and other items like injections. Just like any goods in a warehouse these items can be tagged and monitored. The same set of operations carried out in a warehouse can be applied to a hospital environment, like stocking, usage monitoring and localization/tracking. In this thesis we explore the use of passive RFID technology in the hospitals to track hospital items especially the importance of automatically tracking these items in a trauma care environment. This is an unexplored application of RFID technology or localization and hence we hope to present a nascent system that can be explored further in such a critical application area.

1.2 Research Questions

Since this an explored area of research in terms of the application, we were faced with many questions. We tried to answer some of the fundamental questions that would create an application area that can lead to further research. Some of the questions are listed below.

1. Scalability:

   How big an area can be covered by such a system and still provide an usable system. Hospital rooms can be anywhere between 3m X 2m to really large wards. How does the system for large area of coverage.

2. Accuracy:

   How accurate can the system be? Trauma care is a place which can be life saving, the system should be very accurate to be of any practical use.

3. Cost:

   How much would such a system cost, if planned to be deployed across an entire hospital

4. Performance:

   How fast will the system be in tracking objects? A tracking application especially in a trauma-care environment should be pseudo-real time if not real-time.

5. Usability:

   How will such a system be deployed ? How intuitive would the system be to use ? The hospital environment cannot be altered to fit a new system in place and the user interface has to be intuitive enough to be usable in ‘panic mode’.

6. Extensibility:

   How expandable the system is for future usecases ? Being a nascent area of application , gives the system ample scope for future extensions and new techniques to establish what has been done in this work. What are the characteristics that should be present in the current system that can facilitate easy extension?

In this thesis we present a system that answers these questions. We also present the analysis of the findings that show some clear areas of improvements.

2. Background

RFID systems provide an automatic means to identify physical objects without the need for line-of-sight communication. The main components of a RFID system are tags, Readers, and host computer. RFID tags are attached to physical objects as a means to identify them. RFID readers convert the radio waves sent from the tags to get the digital data and send the collected data to the host computer. RFID tags used in supply chain carry a unique serial number called Electronic Product Code (EPC) [7]. Mandates require that the tags deployed in the supply chain to be primarily passive UHF EPC-compliant tags. The time constraints of these mandates have driven the need for good source of information for comparing UHF EPC-compliant tags. Hence, in this thesis we are concerned about the benchmark metrics for passive UHF EPC-compliant tags.

The passive UHF RFID tags used in the supply-chain form only a small portion of a variety of RFID systems that have been developed. For perspective, a brief history of RFID and the broad classifications of the RFID systems are given before describing passive UHF RFID and passive RFID performance in more detail. In this chapter, we will also discuss a basic background on UHF RF that is relevant to this thesis.

2.1. History of RFID

RFID is a term that refers to a family of technologies that has existed since 1940s. It has been suggested that the first RFID related technology was invented by the British in 1939, and was routinely used by the allies to identify airplanes as friend or foe. This technology was called as Identify Friend or Foe (IFF).

Since the invention in 1939, RFID has undergone significant development with advances in different fields. In the 1960s and 1970s, various governments developed identification technology to track military equipment and personnel [16]. By the late 1970s this identification technology was used for identification and temperature sensing of cattle. However, the wide use of the technology was possible only by late 1980s and 1990s when the semiconductor companies were able to achieve improved performance with size and cost reduction. This enabled RFID systems to be used in many new practical applications. From then on, passive RFID has found its use in access control and security, airline baggage handling, inventory management and asset tracking, and smart cards. There has been continued work on finding innovative methods for achieving low cost and high performing technologies. However for wide scale adoption of RFID such as in the supply chain, RFID systems from different vendors must be compatible with each other and also must be able to operate under regulations from various countries. In order to make different vendors use the same specifications standards are essential. EPCglobal Inc. has been leading the development of industry-driven standards for using RFID in the supply chain.

2.2. Taxonomy of RFID systems

Through the development of RFID to its current state, there have been many different varieties of RFID systems. A classification of the existing RFID systems would help us understand the larger universe of RFID. This will enable us understand the various possibilities with RFID systems and the reason for using UHF passive RFID tags in the supply chain.

The variety and the operating principles of RFID systems have enabled classification along several dimensions. Passive UHF RFID used in supply chain forms only a small part of a larger universe of RFID systems. In this section, we give the description of the larger RFID universe before going into the specifics with passive UHF RFID.

2.2.1. Chip and Chip-less tags

RFID tags can be classified as chip and chip-less tags based on the way the tags store data. The chip tags contain an integrated chip (IC) to store the unique data. The power needed to operate the integrated chip is derived either from the reader's RF signal or from an on-board battery source. Chip-less tags do not contain an integrated chip, but they encode unique patterns on the surfaces of materials. These patterns constitute the data that is reflected back to the readers. An example of chip-less tags is the Surface Acoustic Wave (SAW) RFID tags which are based upon the piezoelectric effect and on the surface-related dispersion of acoustic waves at low speed.

Although chip-less tags seem to provide the minimum of functionality – a read-only device with a unique number, the technology is not mature enough for adoption in supply chain. Though chip-less technology show tremendous promise in the future, the chip tags offer the most near-term solution for the majority of track and trace in supply chain.

2.2.2. Auto-ID Class Structure

Auto-ID center was founded in 1999 to develop an open standard architecture for creating seamless global network of physical objects. Auto-ID center has provided a layered class structure to classify UHF RFID tags based on their operation and functionality. The class structure classifies various mutations of tags into a class structure ranging from the least sophisticated Class 0 to the most sophisticated Class 5. Class 0/1 tags both represent basic capability. They are read only passive identity tags. The passive tags derive the power needed for operation from the reader's RF signal. They communicate back with the reader using backscatter modulation. The Class 0 protocol uses out-of-band signaling while Class 1 protocol uses in-band signaling. Class 0 tags are read-only, programmed by the manufacturer, whereas Class 1 tags are generally viewed as write once and read many where the writing can be done either by the manufacturer or by the user. Class 2 tags are passive tags with additional functionality like encryption or memory. Class 3 tags are semi-passive tags. These tags have a battery source for operating the internal circuitry, whereas they do not have a transmitter for sending back the information. All the tags from Class 0 to Class 3 use backscatter techniques to communicate to the reader at UHF frequencies. Class 4 tags are active tags, which have a battery source and a transmitter. They may be capable of broadband peer-to-peer communication with other active tags in the same frequency band or other readers. Class 5 tags are devices that can power other tags as well as communicate with other Class 4 tags. An example is a RFID reader that is capable of powering up the other Class 0/1 tags. Since the tags used in supply chain will be used on almost every product/case, the tags must cost as less as possible. Of all the tags it is possible to achieve lower costs in near-term with Class 0 /Class 1 tags. Thus, the mandates require Class 0/1 passive tags to be deployed in the supply chain.

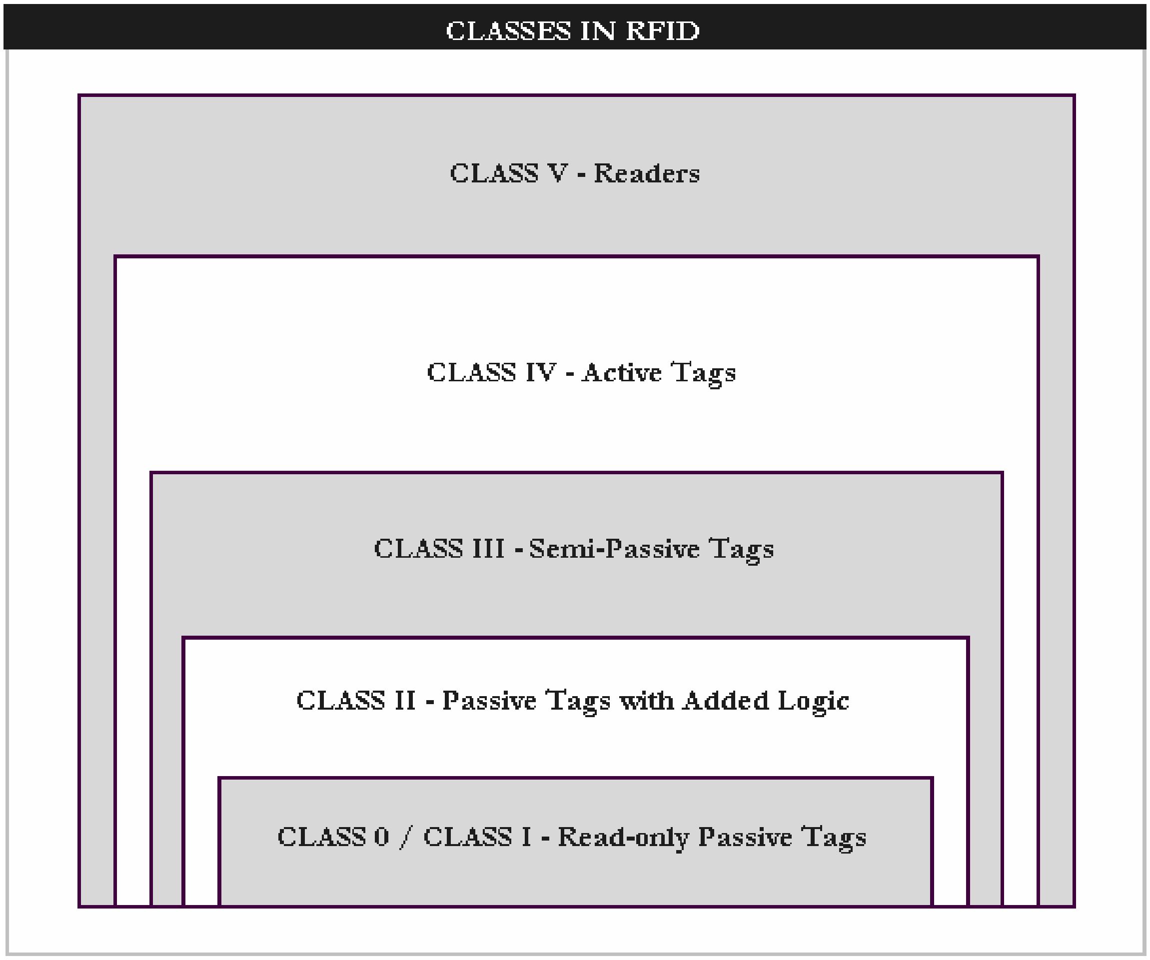


Figure 1 Class Structure for RFID classification

2.2.3. Frequency of Operation

Another major classification dimension is the frequency at which the RFID systems operate. RFID systems generally operate in specific Industrial Scientific Medical (ISM) bands that occupy portions of spectrum from low frequencies like 125 kHz to microwave frequencies like 5.8 GHz. The mandates require that the RFID systems be operated in the UHF (Ultra High Frequency) frequencies occupying the ISM bands in 860 – 960 MHz according to frequency restrictions in different countries. The read range offered by UHF RFID makes this frequency the most attractive for supply chain implementations.

2.3. Components and Functions

As was described at the beginning of Chapter 2, the RFID system consists of three components: tags, readers, and host computer. The general working of passive RFID system is as follows: the reader transmits a query for all the tags to respond. The tags that are powered and which have recognized the query from the reader respond back to the reader. Both the tags and readers typically implement a command protocol necessary for the identification of a single tag or multiple tags within the reading range of the reader. These protocols have anti-collision algorithms that reduce the occurrence of multiple simultaneous transmissions from different tags to a single query from the reader.

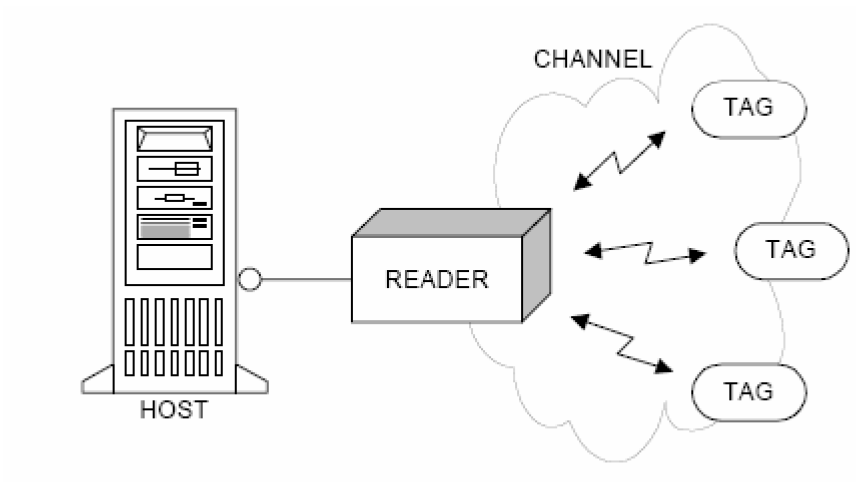


Figure 2 RFID system components

Readers are generally radio transceivers with antennas connected to them. This radio transceivers typically implement a variety of protocols meant for tag-reader communication. Passive RFID tags are attached to objects and the RFID tags contain an identification code (ID). EPC-compliant tags used in the supply chain are programmed with an ID called as electronic product code (EPC). EPC generally consists of a unique identifier, a cyclic redundancy check, and a short password [13]. Physically, a passive RFID tag is composed of a chip, an antenna on top of a substrate, and may contain a label (adhesive paper for attaching to the product). Figure 3 shows the components of a sample passive UHF RFID tag.

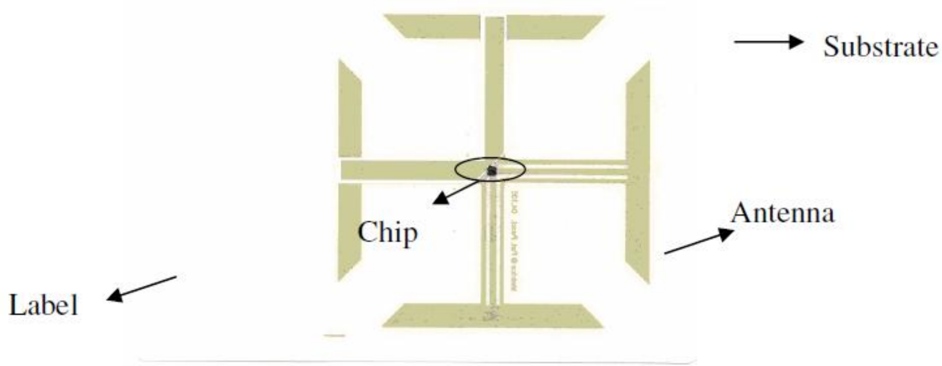


Figure 3 Sample Passive RFID tag

Figure 4 The "Squiggle" passive RFID tag

The other component in the RFID system, the host computer is meant for collecting all the raw EPC numbers from the reader. A middleware can be used in the host computer to Antenna, Chip, Substrate, Label convert the raw EPC numbers to the objects to which they are attached. Different kinds of logistical analysis can be realized through consolidation of the outputs from the middleware.

Before explaining in detail about the way passive UHF RFID works, it would be helpful to discuss the principles involved in the tag-reader communication and vice-versa.

2.4. UHF RF Communication Principles

The communication between the reader and the tags in UHF take place in the ISM band of UHF in various countries. UHF communication between tags and readers take place through the electromagnetic (EM) waves that propagate through the environment. In this section, we will discuss the principles involved in communication between tags and readers in UHF and vice versa.

2.4.1. Reader to Tag communication principles

In supply chain, RFID tags are typically used in the far-field region of the reader antenna. Hence, the reader to tag communication takes place using far-field communication principles. Far field distance from an antenna is estimated using Rayleigh distance or far field distance. For different radiating structures, it has been estimated that the far field distance is given as:

$$r > \frac{2D^2}{\lambda}$$

where *D* is the maximum dimension of the radiating structure and r is the distance from the antenna. It should be noted that this is only an estimate and the transition from near field

to far-field is not abrupt. Typically *D* for reader antennas is 1 foot. The far field distance in UHF ISM band in USA (915 MHz) can be estimated to be 56 cm.

In general, the reader and tag are separated by a distance of 2 foot to ensure that the tag is placed in the far-field of the reader antenna. As is common in most of the wireless communication systems, the coupling takes place using the transmission, propagation, and reception of EM waves. The power received by an antenna from in terms of the power transmitted by another antenna separated at a distance of *r* is given by Friss transmission formula:

$$P_r = \frac{P_t G_t G_r \lambda^2}{(4\pi r)^2}$$

where ${}_{r\,t}P, P$ is the received and transmitted power of the antennas, ${}_{r\,t}G, G$ are the gains of the receiving and the transmitting antennas, and l is the wavelength. The above formula assumes that there is no polarization mismatch between the transmitting and the receiving antennas. Also, the power available to the load must include the impedance mismatch between the load and the antenna impedances. Thus, the power available to the load in the receiver is given as:

$$P_r = pq \frac{P_t G_t G_r \lambda^2}{(4\pi r)^2}$$

where p denotes the polarization mismatch factor between the transmitting and receiving antennas and q denotes the impedance mismatch factor between the loads and receive antenna impedance. The fraction of the received power 1- q is not delivered to the receive load and is scattered. The above equations show the amount of power delivered to the tag from the reader at a given fixed frequency.

The frequency of operation for the reader to tag communication in passive UHF RFID is not fixed. Reader does a frequency hopping in the ISM band in UHF for communicating with the tags. The frequency hopping avoids interference that might occur due to other devices using some part of the spectrum in ISM band. Also, the modulation schemes used in the reader to tag communication depend on the type of the protocol being read.

2.4.2. Tag to Reader communication principles

The passive tags do not have a transmitter to communicate back with the reader. The tags communicate back to the reader by changing the load impedance. The variation of the tag’s load impedance causes a mismatch between the tag’s antenna and load. These causes

some amount of power to be reflected back and scattered through the antenna. The return scattered signal from the tag is detected and demodulated by the reader. The variation of load impedance causes different amount of powers to be reflected back to the reader. This method of communication is called as *backscatter modulation*.

Figure 5 Illustration of Backscatter Modulation

Figure 4 shows the extreme case of backscatter modulation for illustration purposes. When the load (red) and antenna (black) are perfectly matched, the antenna delivers the received power to the load. When the load and antenna are mismatched, the power received by the antenna is reflected and radiated back. The change between these two extreme conditions are used to modulate the response back to the reader. In practice, the load variations are not this drastic. Typical value of impedance for a UHF RFID chip produced by Philips is 16 – j350 _ [17].

2.5. Passive UHF RFID System – Working

A passive RFID system consists of passive RFID tags and a reader capable of reading them. The principles mentioned in Section 2.4 are used for communication between the passive tags and reader. In this section, we put all these principles together and explain the basic working of passive RFID system.

