Passive Indoor Localisation Using Radio Tomographic Imaging and Clustering Techniques to Improve Location Accuracy

A thesis submitted in partial fulfilment of the requirements
for the award of the degree

Bachelor of Engineering (Electrical)

From

University of Wollongong

by

John Martins

School of Electrical, Computer and Telecommunications
Engineering

October, 2015

Supervisor: Dr Montserrat Ros

Abstract

Locating individuals accurately within indoor environments can have many beneficial applications such as safety, security, and automation. The aim of this annual thesis is to help contribute to the wireless tracking industry by bringing forth another localisation technique for reconstructing an objects location image with a relatively high accuracy.

A review of current literature showed that a low-cost, low-power system is suitable for experimental validation of this thesis. By mounting XBee Radio Frequency (RF) modules on the Seeeduino Stalker wireless sensor node, determining an objects location is possible by measuring the difference of Received Signal Strength (RSS) decibel values. This method is known as Radio Tomographic Imaging (RTI).

RTI is the method chosen for imaging an objects location within a tracking environment by utilising the effect of signal attenuation. This localisation technique is defined by researchers as an ill-posed problem, as there are more pixels in a localisation image than RSS measurements taken. Noise is another major contributing factor to this ill-posed problem, as noise is able to cause large variations towards RSS measurements, which can render the results useless.

This thesis aims to contribute another image reconstruction technique, clustering, to the field of wireless localisation. Clustering techniques such as K-Means and C-Means are able to determine the centroid of a cluster of data points. For RTI, these data points are the intersection of attenuated links in the wireless network. K-Means and C-Means are also compared to current techniques developed for image reconstruction.

Experiments conducted confirmed that, by comparing the attenuated RSS values with a RSS calibration matrix, an object was able to be detected within an RTI system with a very small location error. K-means and C-means clustering were also shown, through RTI experiments that they are capable of determining an objects location within a tracking environment with a relatively high accuracy compared to simple localisation techniques.

Acknowledgments

I would like to thank my thesis supervisor Dr Montserrat Ros for her ongoing support and guidance throughout the duration of my thesis. I would also like to thank my fiancée Tegan Murray for putting up with me during the stressful times of my research project.

Statement of Originality

I, John Martins, declare that this thesis, submitted as part of the requirements for the award of Bachelor of Engineering, in the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications or assessment at any other academic institution.

Signature:

Print Name:

Student ID Number:

Date:

Table of Contents

List of Figures

List of Tables

Abbreviations and Symbols

RTI	Radio Tomographic Imaging
RSS	Received Signal Strength, dB
WSN	Wireless Sensor Network
RF	Radio Frequency
TOA	Time of Arrival
AOA	Angle of Arrival
dB	Decibels
ACK	Acknowledgment

List of Changes

Section	Statement of Changes	Page Number
2.1.3	Reworded the section to incorporate a referral to the image.	ii

Chapter 1. Introduction

This thesis researches into current techniques and algorithms that have been implemented within various wireless systems to determine an object/persons position inside an indoor environment. A Wireless Sensor Network (WSN) capable of determining indoor positon is constructed over the duration of this thesis to determine whether certain clustering algorithms, such as K-means and C-means, can be implemented to increase location accuracy.

To be able to locate an object within a WSN, the wireless sensor nodes surrounding the network transmit and receive Radio Frequency (RF) signals and measure the change in the Received Signal Strength (RSS) to determine if a link has been attenuated by an object or not. This technique is known as Radio Tomographic Imaging (RTI) and is one of the main focuses of this thesis report. By being able to determine if a link has been attenuated, a general location of an object, with a slightly large error, can be determined. The clustering algorithms can then be implemented to determine if they are able to help minimize this error.

This thesis is an extension of Stephen Hart's thesis "Comparison of Clustering Techniques for RTI positioning using Graph Theory"[1]. This thesis continues from Stephen Hart's research with plans of replicating his simulated RTI experiments on a hardware platform to justify if clustering techniques are suitable for indoor positioning/tracking. Stephen tested his simulated experiments on a single, double and triple object interference within the simulated RTI system. As this thesis only focuses on a single interference within the system, the clustering localisation results gathered will be compared to only Stephens single interference test.

1.1 Motivation

Each year, many emergency responders are injured because they lack the ability to locate individuals during house fires, hostage situations, or other emergencies [2]. If the location of people in these tragic situations is known prior to entering the dangerous environments, emergency responders will know in which area to focus all of their attention. This could in turn reduce the number of injuries/deaths per year.

Another situation where the knowledge of a person's location could be beneficial to the person being tracked is in health care. Knowing the location of a health care patient can lead to many purposes such as monitoring daily activities, alerting caretakers of abnormal behaviour, or even automating the lights to reduce electricity consumption [3].

RTI provides a solution to all of these situations. RTI does not require the person being tracked to wear any obtrusive electronic devices to be able to monitor their position, which is beneficial in emergency situations. Other position estimation systems such as infrared or optical are able to locate an individual, yet RTI is able to penetrate walls, smoke, and debris as it uses RF signals to locate the person. RTI is also able to operate in dark environments, where optical system such as video cameras cannot.