The primary working principle of a passive RFID system can be explained using Figure 5. Figure 5 shows a part of the circuit for a simple passive RFID chip. The RFID reader sends out RF energy in attempt to read tags. The tag antenna is tuned to receive the RF energy. The bridge rectifier charges a capacitor using the RF energy that the antenna receives. Once the capacitor is charged to a certain voltage, the combination of the capacitor and Zener in breakdown serves as a voltage source. If enough energy is available to drive the internal circuitry of the RFID chip, the tag begins to perform the demodulation and processing the commands sent by the reader. The tag responds to the issued commands by switching the load at the antenna terminals from matched to unmatched conditions according to the tag response signal. The switching of the load between matched and unmatched conditions would absorb and reflect the signal transmitted from the reader. When there are multiple tags responding to a command, the RFID air-interface protocol has an anti-collision algorithm to detect collisions [7].

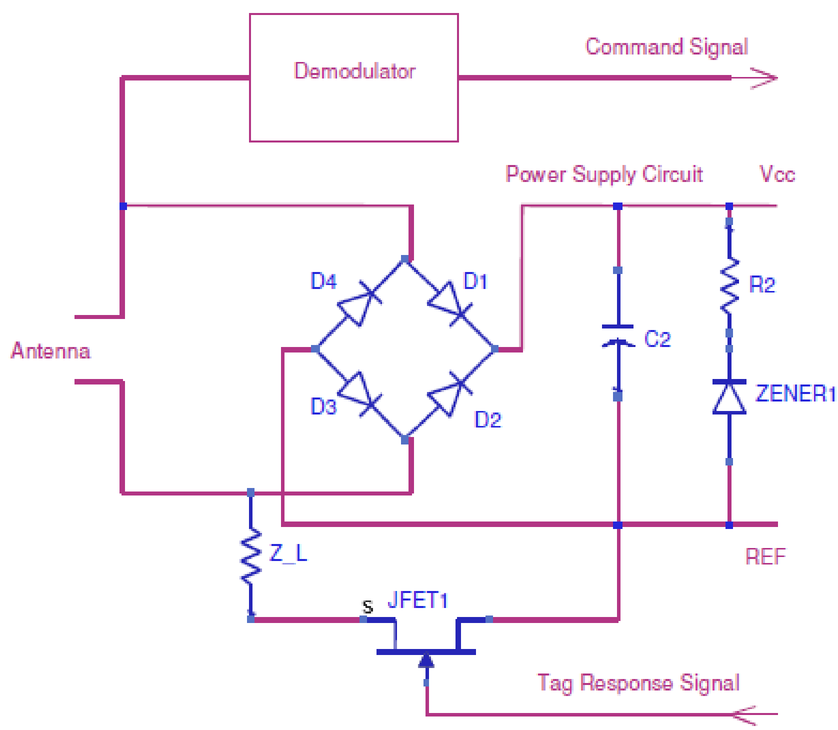


Figure 6 Part of the circuit for a RFID tag

The mandates require the use of EPCglobal Class 0 and Class 1 passive RFID tags.

EPCglobal is a leading standards body for RFID involved in the development of industry driven standards. EPCglobal Inc. standardizes the specifications of the Class 0 and Class 1 protocol. The reader and the tag talk with each other using these protocols. These protocols are briefly summarized in the next section.

2.6. EPCglobal Class 0 and Class 1

The Class 0 protocol is meant for implementation of read-only passive RFID tags [12]. Reader to tag communication is accomplished through an amplitude-modulated carrier. Tag to reader communication is accomplished through passive backscatter of the tag to reader carrier already described to produce widely separated sub-carrier tones. A population of tags to be read by the reader can be represented as a binary tree. The reader scans the tree from the root to leaf to fully define an EPC. The process of finding a single tag in a population by scanning the tree is called as *tag singulation*.

The Class 1 protocol [13] is meant for implementation of read/write tags. Reader to tag communication is accomplished using amplitude shift keying (ASK). Binary data from reader to tag is encoded as pulse width modulation of the low level pulse. The tag to reader communication use passive backscatter that follows a scheme where two transitions are observed for a binary zero and one transition for binary one. When a population of tags is to be read by the reader, the reader puts the tags that are already read to *sleep* so that the reader can focus on reading the difficult-to-read tags [14]. These difficult-to-read tags can be at the edge of the read field or on an RF-absorptive material. In practice, RFID reader performs a sequence of *wake up*, *read*, and *sleep* cycles to ensure that all the tags in the field have been read.

2.7. Performance of UHF Passive RFID

Until now, we have been considering the behavior of EM fields and waves in free space (environment is uniform and there are no objects to interfere transmission and reception). In practice, the environment is not truly free space and there are various factors that might affect the performance of the RFID system.

The main factors that affect the performance of a RFID system are the tags, readers, and the environment in which they are operating. The medium over which the tags and the reader communicate is called the *channel*. The channel affects the communication between the tags and reader due to effects such as attenuation, multi-path, and interference from other readers and RF devices. The physical objects to which the tags are attached also affect the tag's performance. Many common materials that the tags are attached such as metal and water have considerable effects on the performance of the tags. Changes in impedance bandwidth, detuning of the antennas, and reduction in the efficiency of the antenna is some of the factors that change the amount of power being delivered from the antenna to the chip. The environmental effects observed at UHF frequencies can be classified into material effects observed with conductors, dielectrics, and in free air. In this thesis, we separately analyze the environmental effects of tags near metal, water, and free-air and develop different performance metrics.

Another factor that affects the performance of RFID system is frequency of operation. It should be noted that all the performance metrics of the tags are frequency dependent. Also, the ISM band in UHF frequencies varies among countries. For example, the ISM band frequencies are 860 – 868 MHz in Europe, 902 – 928 MHz in USA and Canada, and 950 – 956 MHz Japan. Thus if the tag is to be read globally, it should operate well across the spectrum. In this thesis, we have also developed benchmarks for the frequency dependency of tags near metal and water.

Although RFID performance is a concern for those who are deploying RFID, we are not aware of any published standard or recommendations towards a well-defined set of performance measures. EPCglobal Inc has realized that RFID performance is an issue and is taking steps towards a performance standard. We are aware of a group in EPCglobal Inc working towards a performance standard. But it is currently not visible to the public and there has been no published recommendation towards performance. The only published previous work in this area was [10]. This essentially lists out a set of simplistic approach for comparing different RFID product offerings by end-users.

3. Related Work

There have been quite a few localizations methods for passive RFID based systems.

Concentric Power Levels

One of the popular methods of estimating distances is by controlling the power levels of the antenna and getting a list of tags read at each level. The power levels can be then correlated to a distance measure and the tags can be positioned. Though this method can be used on RFID readers that do not supply the RSSI information and would make it a more generalized solution, we found this method was not any more effective than using the RSSI parameter for estimation. Also in order to estimate the entire set of tags in the read range we need to cycle through at least a few power levels depending on the granularity of the distance estimation. Also tags which are close enough would be found in multiple power levels hence might affect the read counts of all the tags.

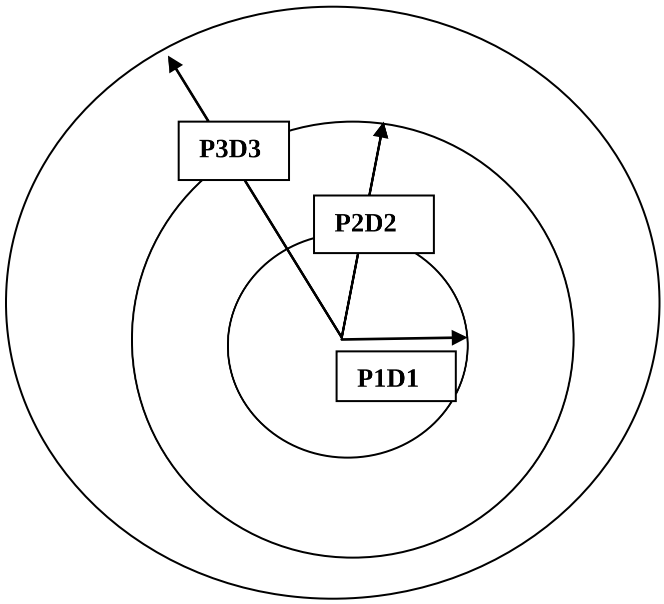


Figure 7 Increasing power levels of the transmit antenna

In the above figure the center of the concentric circles represent the reader position. **P1D1** represents the iteration of the lowest power level P1 and the corresponding distance is D1. All tags found at power levels P1 are assumed to be at distance D1. **P2D2** represents the iteration of the next power level P2 and the corresponding distance is D2. All tags found at power levels P2 are assumed to be at distance D2. **P3D3** represents the iteration of the highest power level P3 and the corresponding distance is D3. All tags found at power levels P3 are assumed to be at distance D3.

We can vary the power level in smaller steps if we need more fine distance estimation. In our experiments with the Alien RFID Reader we used, we found that the steps to achieve an clear P1D1 and P2D2 measure was hard as the we had to significantly vary the power levels to see any difference in the read range and also this was quite inconsistent and slow.

Read Count based Estimation

This is another popular method of estimation. This uses the fact that tags that are closer to the reader would be read more in a given set of iterations. For example the reader is issued about 100 read commands and the list of tags found in each of these 100 iterations is stored. At the end of the 100 iterations the list of tags is sorted based on the number of times they were found in the read and the distance estimation is then a correlation of the percentage (%) of successful reads. Like the previous method this again suffers from the fact that 100 reads are required to make each estimation and hence this cannot be used in a pseudo real-time system like ours.

We use this technique as a secondary method of estimation as explained in **5.2.3**

Reference Tag based Estimation

This method involves having fixed tags in the field of view called reference tags,whose positions are known in advance. At any point of time the entire field is scanned and the position of the given tag is estimated based on a nearest neighbor estimation. In the nearest neighbor estimation the tags signal strength is compared to all the reference tags and 3 nearest neighbors are identified. The tag is localized based on the positions of the 3 reference tags. We found that this technique was computationally intensive as the given tag needs to be compared to all the reference tag in the field to obtain the three nearest neighbor and also the reference tags increase the tag concentration in the field of view which we have observed degrades the performance of the localization system by reporting much lower RSSI values for tags.

K-Means Clustering based Estimation

The distance estimation explained in section 4.1.2 is approximated distance estimation and can actually be seen as a classifier problem. Assuming each unit of distance as a class and the RSSI value is the input for the classifier. We can then identify a class for each of the inputs and the class would then be the distance estimation for that particular RSSI value. The below figure shows the output of the K-Means classifier

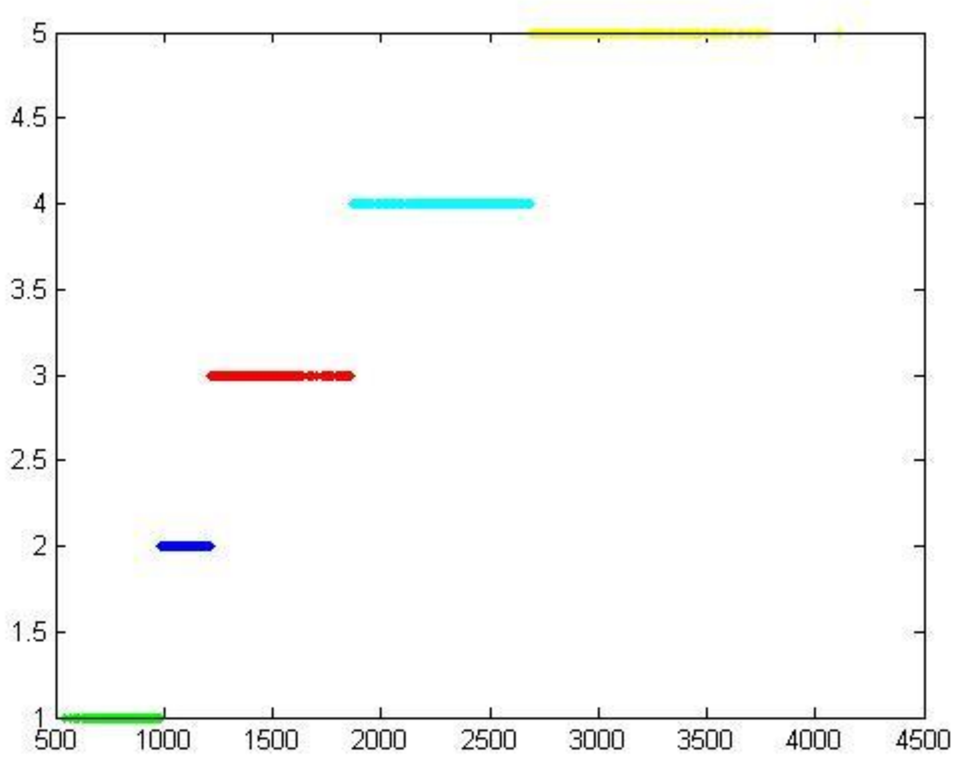


Figure 8 Distance estimation based on K-Means clustering

The K-Means clustering algorithm works quite well for simple cases. We later found out that a Bayesian inference based system outperforms the K-Means clustering based solution.

4. Technical Approach

4.1 Localization:

Localization can be loosely defined as the principle of determining the location of a given object with respect to some reference frame/coordinates. All current localization methods use a signal transmitter, a central reader/receiver which computes the location of the transmitter and sends it back to the transmitter or any other auxiliary system interested in knowing the location of the transmitter.

One of the most popular localization that is in use currently is Global Positioning System (GPS) which determines the latitude, longitude of the given gps receiver. Here the GPS chip/hardware transmits radio signals and the satellites receive, process the same and sends back the location, altitude and speed of the transmitter. GPS is a large scale localization system where an accuracy of a few meters is considered very accurate. Next is the notion of indoor localization where we may want to localize objects inside a building, for example. Here systems like GPS aren't effective. Though this has been successfully done using technologies like Bluetooth, WiFi, the usage of RFID for this is quite feasible and is being increasingly studied/deployed. This has become more popular with the introduction of the UHF models. We would be using a method called "Trilateration" for achieving these using RFIDs.

4.1.1 Trilateration:

The term triangulation is more commonly used when it comes to localization. Trilateration is a similar technique where instead of computing angles we localize objects using only distances.

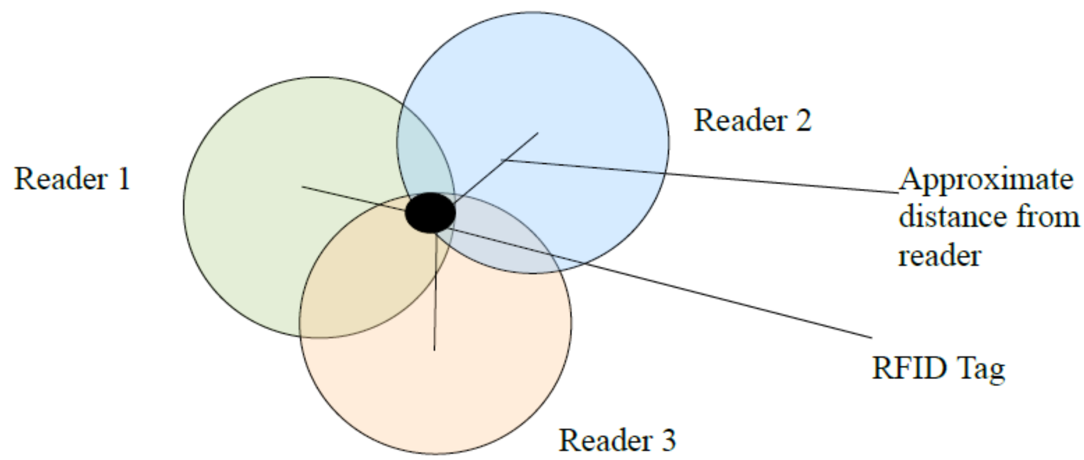


Figure 9 Trilateration illustration

The above figure illustrates the method of trilateration. The black region in the center is the approximate location of the identified object using the 3 readers. In our case the object is an RFID tag and can be uniquely identified using the Tag ID. A RFID reader can read tags at a much greater distance if there is line of sight hence the circles above representing a reader's read range will actually be an eclipse in reality.

Also currently we only consider localization in 2 dimensions though for practical purposes we may need to extend this to 3 dimensions. In case of 3 dimensions the circles above will be represented by a sphere.

4.1.2 Approximated Trilateration

We differ from the traditional trilateration approach in our localization methods. Though this approach has been chosen more due to the needs of the end application of our system, we believe this approach can be used in various applications. The idea is to have a slightly approximated method of trilateration. In our system the area of interest where the tags would be localized, is divided into squares of area 0.5m X0.5m. The 0.5m resolution was chosen using empirical means and gave us a balance between accuracy of estimation and an acceptable least count for estimation. We treat distance as a discrete measure and not as continuous measure to improve accuracy. The effect on accuracy comes in the distance estimation part of the system. We found that treating the estimation problem as a classification/clustering problem gives us better accuracy and more flexibility to use various algorithms than the one currently used. We can now train system with a large set of data for a relatively fewer significant data points.

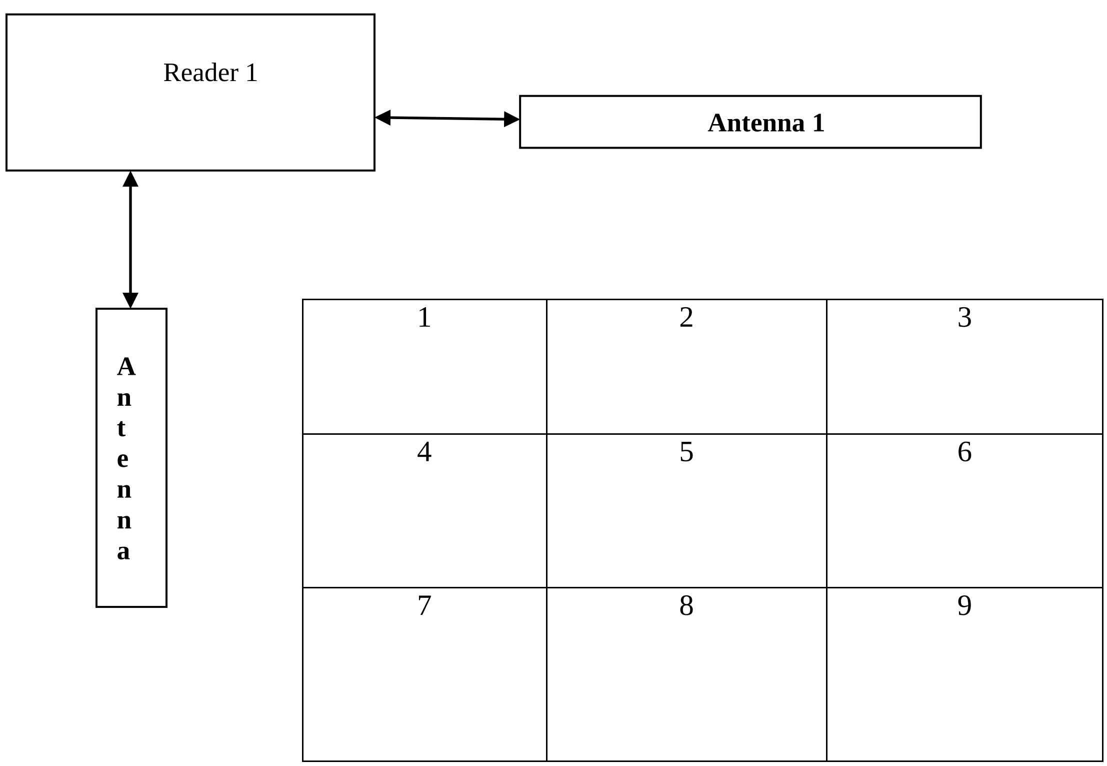


Figure 10 Field of view of one reader and unit of measurement

In the above figure, each square represents a distinct area to which a tag found will be localized. Any tags estimated to be within these squares will not be further localized to a finer resolution. So given a tag the estimation of its position would be 1,2,3 … to indicate which square the tag is in. Since the squares are all of equal size and the position of the readers are known, we can translate this trivially into a distance measure like (1m, 1m) from say position 5. Also we have to note the fact that this involves 2 antennas and just 1 reader which is sufficient to localize tags in the given field.

4.2 Hardware

The process of selecting the right hardware for the system was a challenge in itself. The hardware is such a crucial piece of the system. We spent a few months researching and finding the right hardware for our experiments. Though this was time spent without any actual research, this turned out to be time well spent as the hardware we finally discovered was powerful enough for our experiments and had additional features which helped us design a more feature rich system. The hardware used in our experiments was the Alien 9900 UHF RFID reader. This comes with a rich set of software apis in various programming languages like Java, .NET and Visual Basic. The ideal hardware is an essential blend of hardware and software. The Alien 9900 seemed to be exactly the same. Some of the advanced features of the hardware were

1. Ability to get the Received Signal Strength Indication (RSSI). This was single most essential feature of this hardware that made it ideal for our experiments. Out of the

popular non-enterprise RFID readers Alien was on the only one that was able to provide with this feature.

2. Advanced features like speed, position and direction tracking. The hardware had the ability to calculate the speed in which the tags are moving and the relative position of the moving tags from the reader. This is done at the signal level where the reader counts the number of cycles between a request and a response from the tag to calculate speed and direction.

3. Network interface to connect to the central computer. This means that the computer controlling, reading the RFID readers can be placed in a remote place away from the readers. Also without any custom wiring hubs we can control the whole system using just one computer by plugging all the readers to an Ethernet hub and is inexpensive to setup the system. This also aides to the scalability where we can have multiple readers wired to the system and adding a new reader to the system is as simple as plugging the Reader to the Ethernet hub.

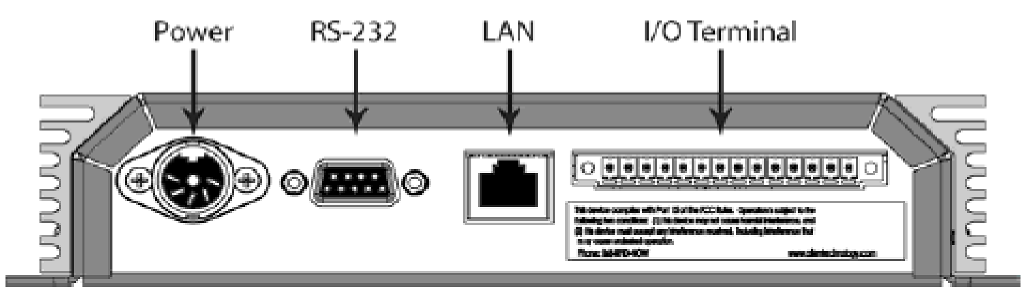


Figure 11 Alien RFID Reader (rear view)

4. Multiple Antennas for each reader. This makes a really cost effective option as the readers can have multiple antennas and for example if we are going to use a triangulation algorithm we can use a Reader with 3 antennas instead of 3 readers. The hardware takes care of avoiding interference and schedules read between antennas in a round robin

mechanism. Though we would not get a true parallel reading between all the antennas, we get readings of about 1 second apart which makes it pseudo parallel and definitely good enough for our mechanisms to locate tags. Since the antennas are more flexible, we also have a better way of physically setting up the system.

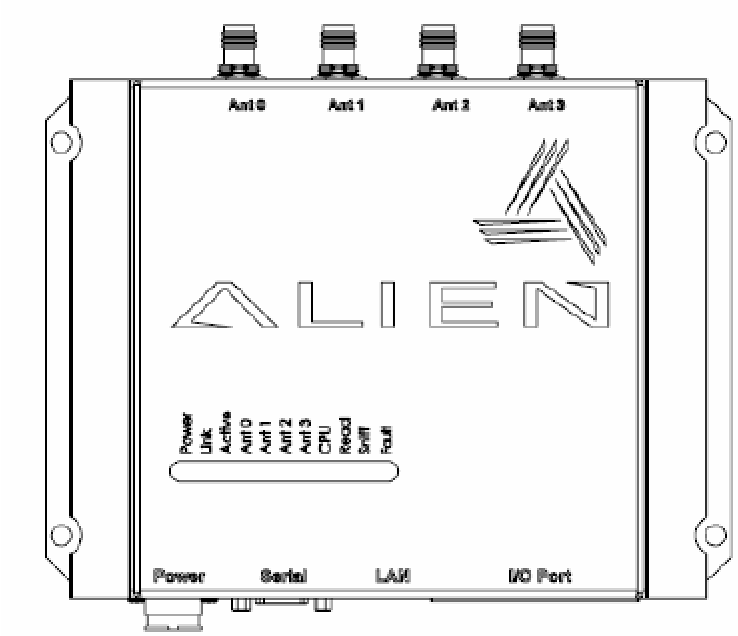


Figure 12 Alien RFID Reader (top view)

5. Protection against communication failure. The reader has the ability to store tags read and saves them if there is a communication failure, the computer can then read these saved tags once the reader is reconnected.

6. Streaming mode for reading tags. Rather than a read, wait, read cycle, the reader offers a streaming mode where the tags are reported to a pre-defined IP address, as and when they are discovered. This makes a more efficient mechanism for reading the tags through the reader.

4.3 Alien RFID Reader Software Interface (Java)

This section provides a brief overview of the software interface of the Alien RFID Reader which was used in our system. The alien reader has a customized version of the Linux 2.6 operating system running inside each reader. Though it supports both RS-232 based serial communication between the host computer and the reader we will be skipping any details regarding the same we the communication in our system happens completely through Ethernet. The RFID reader has a DHCP client which automatically acquires an IP for the reader when the reader is connected to the LAN with the host.

The reader has a web-interface which can be used to configure the reader’s various parameters. The web-interface can be accessed by simply typing the IP to which the reader is connected to, in the browser. After logging in using a username/password, the web interface presents a simple page which gives the details about the reader’s software and hardware versions as shown below in figure. The page has options to run a simple applet which will scan and report tags, edit the parameters of the reader or issue text based commands to perform various operations which can be found in the SDK.

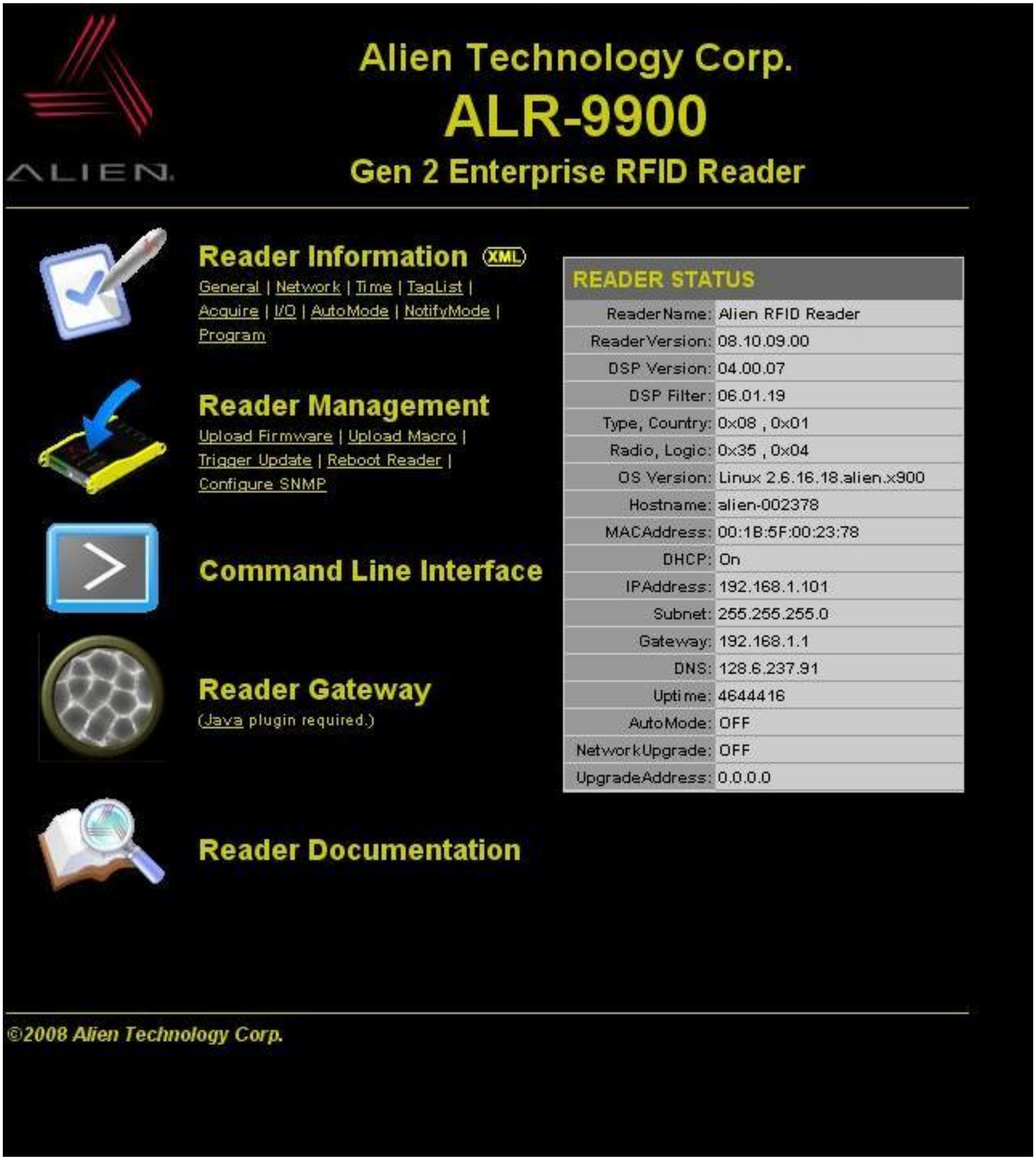


Figure 13 Alien RFID Reader Web Interface (Init screen)

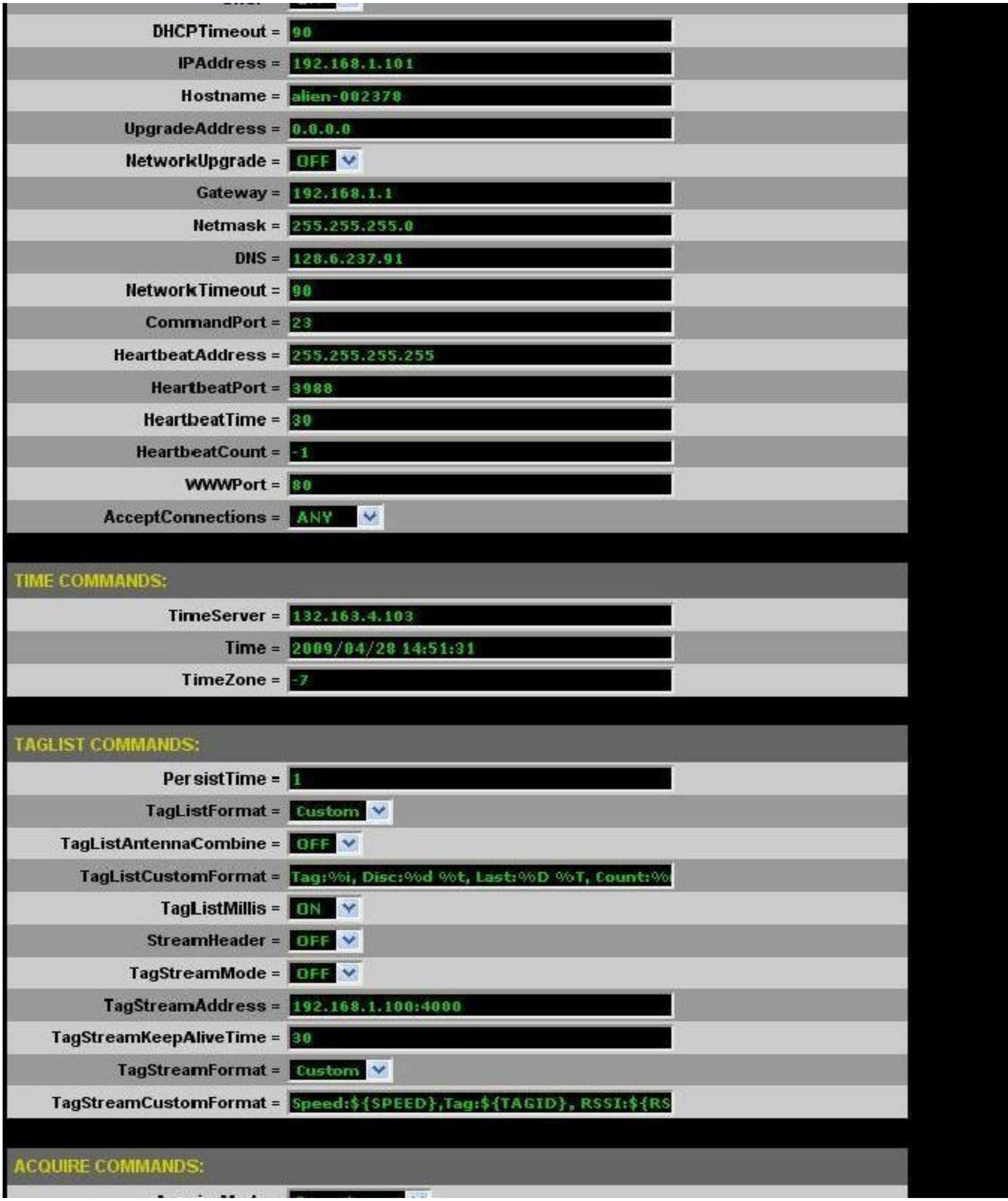


Figure 14 Alien RFID Reader Parameter Screen

Readers which are connected to the host computers LAN can be auto-discovered through a *NetworkDiscovery* class provided. This provides the software application with the readers IP address and the post on which the reader listens for commands. Once we have the IP address of the reader, the *AlienClass1Reader* class provides with a range of options to control and access the reader and its antennas. Once the connected, the commands are sent to the reader over tcp/ip. Interface commands to interact with the reader are string based and returns the response in either plain text or xml depending on the reader configuration. The Java apis for the reader provide a clean abstraction for these commands. For example

AlienClass1Reader.getTagList() method would return the current list of tags seen by the reader. The reader maintains a list of tags seen up to the time specified by the configurable parameter called Persist-Time, which tells the reader to return a given tag in the response to getTagList, even if tag was not found in the scan, up to N seconds specified by the persist-time. The Persist-Time can be controlled by *AlienClass1Reader.setPersistTime(seconds).*

The tag-list returned by the reader is a list of *Tag* objects which contains the tag-id and other parameters like its Received Signal Strength Indication (RSSI), the time it was last discovered by a scan, the antenna in which the tag was discovered and so on. The RSSI value received from the reader is an unit-less parameter which indicated the signal strength, with the higher the value greater being the signal strength. The RSSI is inversely proportional to the distance between the tag and reader among other factors which affect it and we use this property to localize tags in our system. It also gives the number of times this tag was discovered overall. The reader also gives some advance parameters like the speed of the tag's movement, its direction of movement etc.

The power level with which the scan is made can be controlled using the software. This can be done using the command *AlienClass1Reader.setRFLevel(power).* This command takes a value between 170 and 290 and sets the power with which the antenna transmits the scan for the tags. The numbers do not correlate to any standard measure and hence the range of the antenna and the it's impact on the RSSI values of the tags at a given RF level have to be discovered empirically using multiple iterations.

Since the reader supports multiple antennas, the usage of the antennas during scans can also be controlled using the api *AlienClass1Reader.setAntennaSequence(antenna numbers in ',' separated list).* For example *AlienClass1Reader.setAntennaSequence("0,1")* would only use antennas connected to ports 0 and 1. The reader cycles through all the antenna ports for a given scan and hence ordering doesn't matter. The reader has the ability to report tags seen at multiple antennas during a scan cycle, just once using the last seen antenna or report each sighting as a separate tag. We use the latter so that each antenna can be used as if it's a different reader for localizing the given axis.

The tag reading can be done in 2 ways. Manual mode and Streaming mode. In the manual mode we use the api *getTagList()* mentioned above and get the tags list whenever we need. The api is quite fast and can used in about 500ms intervals. The other way is to let the reader stream the tags to the host computer and get notified whenever there was a scan performed. This can be achieved using the *AlienClass1Reader.setAutoMode()* to true. The streaming mode sends tag-lists to a specified IP/Port which can be set to host computer's ip. This can be seen in the configuration page in the reader's web in interface shown in the figure X. The *MessageListener* class provided in the SDK acts as a listener for the messages sent to the ip/port set for auto-mode and notifies the application whenever we receive the message with the list of tags reported in that message.

4.4 Matlab Java Interface.

The prediction of distance of a tag from a given reader/antenna for a given set of inputs is done in Matlab using a neural networks as explained in 5.X In order to avoid a laborious conversion of the Matlab code, we followed an innovative approach where the prediction code runs in Matlab as a Matlab script and is interfaced to the application which uses the RFID reader apis to access/control the reader. In order to achieve this we used the Matlab Java Interface. This is a poorly documented feature of Matlab but we found it to be really useful. These classes can be found in the rmi.jar under java/jars folder of the Matlab installation directory. The Matlab java apis was used to create a bridge between the Java application and the Matlab script using the *MatlabBlockingFevalCommand* class which executes a script and returns the result of the script as a java object if the same is directly translatable into Java. The details of the bridge and its working are provided in the implementation.