1.2 Thesis Objectives

The purpose of this annual thesis is to utilise the passive localization technique, RTI, and clustering techniques, such as K-Means and C-Means, to determine an object/persons location within a tracking environment with a high accuracy. One of the main objectives in ECTE458 is to review current literature regarding indoor localisation system. Literature must be reviewed and compared to determine which localisation technique is preferred, and how clustering techniques compare and contrast to techniques already developed. Another objective for this thesis is to design and implement an RTI system which is capable of locating an object within an indoor environment. During the first semester of this thesis, the main goal was to be able to gather the data points of intersecting attenuated link, which is needed for the clustering techniques. As this goal was reached, ECTE458 continues on this research by implementing K-Means and C-Means clustering techniques to RTI data and determining which clustering method or simple localisation method is able to produce a location image with the highest accuracy.

To develop an RTI system with clustering algorithms, several design questions must be researched and reviewed such as:

1. What hardware will the wireless sensor nodes, which surround the tracking environment, contain?

2. What protocol will the nodes follow for the transmission of wireless signals to each other?
3. How will the nodes be able to determine the RSS of the transmitted RF signals?
4. How will the RSS values be transmitted to a remote computer?
5. What software will the remote computer be running?
6. How will the software use the incoming data to construct an image location of the object?
7. How will the clustering algorithms be implemented within the received data?
8. What factor can determine which clustering algorithm implemented provides the highest accuracy?

1.3 Overview of Thesis

This chapter presented an overview of the scope of this project and the motivation behind conducting it. It also specified certain objectives which are to be met for the fulfilment of this thesis.

The following chapter, chapter 2, reviews current literature regarding the localisation of an object/individual within a certain environment. Chapter 2 discusses detrimental effect which can occur within WSN's, as well as compares and contrasts different techniques and algorithms which can be used for localisation. Stephen Harts 'Comparison of Clustering Techniques for RTI Positioning Using Graph Theory'[1] thesis will also be discussed, as this reports thesis is an extension of his.

The system design and implementation of an RTI system is discussed in chapter 3 of this thesis. This chapter outlines the hardware and software which is utilised to create an RTI system, and a method of how to determine and graph an objects position. An approach of how to implement clustering algorithms on an 8 node and a 28 node RTI system is also discussed.

Chapter 4 discusses the experimental results obtained from the experiments conducted. The first section outlines if RTI is a suitable method for determining a general location of the objects position, and whether these results can be implemented within clustering algorithms to determine an object location accurately. The second section examines how the K-means, C-means and Centroid localisation algorithms compare to each other within a larger 28 node RTI system. Chapter 4 also discusses and compares the results to the theory described in the researched literature.

The final chapter, chapter 5, summarises the entire thesis report, and discusses the key aspects discovered in the experiments conducted. Further steps are also outlined to help further the research into clustering algorithms within RTI systems.

Chapter 2. Literature Review

This chapter reviews all the major studies that have been associated with locating an object within a constrained environment. It outlines the positive and negative effects of wireless communication such as shadowing, free-space path loss, and multipath. The second section contrasts and compares various distance estimation techniques used to locate or track individuals such as received signal strength (RSS), time of arrival and angle of arrival. Section 3 details information about RTI, ultra-wideband and multiple-input-multiple-output, which are all passive localisation techniques. This section also describes applications where these techniques are utilised, and their benefits over other techniques. The fourth and final section discusses various clustering techniques and their differences with one another. It also outlines how it can contribute to wireless location imaging by providing another method to image the attenuation of links in a WSN.

2.1 Wireless Communications

Wireless communication allows for electromagnetic signals to be transmitted between two or more nodes with no wired connection. The use of wireless communications has led to successful applications such as cellular phones, WLAN's, radio, satellite networks etc. A severe challenge with wireless communication is reliable high-speed communication, due to effects such as shadowing, path loss and multipath, which can affect the electromagnetic signals [4].

2.1.1 Shadowing

Between a sender and transmitter in wireless communication systems are various objects and obstacles, which cause the propagating electromagnetic signal to lose signal strength. This effect is called shadowing. Bao Hua, L [5] shows on three different wireless systems that shadowing can have a significant impact on the network performance. In some wireless communication systems, the shadowing effect can be beneficial, as the loss in signal strength shows that between the sender and receiver is an object, which is attenuating the wireless signal. Utilising this information, and deploying a larger mesh-connected wireless system, can allow moving obstacles to be tracked within an environment [2, 3]

2.1.2 Free-Space Path Loss

Shadowing is the effect which obstacles have on the electromagnetic signal, whereas free-space path loss is the effect that the distance and the medium that the signal travels through, has on the propagating signal. From Equation 2.1 it is shown that as the distance and frequency increases between two nodes in a wireless network, so does the free-space path loss. To determine the accuracy of Equation 2.1, an experiment was performed by Katircioglu, O., which confirmed that free-space path loss affects the received signal strength [6]

$$\boldsymbol{P_r = P_t - 20 * \log\left(\frac{4\pi f d}{c}\right)} \quad [dBm] \qquad \textit{Equation 2.1}$$