4.5 Neural Network based Distance Estimation using RSSI of tags

Neural Networks has been a powerful tool for predicting non linear systems.

Our approach uses the RSSI value of the tags as reported by the readers as part of the tag scans as the criteria for prediction. This has been one of the traditional approaches in cases like WiFi positioning and we found this to be an effective means for prediction.

There are two major types of neural networks i. Feed-Forward ii. Recurrent. In feed forward networks, activation is "piped" through the network from input units to output units

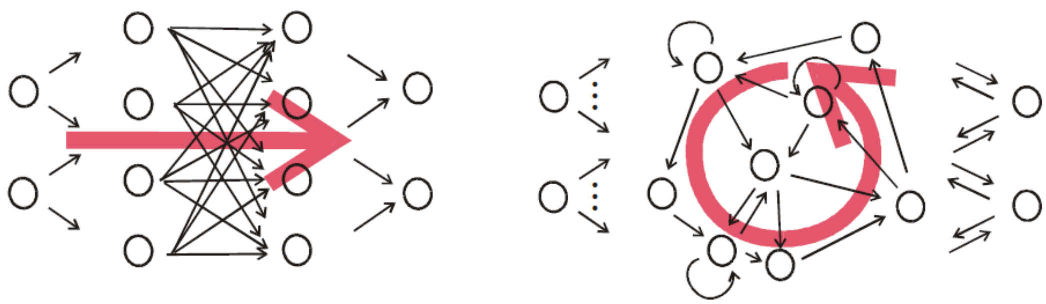


Figure 15 Feed Forward vs. Recurrent Networks

Short characterization of feed forward networks:

• Typically, activation is fed forward from input to output through "hidden layers" ("Multi-Layer Perceptrons" MLP), though many other architectures exist

• Mathematically, they implement static input-output mappings (functions)

• Basic theoretical result: MLPs can approximate arbitrary (term needs some qualification) nonlinear maps with arbitrary precision ("universal approximation property")

• Most popular supervised training algorithm: back propagation algorithm

• Huge literature, 95 % of neural network publications concern feed forward nets (my estimate)

• Have proven useful in many practical applications as approximators of nonlinear functions and as pattern classificators

By contrast, a recurrent neural network (RNN) has (at least one) cyclic path of synaptic connections. Basic characteristics:

• All biological neural networks are recurrent

• Mathematically, RNNs implement dynamical systems

• Basic theoretical result: RNNs can approximate arbitrary (term needs some qualification) dynamical systems with arbitrary precision ("universal approximation property")

• Several types of training algorithms are known, no clear winner

• Theoretical and practical difficulties by and large have prevented practical applications so far

Echo State Networks

The primary issue with RNNs is the fact that there are no practical methods for training the system due to the fact that the recursive nature of the hidden layers of the network would render the system unstable after a few iterations. The Echo State Network (ESN) is essentially a RNN which exhibit the echo state property. We consider discrete-time neural networks with K input units, N internal network units and L output units. Activations of input units at time step n are $u(n) = (u_1(n) \ldots u_k(n))$of internal units are $x(n) = (x_1(n) .. x_N(n))$, and of output units $y(n) = (y_1(n) \ldots y_N(n))$ Real-valued connection weights are collected in a N x K weight matrix $W^{in} = (w^{in}_{ij})$for the input weights, in an N xN matrix $W = (w_{ij})$ for the internal connections, in an L x (K + N + L) matrix Wout = (w^{out}_{ij}) for the connections to the output units, and in a NxL matrix

$w^{back} = (w^{back}_{ij})$

for the connections that project back from the output to the internal units. Note that connections directly from the input to the output units and connections between output units are allowed. No further conditions on the network topology induced by the internal weights(W) is imposed (e.g., no layer structure). We will also not formally require, but generally intend that the internal connections W induce recurrent pathways between internal units. Without further mention, we will always assume real-valued inputs, weights, and activations.

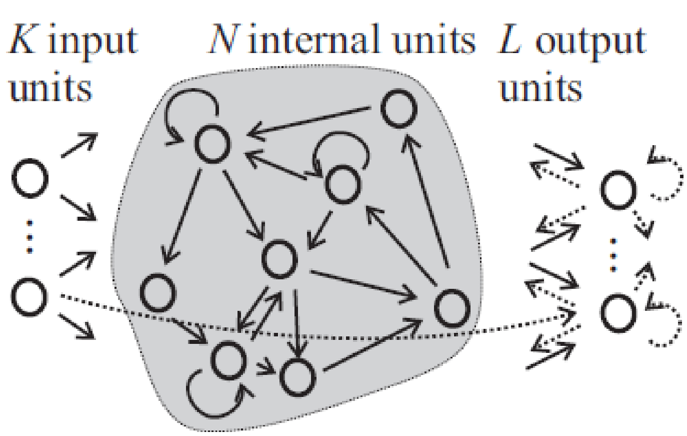


Figure 16 The basic network architecture assumed in this article. Dashed arrows indicate connections that are possible but not required

The activation of the internal units is updated according to

$$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{\mathrm{in}}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{\mathrm{back}}\mathbf{y}(n))$$

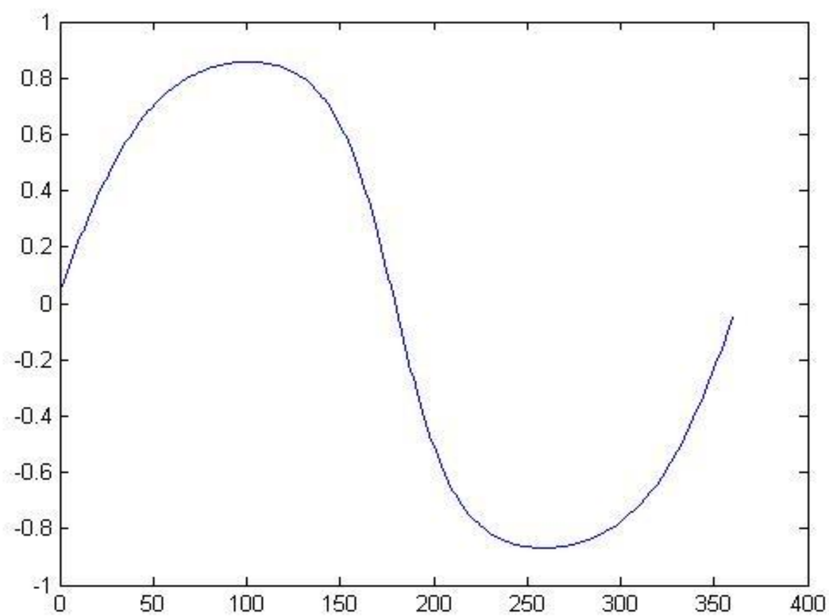
where f = ($f_{1, \ldots,} f_n$) are the internal unit's output functions (typically sigmoid functions).

The output is computed according to

$$\mathbf{y}(n+1) = \mathbf{f}^{\mathrm{out}}(\mathbf{W}^{\mathrm{out}}(\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{y}(n))$$

where $\mathbf{f}^{\mathrm{out}} = (f_1^{\mathrm{out}}, \ldots, f_L^{\mathrm{out}})$ are the output units functions and $(\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{y}(n))$ is the concatenation of the input, internal and previous output activation vectors.

The ESN network performs really well in modeling time series. Like for example sine wave. The below figure shows a sine wave generated using the ESN network we have used in our



system.

Figure 17 Sine wave generated using ESN

We found the accuracy of the system marginally improves if we provide the estimator with a series of RSSI values and averaging the estimated distances than using a single

RSSI value. Hence rather than estimating for every single discovery of the tag we instead take a series of values and estimate the distance for the series and final position estimation is given by the average of the estimated distances.

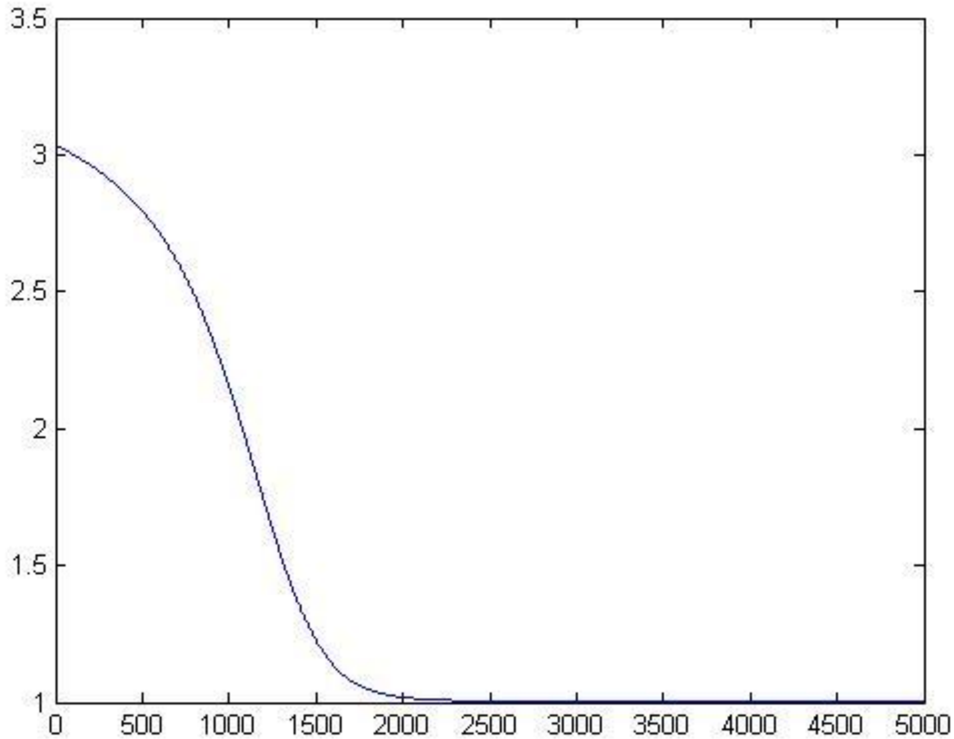


Figure 18 Estimated RSSI vs. Distance using the ESN network

5. Implementation

5.1 The Setup.

One of the critical parts of the implementation of the system in the placement of the antennas of the readers. The antenna of any RFID reader available in the market has an elliptical pattern in which it can scan for tags. The pattern is not smooth and has irregular edges on the periphery of the ellipse. We had to identify the ideal setup through trial and error mechanism. At each stage we used the read range of the reader and the smoothness of the RSSI value received from tags at different distances as a parameter for finding an effective setup. The final setup which we found was effective enough was having the antenna at a height of about 3ft above the plane in which we want to read the tags and slanted at about 25 degrees. This gave us an effective range of about 2m and a smooth RSSI curve for tags at different distances.

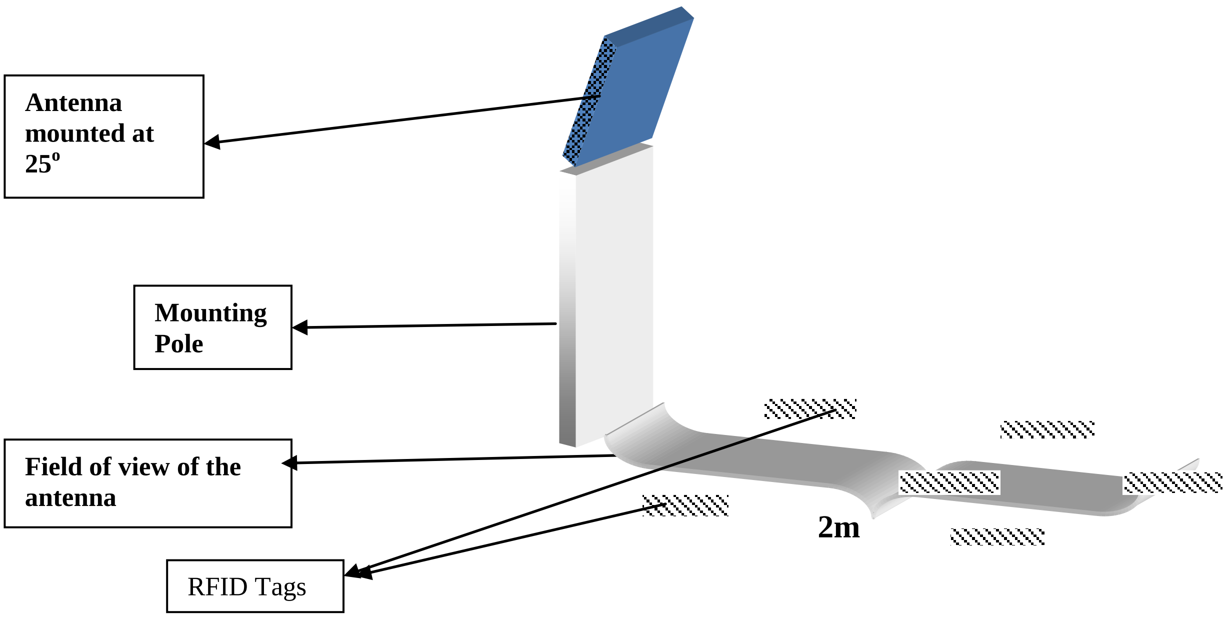


Figure 19 Antenna arrangement

The mounting pole does not need to be a ‘pole’ specifically. It can be any stable point which is elevated a few feet above the intended Field of view of tags. In our lab we have used the antenna fixed to cardboard boxes using duct-tapes!.

Each reader was connected to two antennas and the antennas were placed in the middle of a square with about 2 m sides.

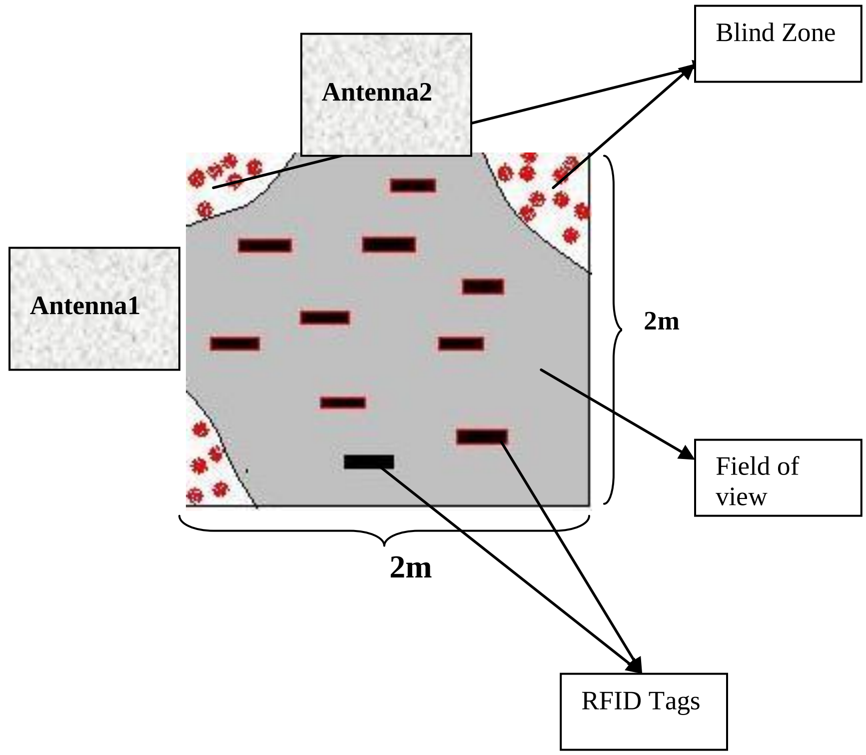


Figure 20 Both Antennas and Field of view with tags

The below figure shows the placement of an array of antennas to cover a larger area. The information about the antenna placement is provided to the software implementation using a configuration XML. The XML file contains the coordinates of the antenna and also information whether this is an axial antenna or an vertical antenna.