Where: $\mathrm{P_r}$ = Recieved signal strength [dBm]

$\mathrm{P_t}$ = Transmitter signal strength [dBm]

f = Frequency [Hz]

c = Speed of light [m/s]

d = Distance from transmitter [m]

2.1.3 Multipath

When an electromagnetic signal is travelling from a sender node to a receiver node, the signal spreads out and may be reflected off objects surrounding the wireless network. These reflected signals result in the same transmitted signal being collected by the receiver node several times, with also a time difference due to the longer paths being taken [7].This is known as Multipath distortion, and can be seen in *figure 2.1*.

Multipath distortion can in turn affect the signal strength of the transmitted signal due to constructive and deconstructive interference. When incoming signals have the same phase, the result is constructive interference, which increases the combined signals amplitude. Whereas if the phases are opposite, deconstructive interference occurs, where the amplitude of the combined signal decreases [3] . To minimize the effects of multipath distortions, some systems implement directional antennae's, which radiate the electromagnetic signal in only one direction [7].

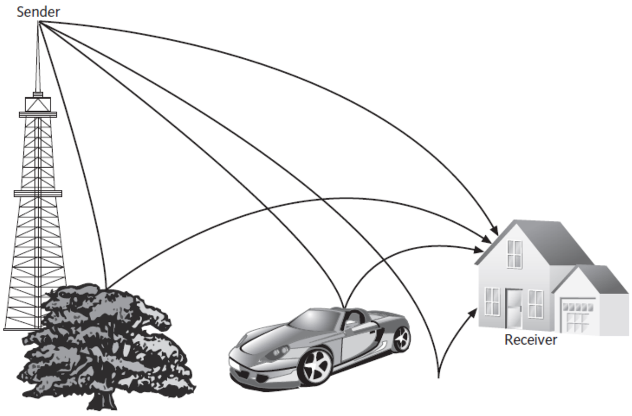


Figure 2.1: Multipath Interference [7]

2.2 Distance Estimation Techniques

"One of the most challenging topics when dealing with WSN's is the localisation and tracking of objects from measurements collected by the nodes themselves" [8]. There are two types of wireless tracking systems available: active and passive. Active systems require the object which is being tracked to wear an electronic device/tag, whereas passive does not require any physical device to be attached to the object [9]. Both active and passive systems require the object to be within a tracking environment, which consists of wireless sensor nodes surrounding the area. These nodes use various features of the transmitted signals such as Received Signal Strength (RSS), Time-of-Arrival (TOA), or Angle-of-Arrival (AOA) depending on the system implemented for tracking purposes.

2.2.1 Received Signal Strength (RSS)

Received Signal Strength utilises the effect of shadowing in WSN's to be able to determine an object within an environment. Compared to other forms of measurements, RSS is relatively inexpensive and simple to implement into hardware, requiring no additional bandwidth or energy requirements [10]. Even though RSS utilises the effect of shadowing; multipath and free-space path loss still has a detrimental impact on the RSS value. Joey Wilson and Neal Patwari [2] defined a mathematical formula for received signal strength as shown in *Equation 2.2* which shows what factors can affect the RSS measurements. They also go to show that just by using the RSS values of a network of nodes surrounding a $40m^2$ area, they are able to locate and track a person wearing no active tags or electronic devices. Neal Patwari and M. Bocca managed to achieve a low error of 0.23m from using RSS values to determine human positioning [3].

$$\boldsymbol{y_i(t) = P_i - L_i - S_i(t) - F_i(t) - v_i(t)} \qquad \textit{Equation } 2.2$$

Where: $y_i(t)$ = Signal Strength of a link (dB)

P_i = Transmitted Power (dB)

L_i = Static losses due to distance, antenna patters etc. (dB)

$S_i(t)$ = Shadowing losses due to attenuation through objects (dB)

$F_i(t)$ = Fading loss that occurs from multipath distortion (dB)

$v_i(t)$ = Measurement noise (dB)

2.2.2 Time-of-Arrival (TOA)

Time-of-Arrival determines the distance between a mobile node and a stationary node by measuring the time at which the wireless signal arrives at the receiver node. The Time-of-Arrival is calculated by the time of transmission plus a propagation-induced time delay. The TOA system is similar to an RSS system; the only difference is the RSS system uses the received strength to determine attenuation due to an object, whereas TOA uses the propagation delay due to the object. As Neal Patwari mentioned in [10], major sources of error in TOA systems is additive noise, and multipath. TOA systems are mainly used within active localisation systems, where the object being tracked wears a tag, which transmits signals to the surrounding sensor nodes.

2.2.3 Angle-of-Arrival (AOA)

TOA provides information corresponding to the distance between nodes, whereas Angle-of-Arrival gathers localisation data by measuring the direction the signal transmitted from. Neal Patwari mentions two common ways for sensors to measure AOA [10]. The most common method uses sensor arrays as nodes within the network, where the AOA is estimated from the difference in the arrival times between each sensor, this is shown in figure 2.2. The second method utilises the RSS ratio between two or more antennas located on a single node to determine the AOA.

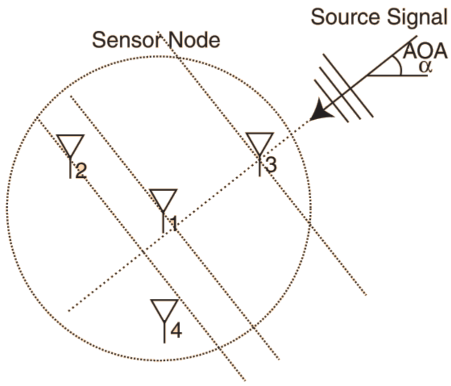


Figure 2.2: AOA being estimated from 4 sensors on a single node [10]

Having multiple sensors on the one node increases the cost and system complexity, whereas other distance estimation techniques such as RSS and TOA do not require this additional hardware meaning they are more cost effective to implement in larger networks [9].

An AOA commercial location estimation system called Ubisense has a very high precision, having an error of approximately 0.15m [9]. Comparing it to the error of 0.23m achieved by N. Patwari and M. Bocca, the AOA system provides a more accurate position estimation compared to the RSS-based system. Yet, the advantage of the RSS system is that the body being tracked does not have to wear any active tags to determine their position.

2.3 Passive Localisation Techniques

Compared to active localisation systems, passive systems have shown a growing interest in recent years as they do not require additional infrastructures, and do not require the object being tracked to wear any electronic equipment.

2.3.1 Radio Tomographic Imaging (RTI)

RTI is a method which locates an object within a WSN by measuring signal attenuation, then applying an image reconstruction technique to produce the objects location [2]. RTI has many different applications, from rescue operations, as RTI is able to penetrate smoke and walls, too “smart” buildings, where knowing an individual’s locations can trigger certain events such as lighting, air conditioning and heating [2]. RTI has the advantage over other systems such as optical or infrared as RTI is able to operate in the dark and also travel through various obstructions.

The wireless signals are produced by nodes, which surround the imaging environment, as seen in figure 2.3. Each node in the system broadcasts a signal to every other node, these receiving nodes then compute the RSS of the link then send the value to a computer for image processing.

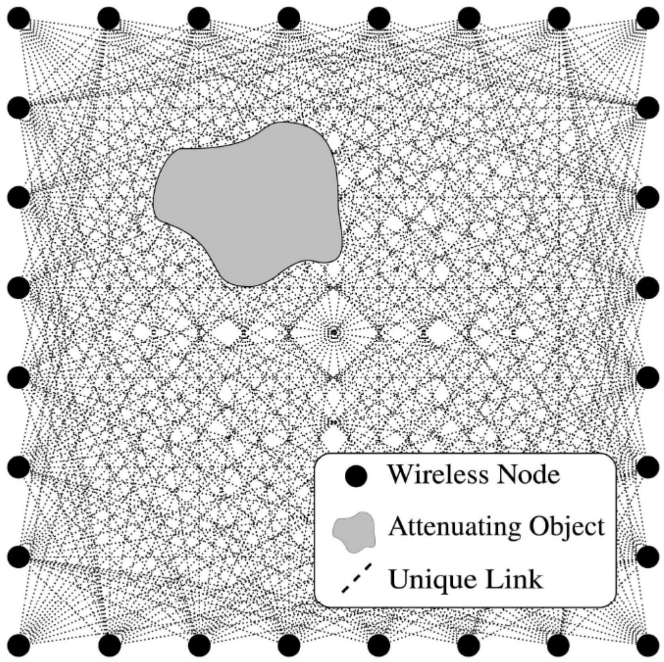


Figure 2.3: RTI network [2]

Each unique link in the system has to be assessed to determine whether an object has obstructed that link. The more nodes in the RTI system results in a substantially larger amount of links, which in turn will result in an object obstructing more links giving a higher location accuracy. The amount of links within any system is shown in equation 2.3 *[2]*.

$$\boldsymbol{M = \frac{K^2 - K}{2}} \qquad \textit{Equation 2.3}$$

Where: M = Amount of links

K = Amount of nodes

RTI models the attenuation occurring within an image vector of $\mathfrak{R}^N$, where N is the voxels of a network region [2]. As the voxel locations within a system can be predetermined, we are able to determine where the attenuation for certain links are occurring.