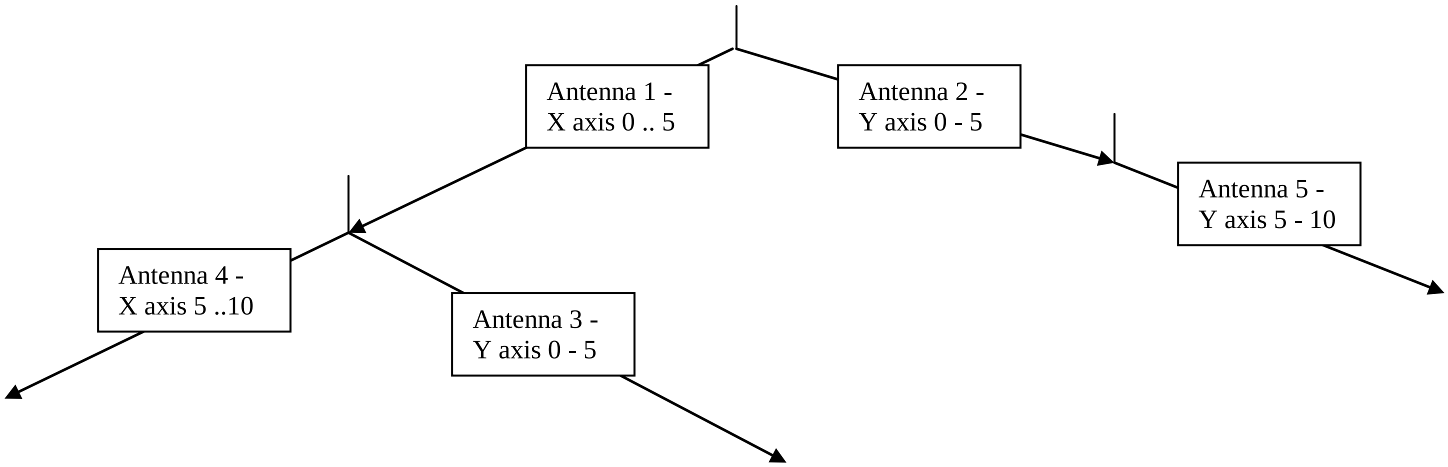


Figure 21 Array of antennas

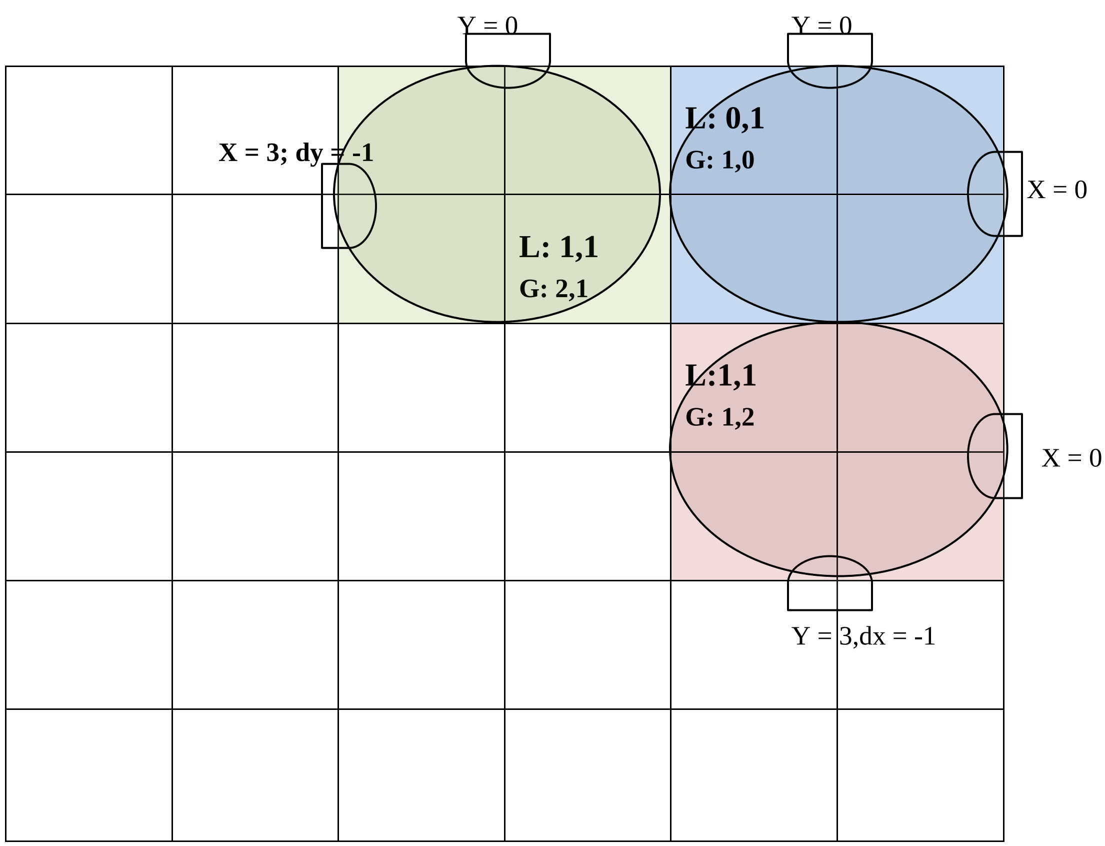


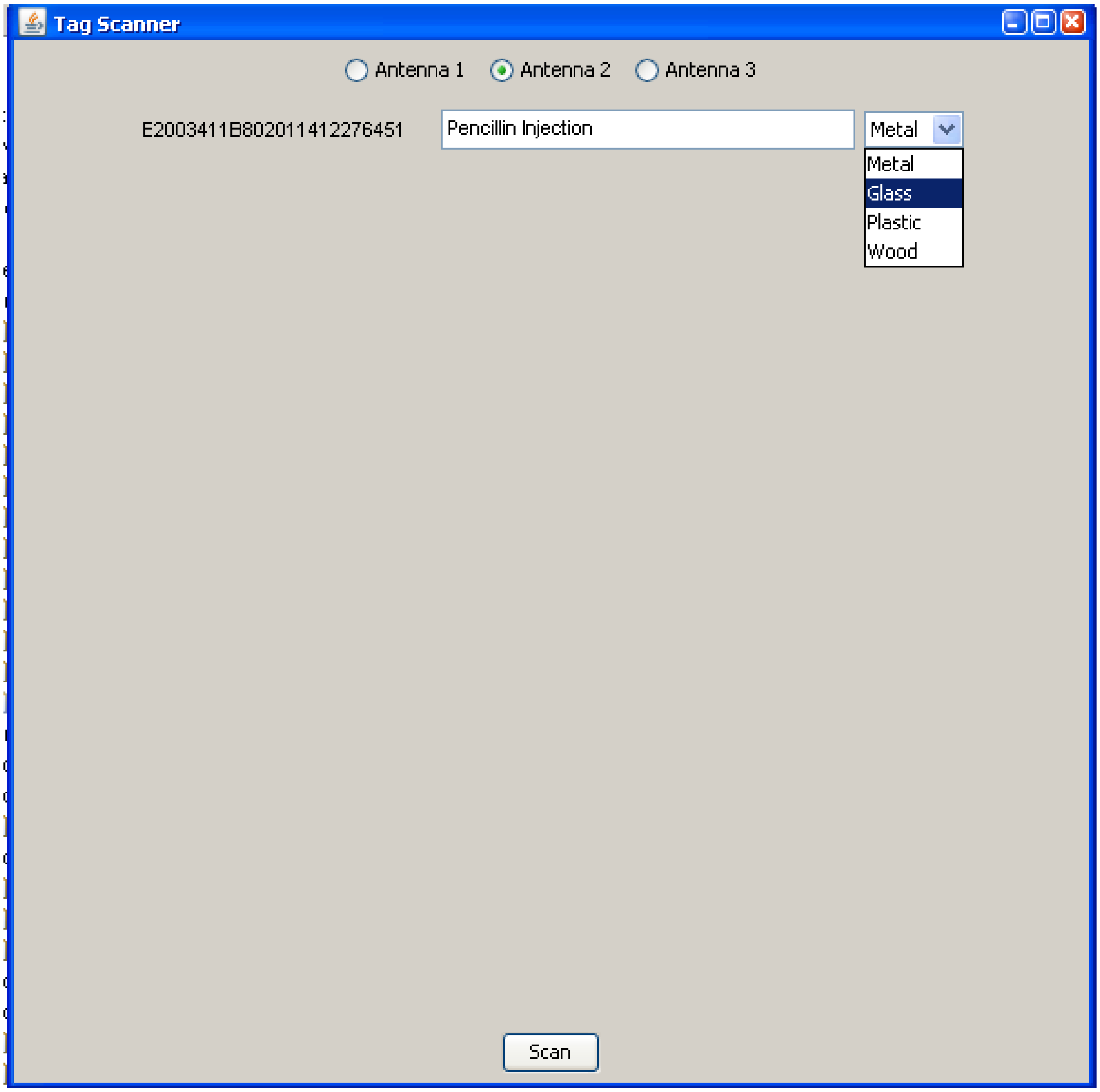
Figure 22 Mapping local coordinates in antenna array from tag distances to global coordinates

5.2 Software Implementation

The software implementation of the system was done in Java using the Java based library provided by Alien for the RFID readers. The software implementation can be divided into three main components.

1. Tag description scanner.
2. RFID scanner
3. Distance Estimator and the Java-Matlab bridge
4. Tag location and visualization

5.2.1 Tag description scanner.



Like maintaining an inventory in a store, when the tags are first fixed to an object the object has to be scanned using the scanner application in the system. The user can then select the material type on which the tag is stuck and the description of the object. This is then stored in a MySQL database.

5.2.2 RFID Scanner:

This module primarily consists of the RFIDReader class which records the tag stream sent to the host computer by the RFID reader. The stream is read using the MessageListener class provided. The stream consists of messages which contain a list of Tag objects. On receipt of the messages they are simply stored in a queue which is processed by a background thread in the TagEventHandler class. This thread later groups the tags into clusters based on the tag-ids using a hash map mapping the tag-id to the tag information represented by a DBTag object.

The TagEventHandler class stores the tags in a Hashtable with the tag-id as the key. Each entry consists of an array of Vectors which has one entry for each of the antenna used. Each entry in this array further consists of an Vector(array) of Tag objects found in each iteration. The messages from the reader combine all the antenna's scans into one message. Since each antenna represents one of the axes of the 2D plane in which we localize tags, the grouping by antennas makes it easier for further processing.

5.2.3 Distance Estimator

The tag collection above is processed by another thread in the RFIDReader which extracts the RSSI values of the tags from the vector tag objects for each antenna. This array of RSSI values are sent to the Matlab based predictor using the Java-Matlab bridge explained below.

Java-Matlab Bridge

The bridge in brief is similar to an RPC call and returns an array of predicted distances for the given rssi values. Since Matlab does not provide with a library that can be used from Java directly to execute Matlab scripts, we had to come up with an alternate way to avoid rewriting the Neural Network code we had available on Matlab. The bridge is quite generic enough and can be modified easily to make it 'generatable' from a stub just like rpc-gen.

Why a bridge ?

The primary reason for a bridge is the fact that Matlab doesn't really have a clean Java api that can be linked to applications for executing Matlab scripts. The complete Matlab engine has to be started using the desktop based application and cannot be instantiated using api calls from other Java applications. Matlab on the other had has the convenience of being able to run Java applications from the Matlab command shell. And thirdly there is the rmi.jar library that contains the apis for running Matlab commands from Java applications. So the bridge is a small overlay on top of these components to create a seamless "rpc like" communication layer.

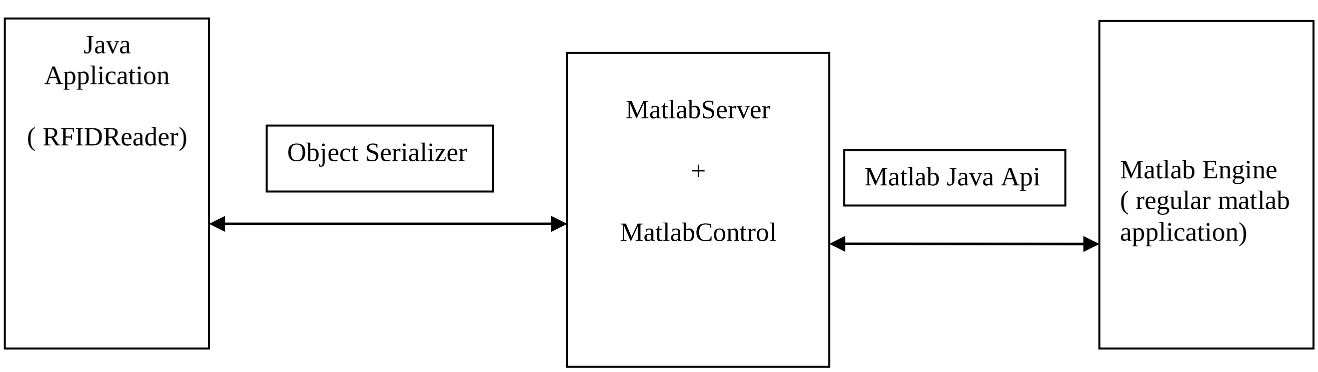
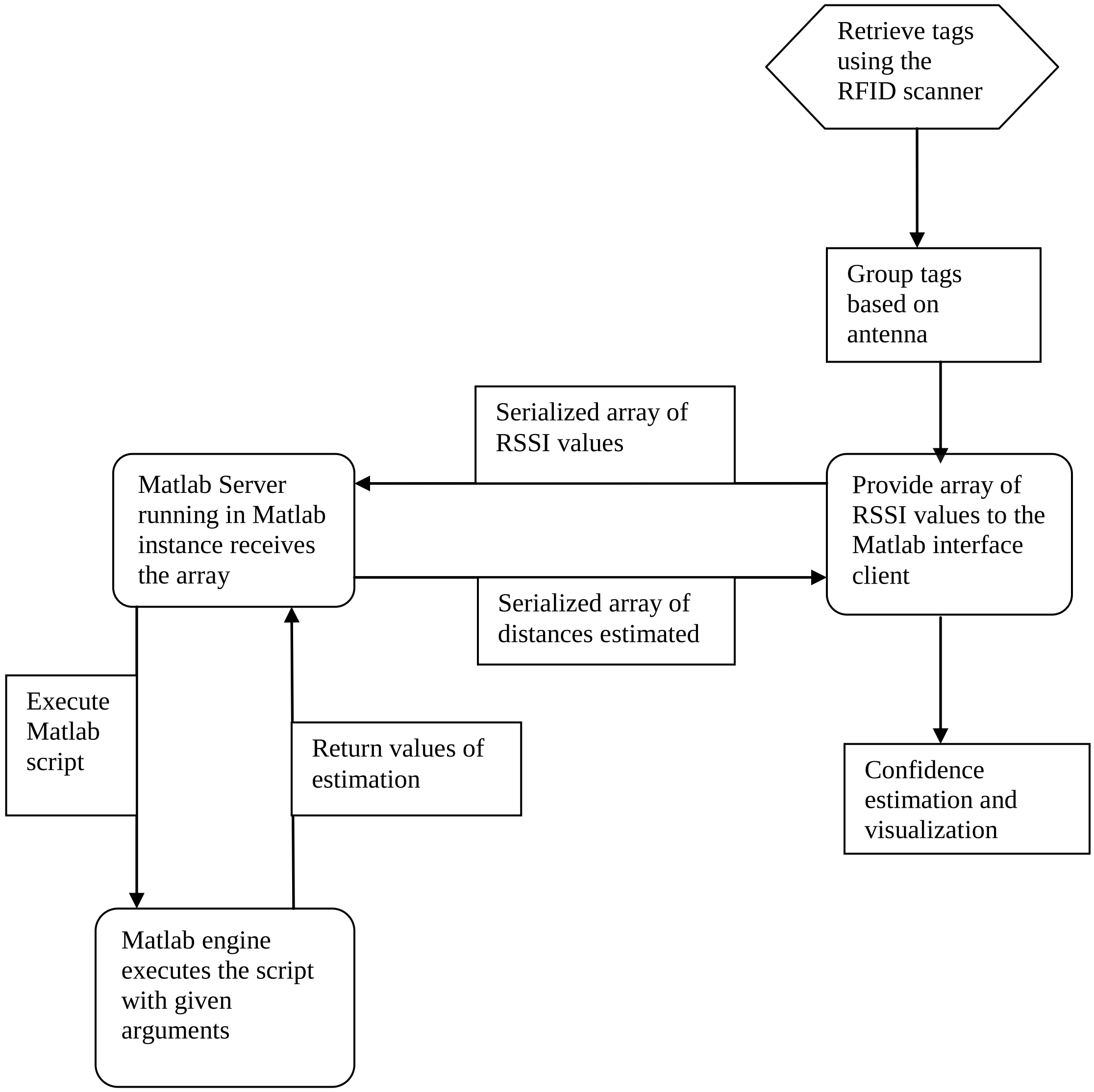


Figure 23 Java-Matlab Bridge

The interface to the Java application in this case the RFIDReader class is a simple api that takes an array of rssi values and returns back the predicted distances. The rssi values are communicated to the MatlabServer using regular TCP streams by serializing the array object using Java's native object serializer (ObjectOutputStream) . Using this we can transfer any complex data structure to the MatlabServer. The results from the MatlabServer are read using Java's native object deserializer over the TCP stream (ObjectInputStream).

The reason that in order to use the Matlab engine from Java we need to start the java application from inside the Matlab command line is the primary reason for this bridge. The MatlabServer is a simple sever listening on a TCP port. It receives the name of the Matlab script to run and the parameters as serialized java objects from the client. It gets an instance of the current Matlab engine using the Matlab class. The script names and the parameters received from the client are executed on the Matlab object using the method *mtFevalConsoleOutput*. Calling this method is like executing a Matlab command with the given parameters from the console window of Matlab, it execute the command and prints any console output to the Matlab command line window and returns the results. Any Java object can be returned as long as the class is in the class path of the Java application as well, and this is just a basic JVM rule and not a short coming of the bridge. In our case the predictor script returns an array of double values representing the distances for the given RSSI values. This is serialized and sent back to the Java application.

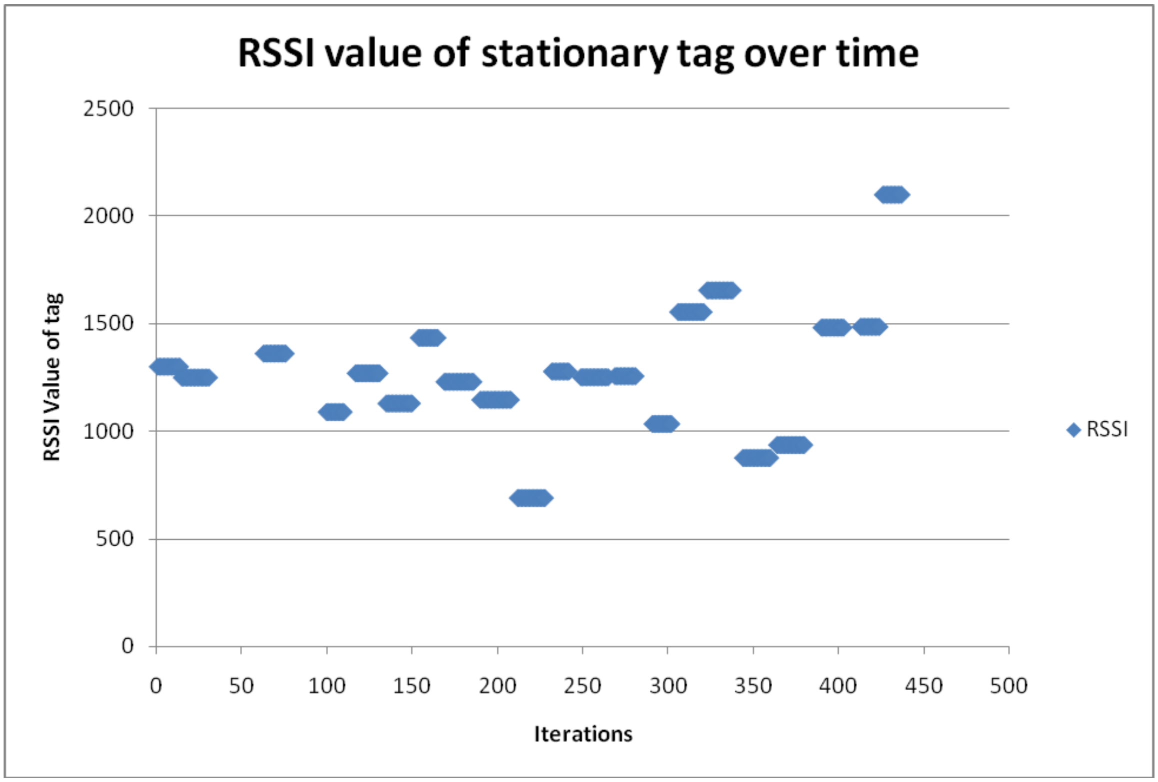


5.2.4 Tag Locator

The array of distances obtained from the Matlab based predictor is fed to another thread called the TagLocator. This thread does two key things. First we found that the RSSI values are quite inconsistent so we had to come with a quick and efficient method for outlier and secondly this thread translates the distances values into a Cartesian value which can be used to plot by the visualizer module.

Outlier Removal

The RSSI values obtained though indicative is quite inconsistent and can have a wide range. The below figure is an example from one of the data collected.



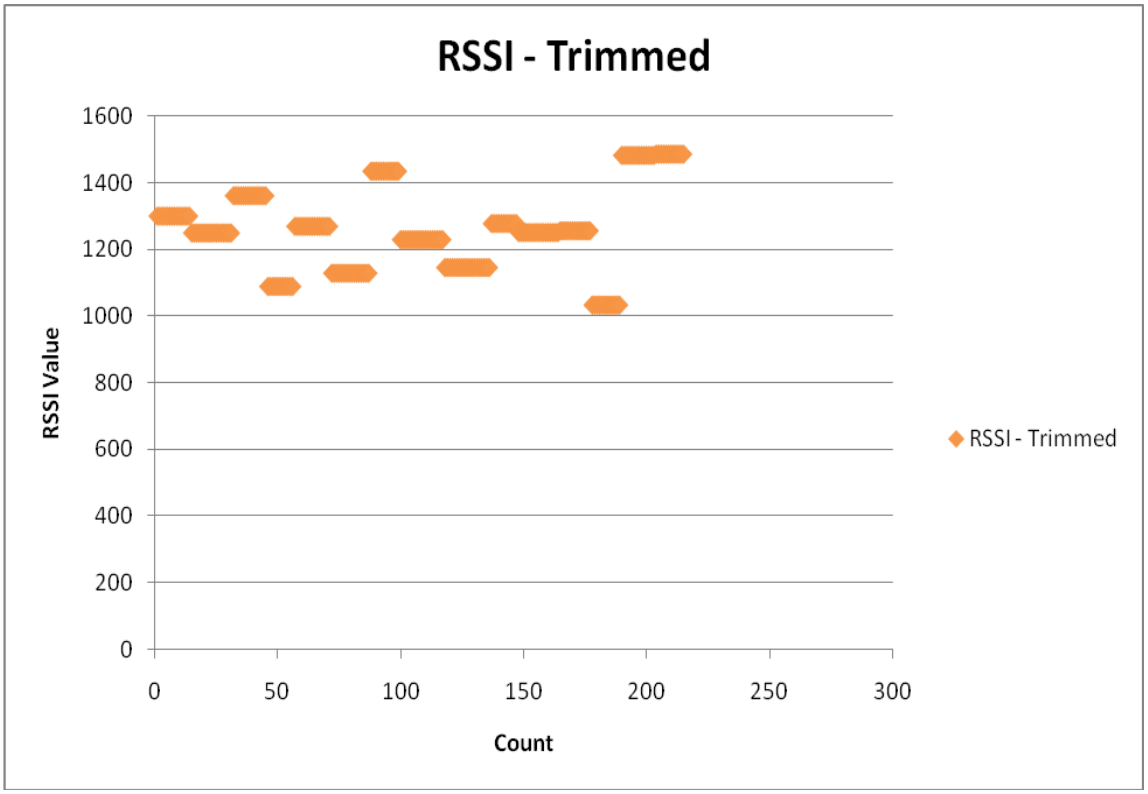
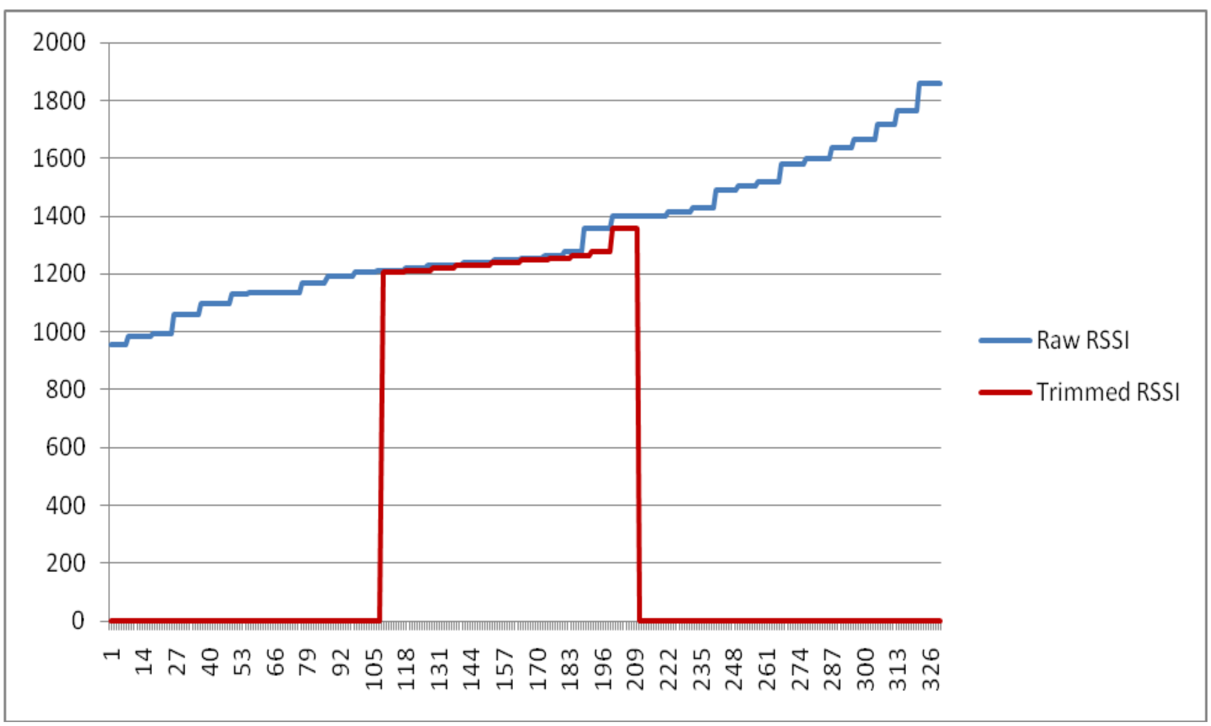


Figure 24 RSSI values of tags at the same distance



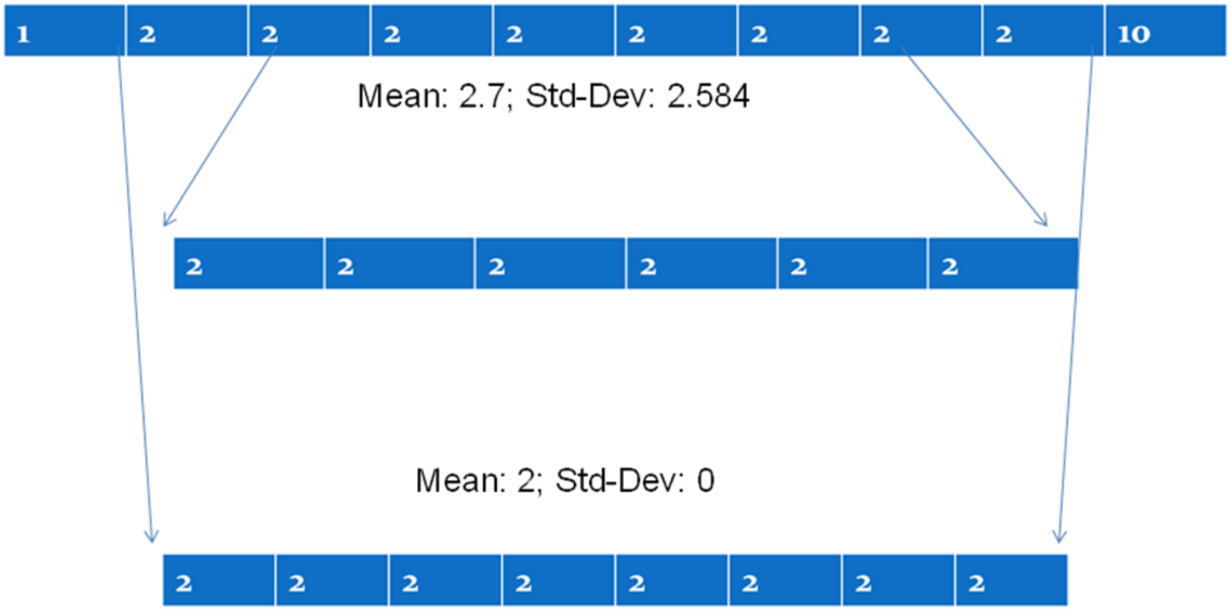


Figure 25 Filtering data using trimmed mean and standard deviation of trimmed data

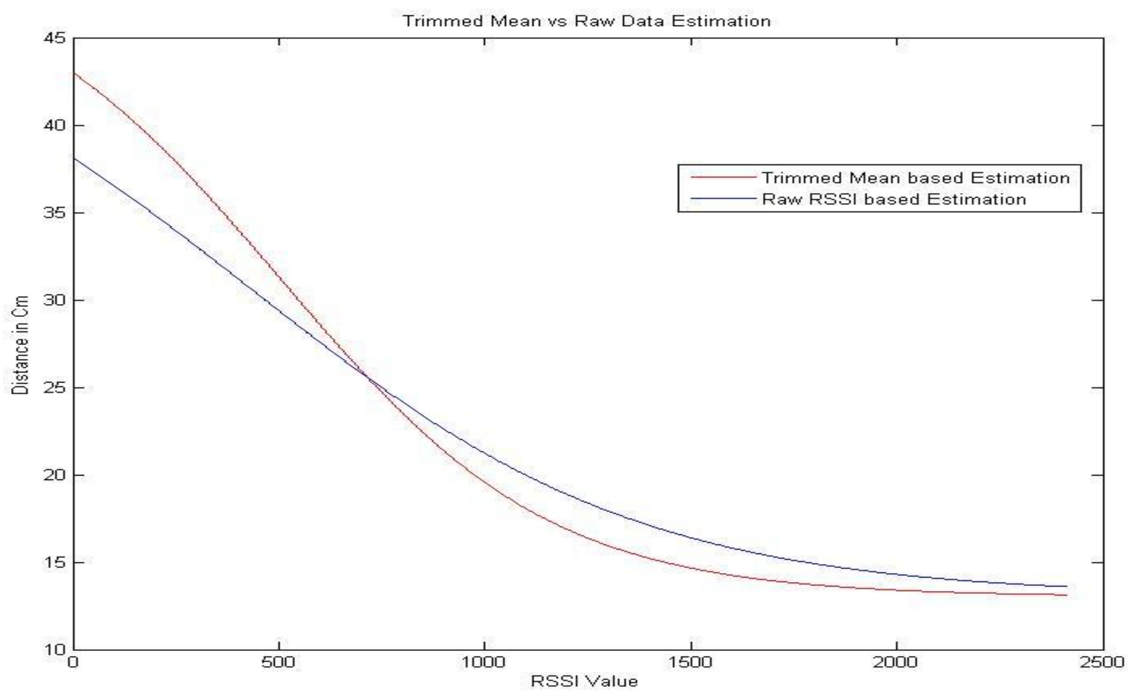


Figure 26 RSSI vs Distance curve from estimator

We calculate the “trimmed-mean” which is the mean of the data sample within a given confidence interval. We empirically identified the interval as between 0.25 – 0.75. We now find the mean of this interval, which would be the trimmed mean and would represent the central mean of the data sample. We then find the standard deviation of this sample of the data. Now any value in the entire data sample which exceeds the standard deviation of the central sample is ignored. This is better than just choosing the central range of values as we may have valid data point across the set.

This was found to improve the accuracy considerably.

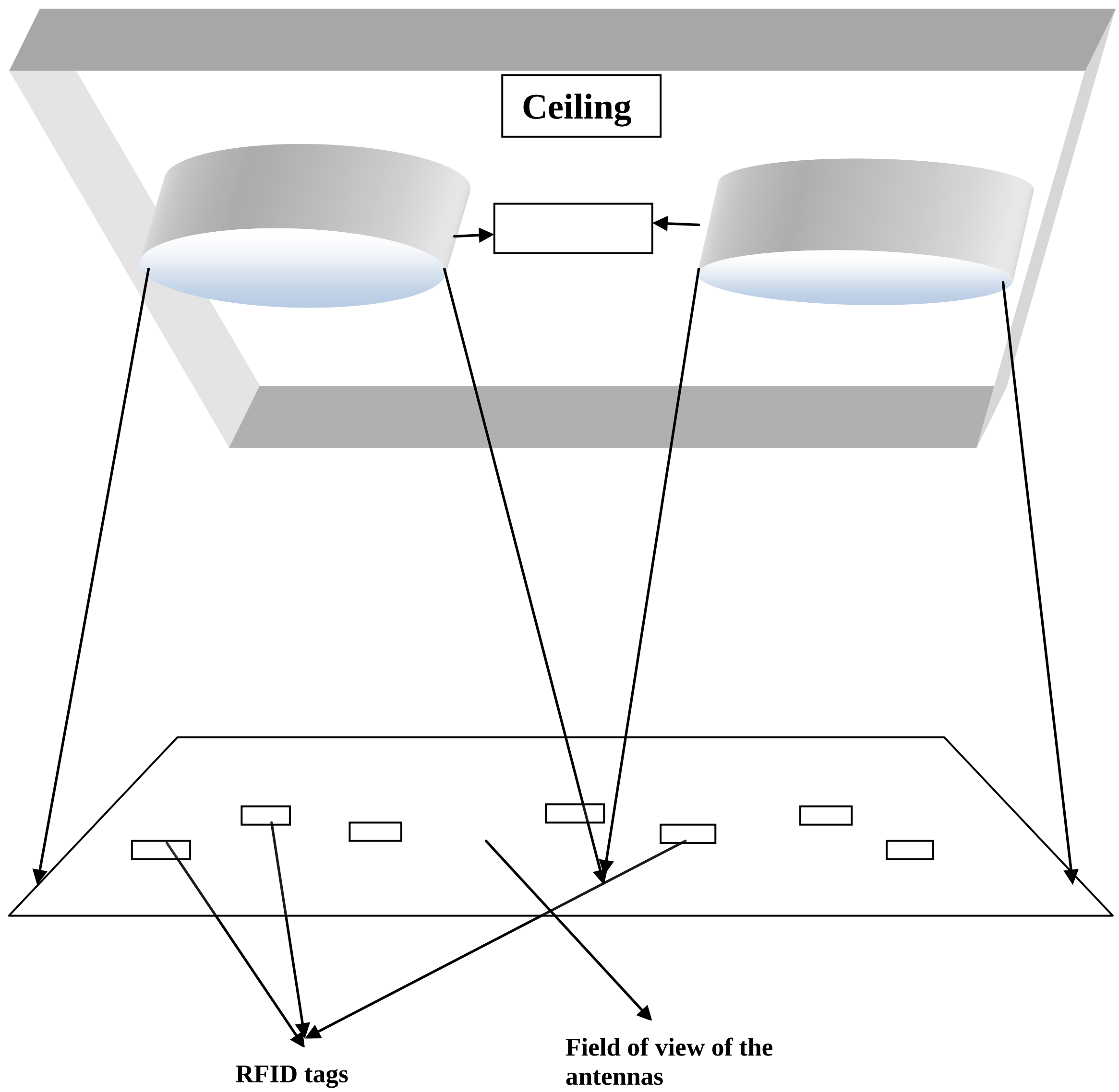
Read Count - Secondary Parameter of Estimation

The number of times a tag is read over a set of iterations is called the read count and read count can be used as an alternative parameter of estimation of distance. When tags are grouped closed to each other or when the tag is occluded by an object the RSSI value of the tags drop quite significantly. This results in an erroneous estimation of the distance of the tags. The class PositionAdjust does exactly this by maintaining a table of read counts of each tag. A new position is calculated using the read count. The calculation of position is just a lookup table which maps the percentage of successful reads to coordinates.

The issue with erroneous tag positions only occurs with loss in signal strength. A case of a given tag having much higher RSSI reading when it’s at a distance was almost nonexistent. Hence the PositionAdjust is used only for tags which are estimated to be more than 2 units away. If the PositionAdjust indicates that the tag is closer by the virtue of the read count this overrides the RSSI based estimation. The read-count based estimation is largely oblivious to occlusion as the tag is anyways read even if its occluded only that the signal strength is lowered. The speed of the tag is monitored using the hardware and the counters for measuring the read-count percentage is reset if the tag had moved or if the tag is invisible for a large period.

The Third Antenna based Approximation:

This novel approach provides a third level approximation for the system. We use a third antenna vertically over the field of tags.



This antenna helps us identify the tags within the 3mX3m block which the TagLocator is focused on. This can help us in scenarios where we can identify if the given tag was brought into a given area or not though this currently cannot be used to localize tags within the area of interest.

Figure 27 Vertically mounted antenna on the roof

5.2.3 Tag Database

Even though the system can track any tags without prior knowledge, just the tag-id would not make sense to the end user. Hence we create a database of tags with other meta information in it. The meta information includes information like what is the item on which the tag would be or has been tagged, what is the material on which the tag has been fixed like plastic, metal etc.

The meta information in the database is later used to display information to the user when the tag is located when the system is running. The tag database in addition to the meta information also contains another database which is a detailed log of all the events that happen while the tags are being scanned. Each event here would be locating of a single tag. The tag logging database consists of information like the tag-id, the RSSI, the timestamp of the localization, the speed of the tag etc. Also along with this information it is also logged whether if the localization information is from an active reading from all antennas or from previous known location.

The tag DB is maintained as a MySQL database and is interfaced through the MySQL Java Database Connection (JDBC) interface. The meta information is stored as a Full Text Search database table, which makes the meta information “searchable”, i.e. we can get a list of tags in the database which matches a particular keyword.

The information is added to the database through the Scanner class, which lets the user select a given antenna and displays the tag, which is held close to it. Once a tag has been successfully read through the selected antenna, we can then enter the meta-information and this gets stored in the above-mentioned DB. The Scanner module should be used while the localization system is not running for obvious reasons like interference or used with a reader/antenna, which is not used for localization.

5.2.4 Tag Location Visualization

Once the tags coordinates are estimated by the Tag Locator module, the tag-ids and the coordinates are passed onto the graph visualization module which is handled by the class PlotTags. PlotTags is passed a Hash-Map consisting of the tag-id as the key and the tuple of the antenna where the tag was found and the distance of the tag from that antenna. Since each antenna corresponds to an axis in the graph the same is translated into a 2D graph. The tags are represented by their tag-ids and with a color code and the accuracy range of the localization.