Wilson and Patwari describe RTI as an ill-posed inverse problem, which means that small amounts of noise can affect the system which will render the measurements useless [2]. Different data analysis methods have been implemented such as Linear Back Projection (LBP), Tikhonov Regularization, and the Projected Landweber Iteration [11]. Each method uses the same model to begin with, which can be described in matrix form as: *[2, 3, 8, 11-14]*:

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{n} \qquad \textit{Equation 2.4}$$

Where: y = $[y_1, y_2, y_3, \dots, y_M]^T$ (Shadowing Losses Vector)

x = $[x_1, x_2, x_3, \dots, x_N]^T$ (Voxel Vector)

n = $[n_1, n_2, n_3, \dots, n_M]^T$(Noise Vector)

W =$[w_{ij}]_{MxN}$ (Weight Matrix)

LBP is currently the simplest method available to be able to reconstruct an image. It has a great implementation rate within RTI systems as it also requires very little hardware to operate, and can be easily implemented on wireless sensor nodes. The drawback of LBP compared to Tikhonov Regularization and Projected Landweber Iteration is that it doesn't produce an extremely clear and accurate final location image. [11]

Using Tikhonov Regularization for image reconstruction results in a much higher image accuracy as Tikhonov also takes into account the noise in the testing environment, which it aims to supress. Another major advantage of Tikhonov Regularisation is that most of the computation required to produce an accurate image can be pre-calculated, and then applied to the incoming data resulting in a fast and efficient RTI system [14]. Even though Tikhonov Regularisation produces a higher image accuracy compared to other methods, it may not be favoured as this regularisation technique requires large amount of computation due to long calculations of large matrices. [11, 14]

The Projected Landweber Iteration differs from the two previous image reconstruction techniques as it is mainly used where time constraints aren't needed. Tikhonov and LBP are ideal for real time systems where the image must be displayed in a few seconds to be able to track a body, whereas Projected Landweber is utilised when the need for tracking an asset is not needed. This is the case as Projected Landweber Iteration, as the name suggests, requires a large amount of iterations to produce a high quality image. The main benefit of using Project Landweber Iteration is that it produces a higher quality image compared to other techniques, yet it takes longer to construct the location image.[11]

Liu Heng compares these 3 different image reconstruction techniques in an experiment using the same system to determine which method produces the most accurate results [11]. He found that LBP needed the least computation and provides acceptable results, Tikhonov provides more accurate results than LBP but needed a high amount of computation, and Projected Landweber Iteration resulted in the most accurate results but required the most amount of time to generate.

2.3.2 Ultra-wideband (UWB)

Where RTI measures signal attenuation in a WSN, Ultra-wideband monitors signal changes which are caused by an object/person within the tracking environment.[9]

UWB systems used a phased array of various radars, which emit UWB pulses. These emitted pulses are then reflected and scattered by objects, which are measured by receiving nodes to estimate a range and bearing of the object [2]. As UWB systems measure the multipath effects of wireless signals, they are extremely accurate when the object is close to the emitter. The further away the object is, the less accurate these systems are due to incorporated noise in signals, scattering losses, and the large bandwidths.

UWB can be beneficial for many scenarios where RTI cannot such as wall imaging and ground penetrating radar (GPR) [9]. Wall imaging and GPR involve locating objects within dense walls and the ground. UWB is able to conduct wall imaging and GPR as it measures the echoes that the object causes from the emitted signal, where RTI does not measure signal reflections.

The advantages of using UWB over RTI is the decreased interference noise has to the UWB system, and its susceptibility to multipath. For RTI, multipath adds detrimental features to the RSS values obtained by receiving nodes, yet UWB uses multipath to its benefit, as it measures angles of reflected signals to determine an accurate location. UWB is also able to use TOA and AOA in conjunction to determine a more accurate position estimation, whereas RTI can only utilise RSS measurements.[8]

2.3.3 Multiple-Input-Multiple-Output (MIMO)

Multiple-Input-Multiple-Output wireless network systems are similar to UWB systems where they use antennae's on each of the nodes to measure the range and angle of objects, which are scattering and reflecting signals, as seen in *figure 2.4*. Where MIMO differs is that each node in the system has multiple antennae's on each node. Having multiple antennae's on each node provides enhanced performance in the wireless network resulting in higher data rates without increasing the transmitted signals power or bandwidth [4].

Having an increase in antennae's on each node results in a higher cost compared to other localisation techniques such as RTI and UWB. The size of the node also increases, as well as the added complexity of computation required for multi-dimensional signal processing. MIMO differs from RTI in the same way RTI differs from UWB. Instead of measuring the change in RSS values between nodes caused by objects, MIMO measures reflections of transmitted signals caused by the object/person within the tracking environment.[2]