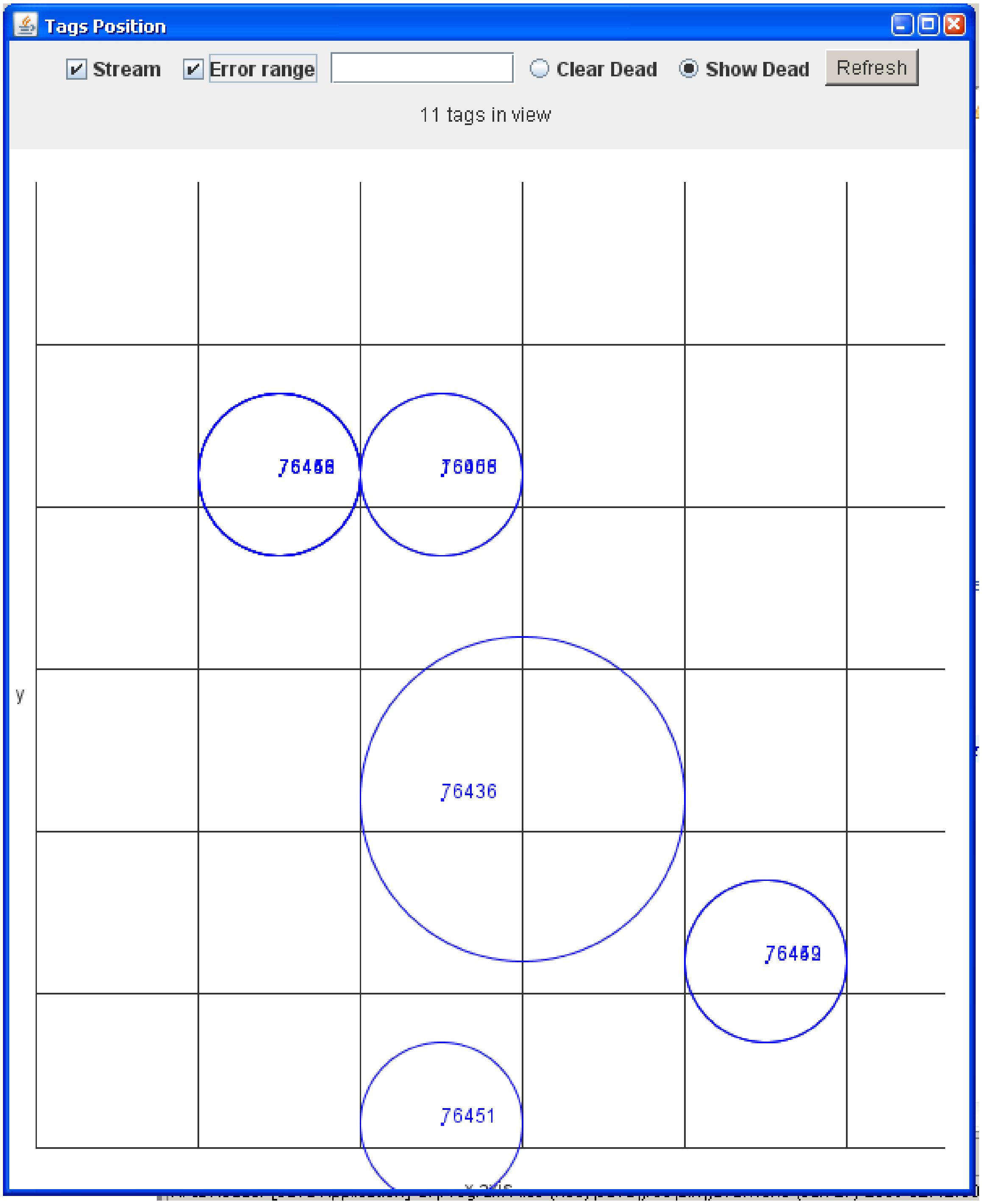


Figure 28 Tag Position Visualization

The UI has the ability to track a given tag based on the ID. The below image shows the tracking of a single tag based on the ID. Along with the tag-id we can also track tags based on matching meta information. An example of this would be , if we have a tag which was earlier identified as insulin. We can also track all tags which have “insulin” in their meta information

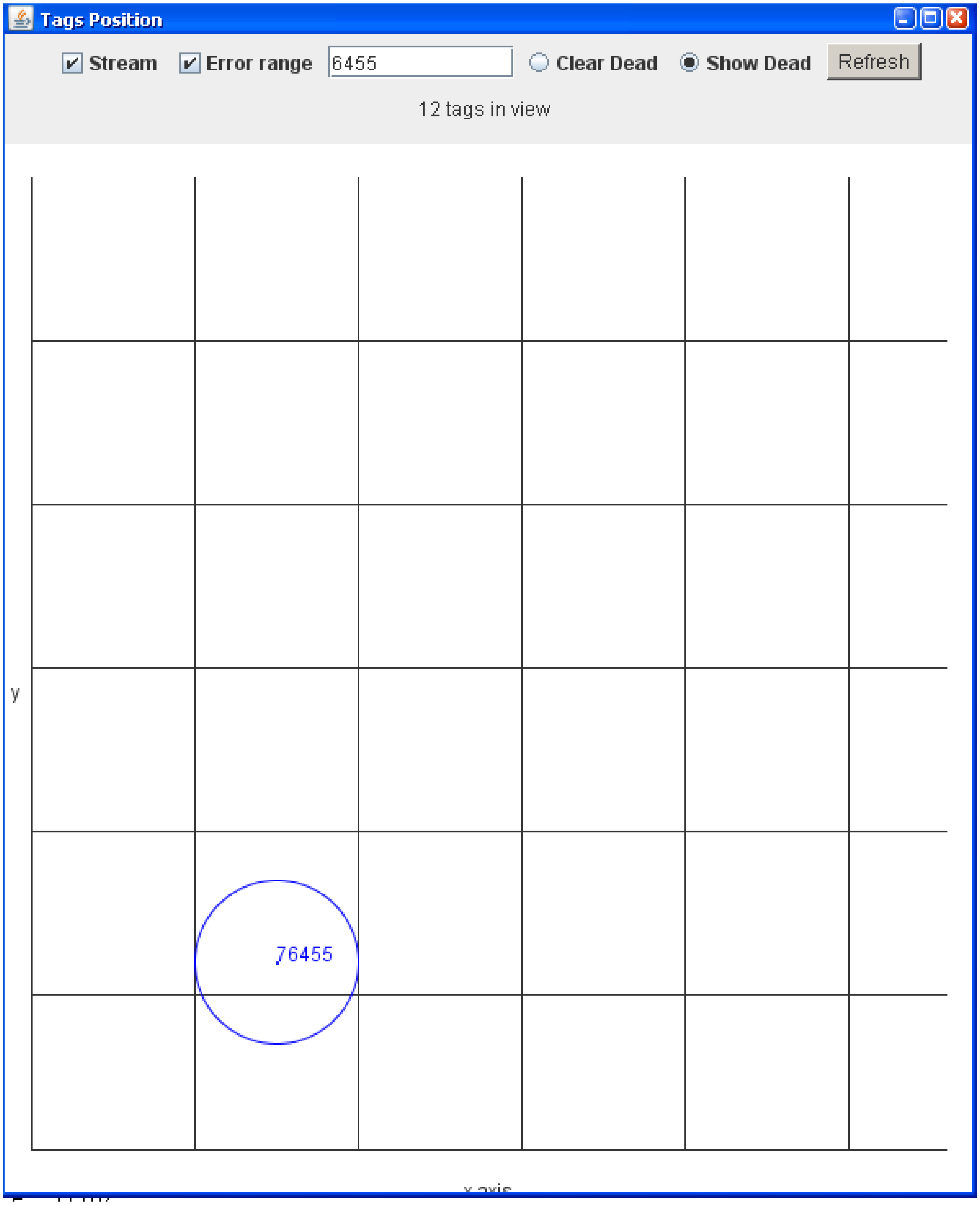


Figure 29 Tracking a single tag based on ID

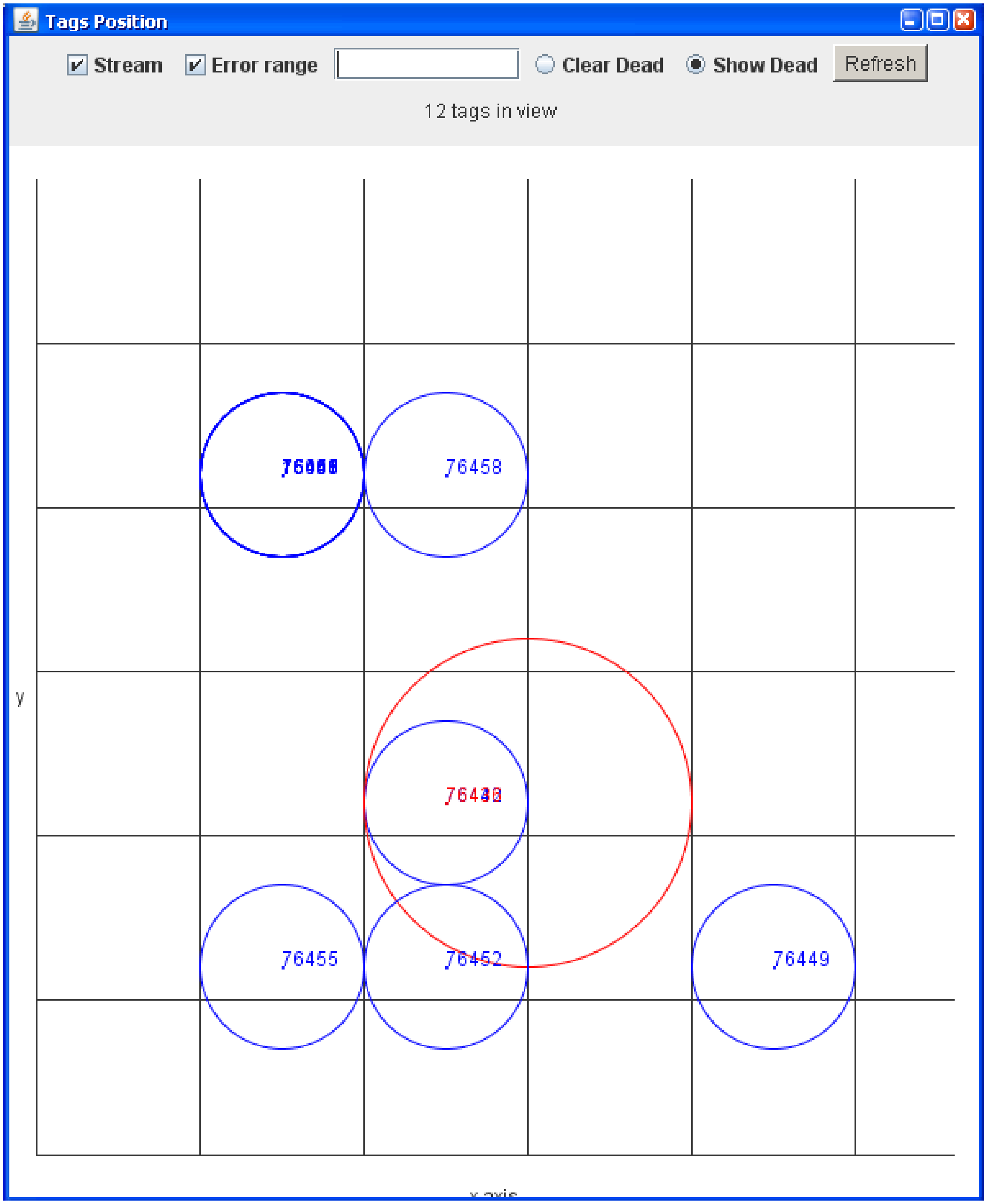


Figure 30 "Dead tags" shown at the last known position.

6. Results and Observations

During the course of development of our system we observed several interesting points which leads us to believe that this is an effective method for localizing RFIDs and definitely is a very useful tool in the field of trauma care and in hospitals in general.

Localization of each tag takes less than a second. The following is the split of the various stages of the localization. The values given are rounded average values over a few 100 iterations

Reader Scan for tags - 50ms

Matlab Estimator - 20ms

Rest of the Software - < 1ms

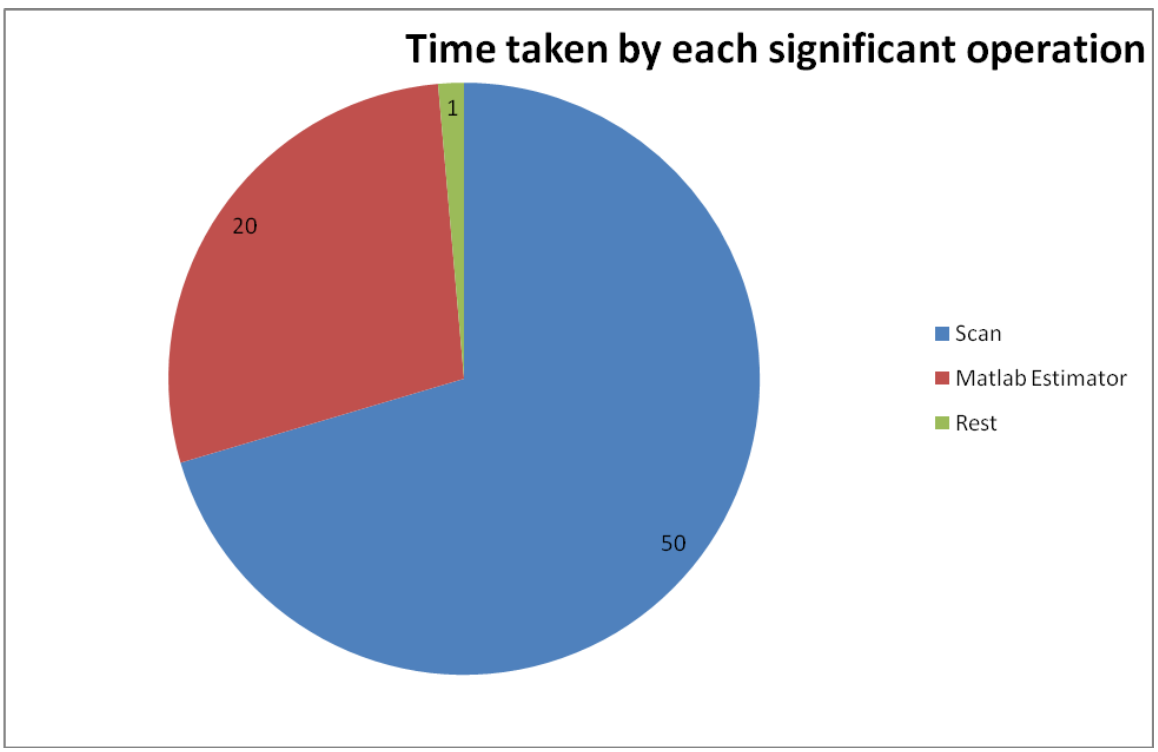


Figure 31 Time taken by each operation

Tags are reported to the reader in batches so rather than reading one tag per 50ms there are a set of about 5 tags scanned every 250ms on a average.

The accuracy of the system varies as per the operating environment. There are various factors that affect the accuracy of the tag localization.

i. Material on which the tag has been stuck
ii. Obstacles between the tag and the antenna
iii. Distance between the tag and the antenna.
iv. Concentration of tags.

Background material:

We found that different materials change the RSSI value of the tags and the relative signal strength of tags stuck on various materials stuck on different materials is given below. In our case the tags were stuck on a homogenous material and hence we have currently ignored using the material to affect any of the estimated parameters of the system.

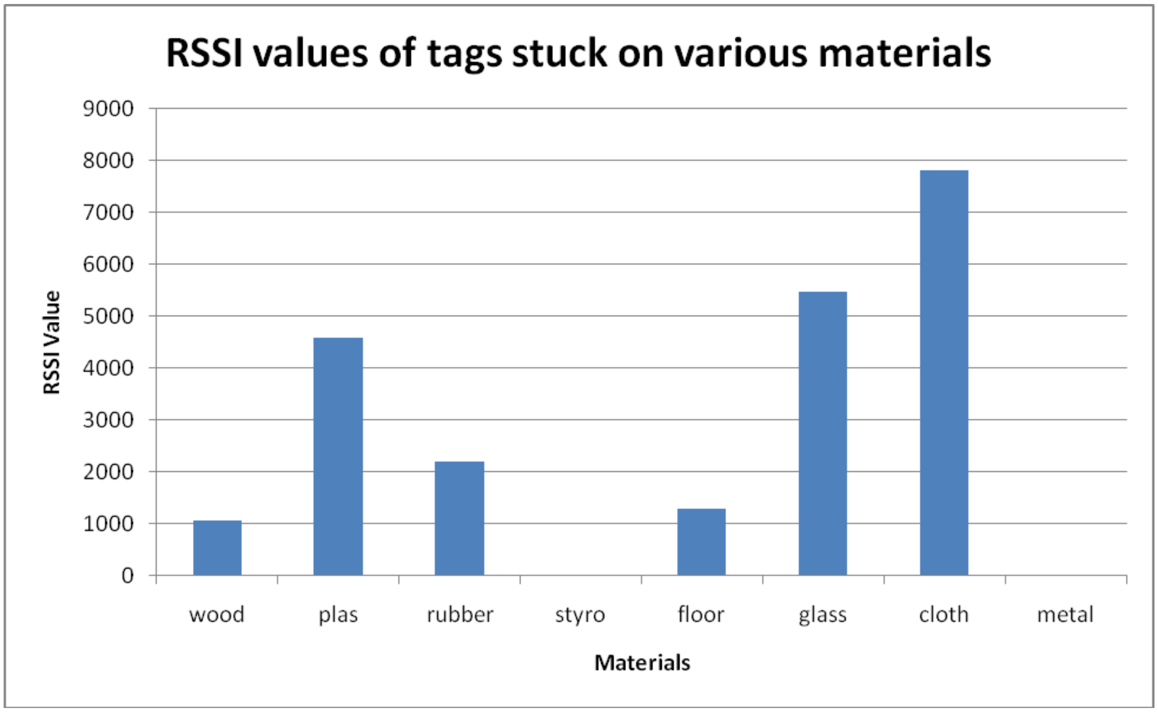


Figure 32 RSSI values of tags at same distance with different background materials

The higher the signal strength of the tag based on the background is longer is the range of the reader in tracking such tags. In a heterogeneous system with tags being attached to various such materials we have studied the using a "material-scaling" model though it has not been implemented. Material scaling assumes the fact that the tags we use are stuck to a particular object once and are not transferred. With passive RFID tags being relatively cheap there would be no need to reuse tags in any scenarios. Material scaling involves in identifying the background material a tag based on a static database and normalizing the signal strength received from the reader to the lowest common denominator. In the above graph we can observe that wood has the lowest signal strength with 1000 and glass has a RSSI value of 5500 for the same distance. Based on our experiments we were able to observe that this rough scaling (5.5 in this case) holds good consistently. Hence a tag stuck to a wooden object with RSSI value of 1000 is at the same distance of a tag stuck onto a glass object with RSSI of 5500.

Obstacles:

The ideal environment for a passive RFID tag is an obstacle free path with the tag vertically on the perpendicular line to the center of the antenna. Unfortunately we cannot achieve an obstacle free path. Hence our novel method to overcome this was using a third antenna to augment the two antennas that are used for estimation.

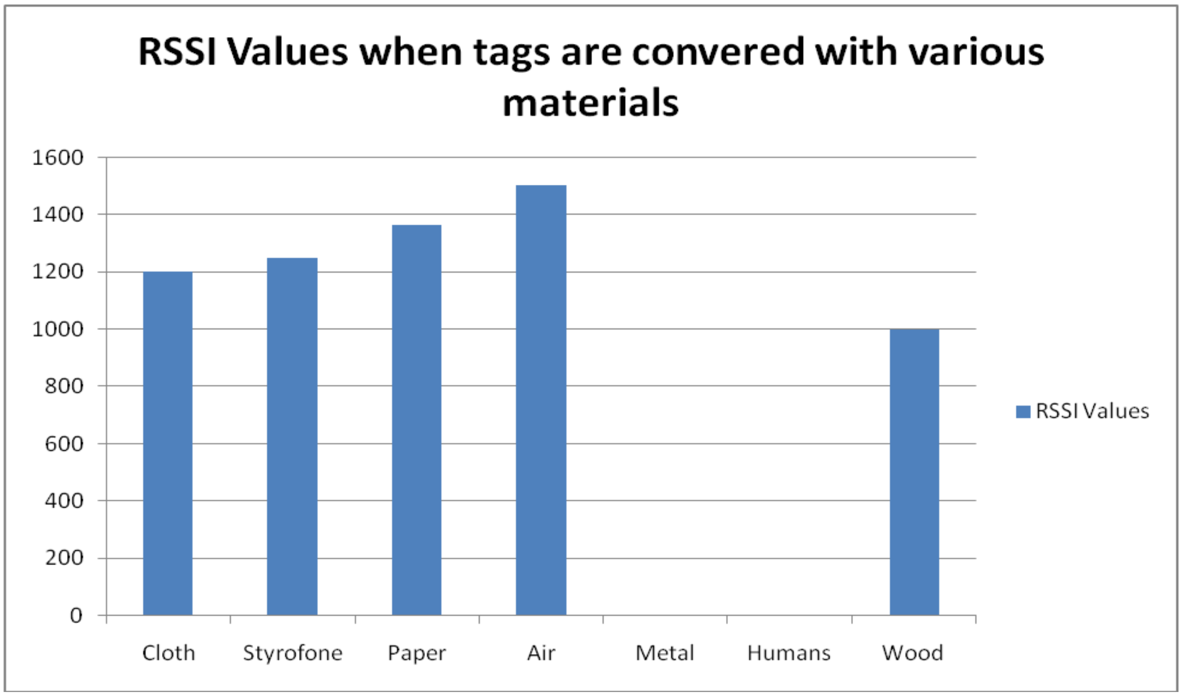


Figure 33 RSSI values vs. obstacles

Distance between the tag and antenna:

In an obstacle free environment with low tag concentration the accuracy of the system is ≥90%. The accuracy is nearly 100% if the tag is within 2 block. As in any localization system the accuracy drops at boundaries of 2 different blocks

Concentration of tags:

The concentration of tags affects the RSSI value of the tags quite significantly. This drop happens at all distances consistently. The below chart shows the comparison of the RSSI value to the number of tags in each small area. The tags here were spread within an area of 1Feet X 1Feet. The RSSI value shown here is of the tag with the lowest RSSI value.

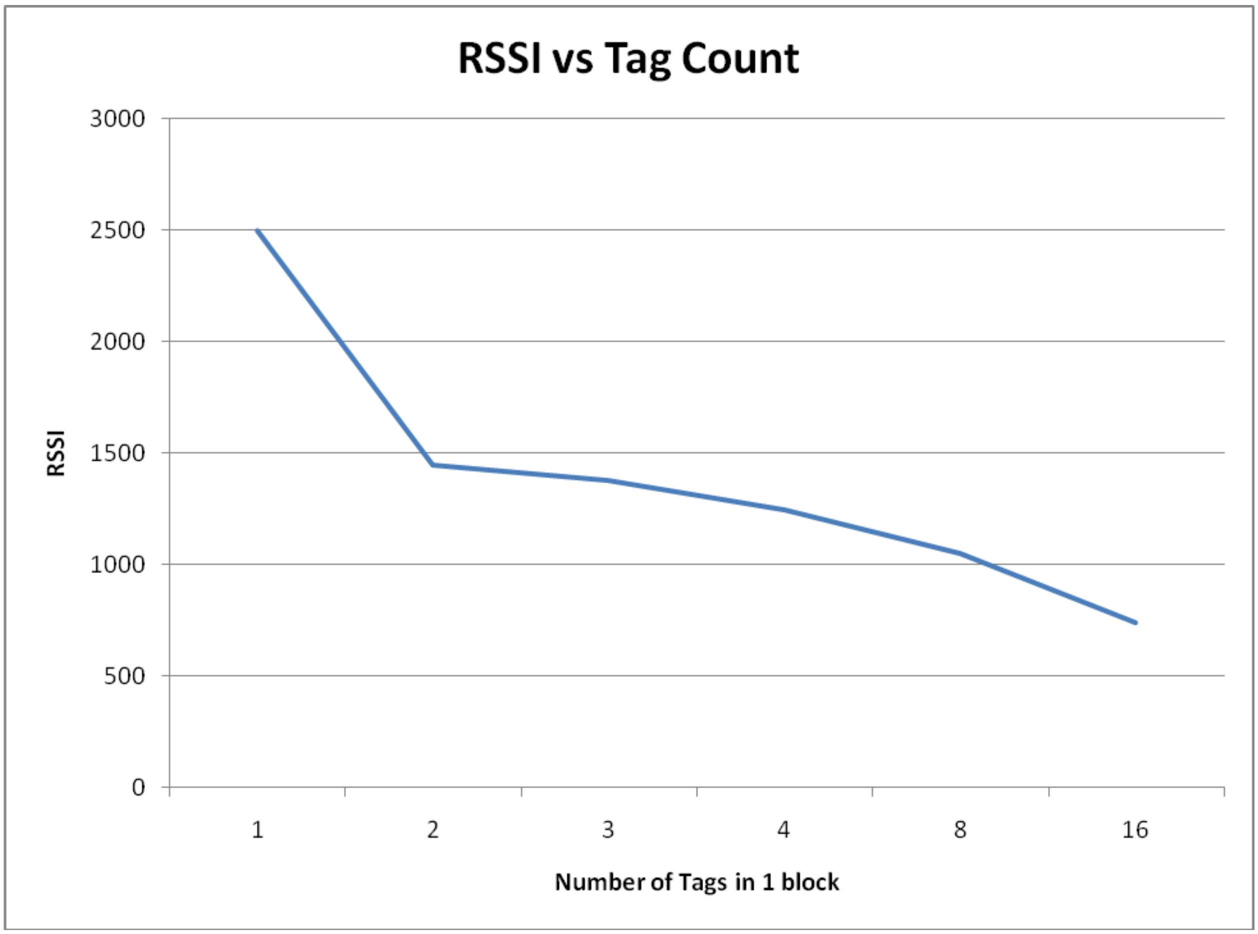


Figure 34 RSSI vs. Tag Count

In a test 16 tags in 1 block, with the help of the read count about 20% of the tags were corrected and overall about 60% of the tags were accurately identified.25% of the tags were identified to be in the adjacent block and about 15% of the tags were identified to be within 2 blocks of the actual location.

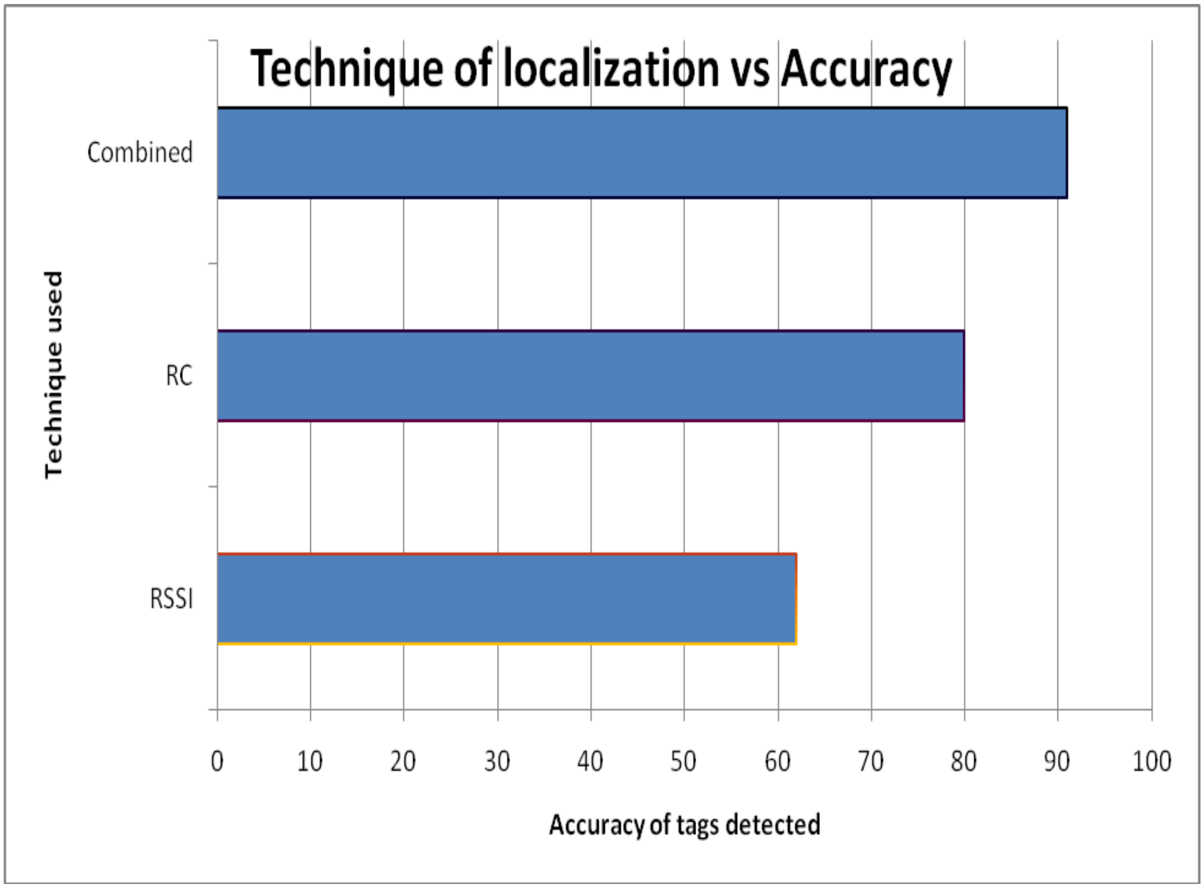
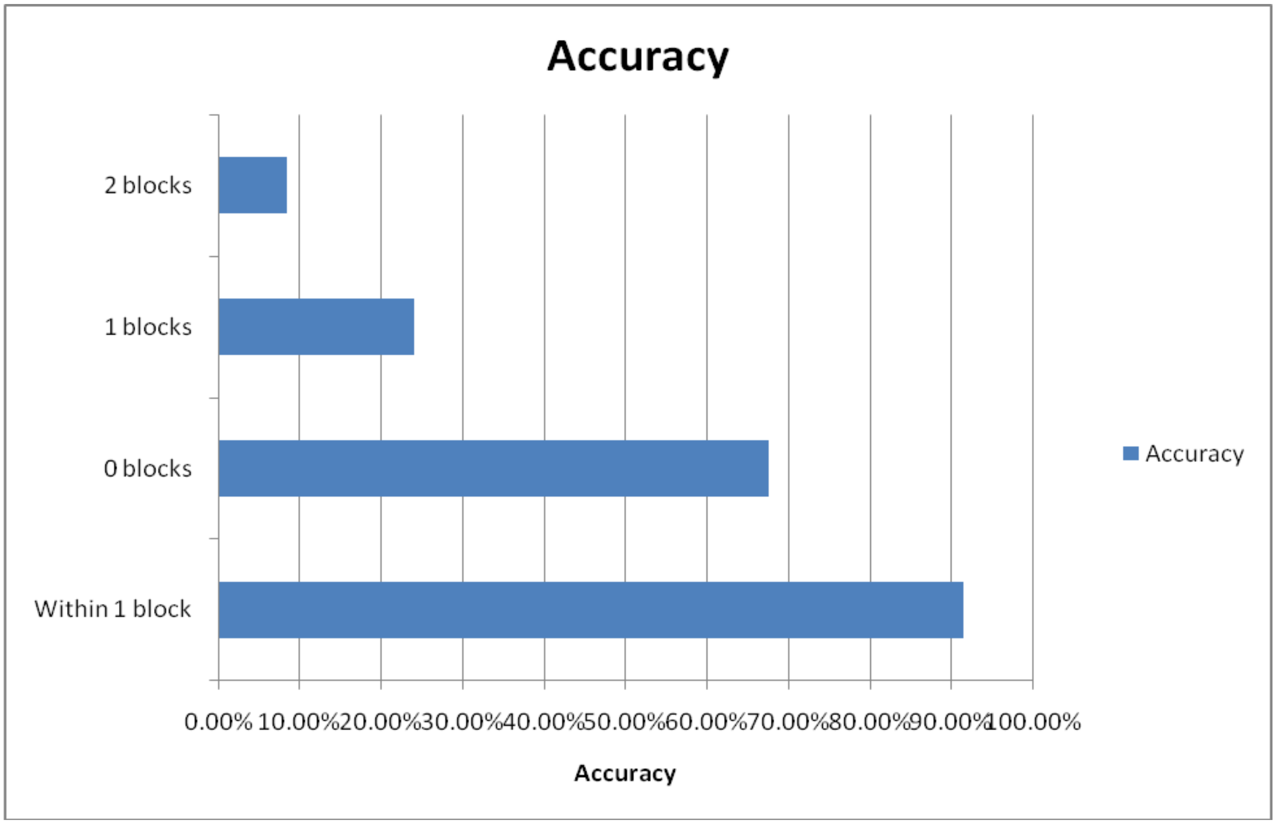


Figure 35 Accuracy of various techniques used in the system for localization



As we can see from the graphs in a dense environment the usage of the combination of the two techniques is able to achieve more than 90% accuracy which is consistent with the numbers reported in various approaches.

7. Future Work

We found that the third antenna used in the system can be used for more accurate predictions than just identifying if the tag is present in the given area or not. The pattern of reads from the elevated antenna was not easily "predictable" using our current model and needs a better estimation model for accurate predictions.

The Read-Count implementation needs to be modified from a lookup table to a more complex estimation model.

We also found that an alternate model of estimation of tags, again treating them as a classification problem, using a Bayesian model was able to provide interesting results.

8. References

1. Herbert Jaeger A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach http://www.faculty.iu-bremen.de/hjaeger/pubs/ESNTutorial.pdf
2. Herbert Jaeger, The "echo state" approach to analyzing and training recurrent neural networks.
3. Karthik Narayanan Moncombu Ramakrishnan, Performance Benchmarks for Passive UHF RFID Tags
4. David Mazza, Practical Localization of Passive RFID Tags
5. Jian Zhang Ivan Marsic, Link Quality and Signal-to-Noise Ratio in 802.11 WLAN with Fading: A Time-Series Analysis
6. Alien 9900 RFID Reader Overview, http://www.alientechnology.com/docs/products/DS_ALR_9900.pdf
7. E. Inc. EPCTM generation 1 tag data standards, version 1.1 Revision 1.27, technical report, EPCglobal Inc., 2005.
8. Matlab from Mathworks, http://www.mathworks.com/matlabcentral/
9. Java DataBase Connectivity Api, http://www.ch-werner.de/javasqlite/
10. Pavel Nikitin, K.V. Seshagiri Rao, Sander Lam, Vijay Pillai, Rene Martinez, and Harley Heinrich, Power reflection coefficient analysis for complex impedances in RFID tag design, September 2005.
11. N. Eberhardt, Towards RFID Performance benchmark tests, Technical report, Auto-ID Center, Massachusetts Institute of Technology, 2002
12. MIT Auto-ID Center, Draft protocol specification for a 900 MHz Class 0 Radio Frequency Identification Tag, 23 Feb 2003
13. Auto-ID center, 860MHz–960MHz Class I Radio Frequency Identification Tag Radio Frequency& Logical Communication Interface Specification Recommended Standard, Version 1.0.0, 14 Nov 2004
14. Alien Technology, EPCglobal Class 1 Gen 2 RFID Specification, White paper, 2005.
15. Active RFID Information, Gao RFID, http://www.gaorfid.com/index.php?main_page=index&cPath=130
16. RFID-based Indoor Antenna Localization System using Passive Tag and Variable RF-Attenuation, Ren. C. Luo et al, IEEE 2007
17. Adaptive Likelihood Distribution Algorithm for the Localization of Passive RFID Tags, Yuuki Ota et al, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2008
18. An Exposure System for Evaluating Possible Effects of RFID on Various Formulations of Drug Product, H Bassen et al, IEEE 2007
19. Utilizing RFID Signaling Scheme for Localization of Stationary Objects and Speed Estimation of Mobile Objects, Paul Wilson et al, 2006

20. Toward Hazard Aware Spaces: Localization using Passive RFID Technology, Peter Bajcsy et al, NCSA,2006

Appendix 1: Sample configuration xml

```xml
<reader>
  <ip>192.168.1.100</ip>
  <runningtime>100</runningtime>
  <antennaseq>0,1,2</antennaseq>
  <outputfile>rc9.csv</outputfile>
  <gui>1</gui>
  <antenna>
    <source>1</source>
    <coords x="0" dx="1" ></coords>
  </antenna>
  <antenna>
    <source>2</source>
    <coords y="0" dy="1" ></coords>
  </antenna>
  <antenna>
    <source>0</source>
    <coords startx="0" endx="4" starty="0" endy="4"
vertical="1"></coords>
  </antenna>
</reader>
<reader>
  <ip>192.168.1.102</ip>
  <runningtime>100</runningtime>
  <antennaseq>0,1,2</antennaseq>
  <outputfile>rc9.csv</outputfile>
  <gui>1</gui>
  <antenna>
    <source>1</source>
    <coords x="0" dx="1" ></coords>
  </antenna>
  <antenna>
    <source>2</source>
    <coords y="0" dy="1" ></coords>
  </antenna>
  <antenna>
    <source>0</source>
    <coords startx="0" endx="4" starty="0" endy="4"
vertical="1"></coords>
  </antenna>
</reader>
```