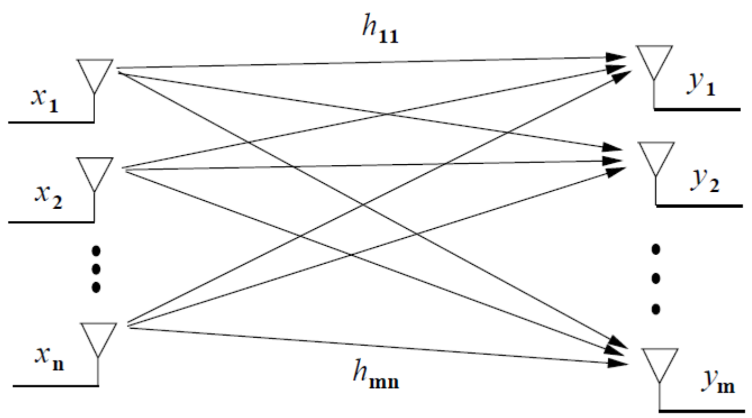


Figure 2.4: MIMO system

2.4 Clustering Techniques

Clustering is defined by Pavel Berkhin as "a division of data into groups of similar objects" [15]. Representing data within clusters may necessarily lose some fine detail points, but by using clustering methods we can achieve simplification. There are many clustering algorithms as there is no 'proper' way to cluster a set of data [16]. Clustering algorithms are separated into two different methods, the Hierarchical Method and the Partitioning Method. Two partitioning methods, K-Means and C-Means are also analysed and compared in this section.

2.4.1 Clustering Methods

Hierarchical Method and Partitioning Method are both methods used for clustering data. Hierarchical clustering generates a cluster hierarchy, also known as a tree of clusters named dendrograms [15]. There are two different ways that a hierarchical clustering method can be categorized, either an agglomerative (bottom-up) or divisive (top-down). Agglomerative starts with one point clusters, then continues to merge more one point clusters, whereas divisive clustering starts with one cluster of all the data points, then gradually splits them into smaller clusters.

Partitioning Methods move data points from one cluster to another, beginning from an initial partitioning. This method requires the amount of clusters generated to be of a known value prior to implementation.

The most beneficial clustering method to be used in conjunction with RTI is Partitioning Methods. This method is chosen as Partitioning Methods provide a means of dealing with data points which lie outside of clusters, and it also provides a way to determine the centroid of clusters, which is needed to locate an object/individual [16]. The data points, which will be used for clustering in this thesis, are the intersections of broken links in the RTI network.

2.4.2 K-Means Clustering

K-Means Clustering, also referred to as hard K-Means is the most commonly used and simplest clustering algorithm to be implemented. The K-Means clustering algorithm partitions all the data points into a user defined number of clusters [16]. For RTI the number of clusters that need to be generated will be the number of objects/people which need to be tracked within the wireless sensor network.

The K-Means clustering algorithm starts with an initial set of cluster centres, which can be chosen at random, or according to some procedure. In each iteration of the algorithm, another data point is assigned to a cluster according to its Euclidean distance, then the cluster centroid is recalculated. To determine the centroid of each cluster a formula is used as shown in equation 2.5.

$$\boldsymbol{\mu_k = \frac{1}{N_k}\sum_{q=1}^{N_k} x_q} \qquad \textit{Equation 2.5}$$

Where: N_k = Number of instances belonging to cluster k

μ_k = Mean of cluster k (Centroid)

Figure 2.5 shows an example of K-means clustering, where the data set is chosen to be separated into 3 clusters. K-means is also able to calculate the midpoint of all the clusters.

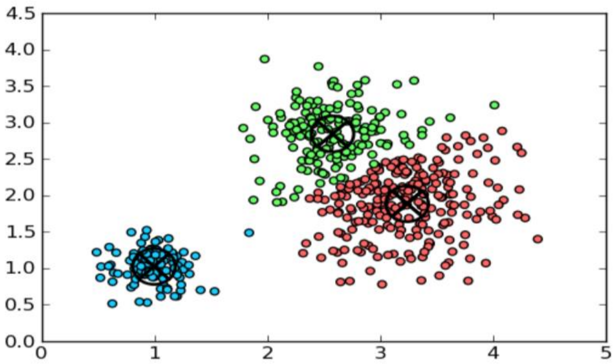


Figure 2.5: K-Means Clustering showing centroids of 3 clusters [17]

2.4.3 C-Means Clustering

C-Means clustering, also known as Fuzzy C-Means clustering, provides a fuzzy solution to clustering data and finding the centroid of clusters. This clustering technique is shown to provide more accurate results compared to K-means clustering for a simulated RTI network [1]. As C-Means is a fuzzy clustering technique, undertaking the same experiment twice can result in different end results. The only drawback to implementing C-Means over K-Means is the computation time required to process the data.

2.4.4 Stephen Hart Thesis

This thesis is an extension of the thesis ‘Comparison of Clustering Techniques for RTI Positioning Using Graph Theory’ by Stephen Hart [1]. Stephen compared clustering techniques such as K-Means and C-Means, to determine if it is possible to deploy clustering techniques in an RTI system to accurately determine a person’s location.

MATLAB simulations were undertaken by Stephen to implement each clustering technique on different sets of data to compare the clustering techniques and conclude on which achieves the greatest accuracy for an RTI network. The MATLAB simulations showed that “C-Means clustering demonstrated a markedly enhanced capability at estimating centroid locations compared to K-Means.”[1]. A position location with an error of 4cm was produced using the simulated data in MATLAB. Unfortunately, due to time constraints, Stephen wasn’t able to deploy the clustering techniques on hardware, which is where this thesis takes over.

Stephen implemented K-means and C-means clustering algorithms on a single, double, and triple target interference. Table 2.1 shows the average error with both clustering techniques against the different amounts of object obstructions.

		K-Means	C-Means
Single Interference	Min Error (m)	0.0100	0.0000
	Max Error (m)	0.0818	0.1179
	Average Error (m)	0.0371	0.0404
Double Interference	Min Error (m)	0.0087	0.0016
	Max Error (m)	0.7524	0.2804
	Average Error (m)	0.1266	0.1006
Triple Interference	Min Error (m)	0.0007	0.0007
	Max Error (m)	0.9066	0.4371
	Average Error (m)	0.1662	0.1109

Table 2.1: Min, Max and Average values of K-means and C-means for single, double and triple interferences [18]

As seen from the results obtained through Stephens's simulated experiments, the position of each clustering methods location was extremely accurate. These accurate results were generated due to the fact that this thesis was simulated, with no interference at all. This reports thesis will construct a real RTI system using clustering algorithms to determine if these results are able to be generated on a physical system which can be affected by path loss, multipath, and interference.

As this reports thesis is only looking at single interference results for the clustering techniques, the final results gathered will be compared to Stephens single interference experiment which concluded that for a single interference within an RTI system, K-means clustering proved to be more accurate than C-means with an average error of 3.71cm compared to 4.04cm.

Chapter 3. System Design and Implementation

The purpose of this section is to outline the hardware and software utilised to create an 8 and 28 node wireless sensor networks for RTI, and to implement K-means and C-means clustering techniques on the data obtained through RTI. The first section shows how the wireless sensor networks make use of sensor nodes and RF modules to produce a reading of the RSS between nodes. Part two details how the computer software 'MATLAB', can process the RSS values of all links created in the system to determine which link is being attenuated by an object, thereby locating the objects position. The final section shows how K-means and C-means clustering can be implemented to the data sets obtained, and how the output can be compared to simple techniques, such as the centroid of the data set.

3.1 Wireless Sensor Network

Two wireless sensor networks, 8 nodes and 28 nodes, are constructed for the purpose of this thesis. An 8 node network surrounding a 0.8mx0.8m environment is constructed to ensure that by gathering RSS measurements of each node, an objects general location can be found with a relatively small error. The 8 node system will also implement clustering techniques to prove that a position, even if it has a low accuracy, is able to be determined using these methods.

The second system contains 28 nodes surrounding an environment of 2.7mx3.9m. This system is used to compare the three localisation methods, K-means, C-means, and centroid, to determine which can provide the highest accuracy within a large room environment. The 28 node system was designed by Daniel George [18], whom allowed me to use the data collected from the experiment to aid my thesis. Figure 3.1 shows the layout of the 8 node system whereas figure 3.2 shows the layout of the 28 node systems.

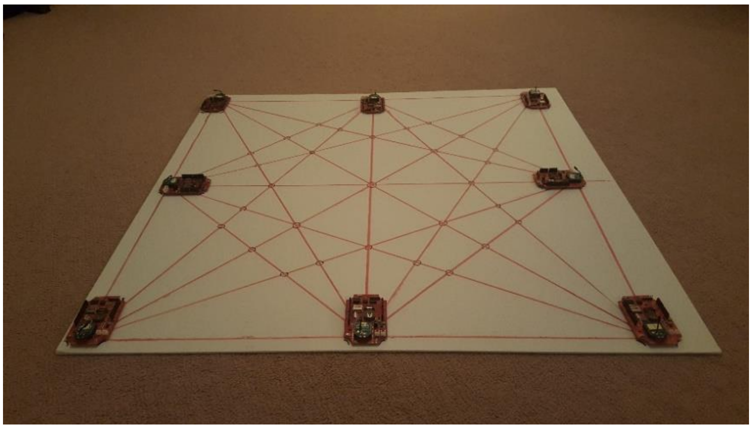


Figure 3.1: 8 node wireless sensor networks

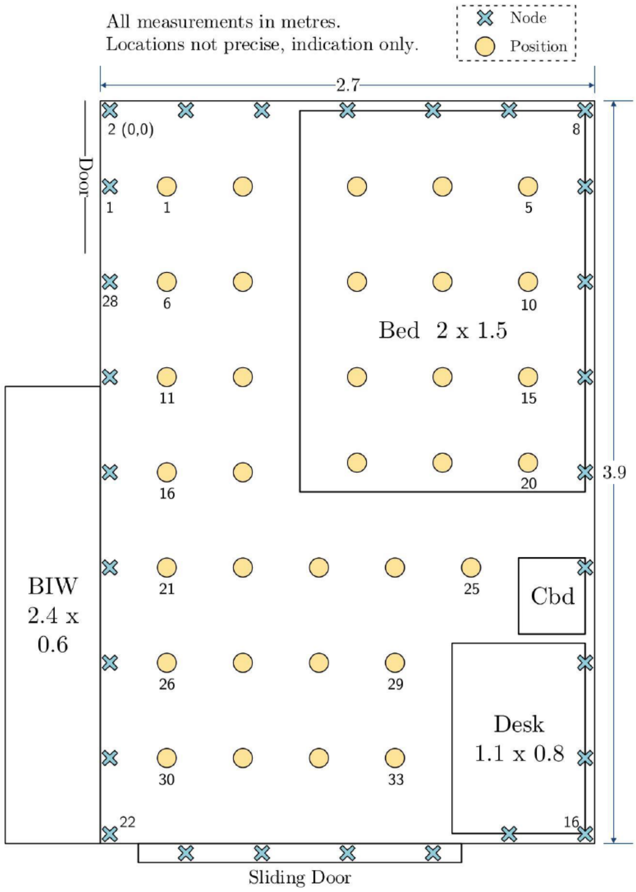


Figure 3.2: 28 node wireless sensor networks [18]

Each individual node within either system consists of an XBee S1 RF module, which meets the IEEE 802.15.4 standards [19], mounted on a Wireless Sensor Network node, the Seeeduino Stalker v2.3 [20]. Each node in the network is able to communicate to each other using RF signals. This allows for information such as RSS measurements and payloads to be shared around all the network nodes.

The Wireless Sensor Networks share the RSS measurements of each individual link amongst all nodes. These stored RSS values are then repeated in a signal transmission, which is received by a node connected via serial to a computer.

3.1.1 RF Communication

The XBee S1 RF is a low cost and low power module which is engineered to meet the IEEE 802.15.4 standard for use in wireless sensor networks [19]. These modules have an indoor data range of approximately 30m, which makes them perfect for indoor localisation in household environments. To configure the XBee for use in the wireless sensor network, the program ‘XCTU’ is utilised to change a number of parameters for each individual XBee. The key parameters to be modified for the XBee modules are the channel, PAN ID, destination address, MY address, baud rate, and API enable.

For all XBee modules to be able to communicate with each other, they all must be operating in the same channel with the same personal area network ID. The 802.15.4 protocol defines 16 different channels for the XBee, which operate in the 2.4GHz frequency band. A scan can be deployed from an XBee using the XCTU software to determine a suitable channel with the least amount of interference. The PAN ID ranges from 0x0000 to 0xFFFF and can be chosen at random, yet each XBee must have the same PAN ID to be able to communicate with one another

The destination address of an XBee is the address of the module that an XBee wants to transmit to. For the WSN, since a single XBee must be able to transmit to all other modules in the network, the destination address can be set to 0xFFFF, which is broadcast mode. MY address is a 16 bit unique identifier for each XBee within the network, which is chosen to be zero to eight. Node zero is connected to the computer via serial and nodes one to eight are placed around the tracking environment, starting in the top left corner, and moving clockwise. For the experiments conducted on the wireless sensor networks, the baud rate for each XBee is set to 57600 baud for data transfer, as the next baud rate, 115200 baud, was shown to produce random errors within the preliminary testing phase.

The XBee has 2 operating modes, API or AT mode. In AT mode, an XBee transmits its data to each receiving XBee, whereas in API mode, the data packets not only include the data, but also the ID of the sending XBee, size of the data, and the RSS value of the link.

API mode is chosen for the network as the sender ID and RSS values are key values needed for RTI.

	Channel	PAN ID	Destination Address	MY Address	Baudrate	API Enable
XBee Module	C	666	0xFFFF	0-8 0-28	57600	2

Table 3.1: XBee parameters using XCTU software

3.1.2 RSS Data Management

The Seeeduino Stalker v2.3 is used as the wireless sensor nodes in the RTI network. The purpose of the Seeeduino Stalker in the wireless sensor network is to read values from the mounted XBee, process the values and store them in an array, and transmit the array back through the XBee. The code produced for the Seeeduino nodes is developed to be universal, where all nodes in the system, regardless of their position or number, have the same code uploaded to them. The code uploaded to all the Seeeduino stalkers in the system is shown in Appendix B.

Once all the nodes are powered on, each Seeeduino sends an AT command to its mounted XBee module, which allows the Seeeduino to obtain the address of the XBee. After each XBee address is stored in the corresponding Seeeduino, all nodes go into standby mode, where they wait for a certain starting code to be received. The system will only start when the code 0x66 is transmitted from node 0, which is connected to the computer.

Once the starting code has been received, each Seeeduino uses a token passing protocol, where node 1 will transmit data first, then node 2, then all the way around to the final node. A visual representation of the token passing protocol can be seen in Figure 3.3.

When a single node is transmitting, all other nodes are listening for API packets. When a receiving node accepts data from a sending node, the receiving node reads the RSS value of the link, and stores it in an array. This array is then transmitted as the payload of the data packets. After a single round of transmission has been completed, each node in the system will have the RSS values of every other node.

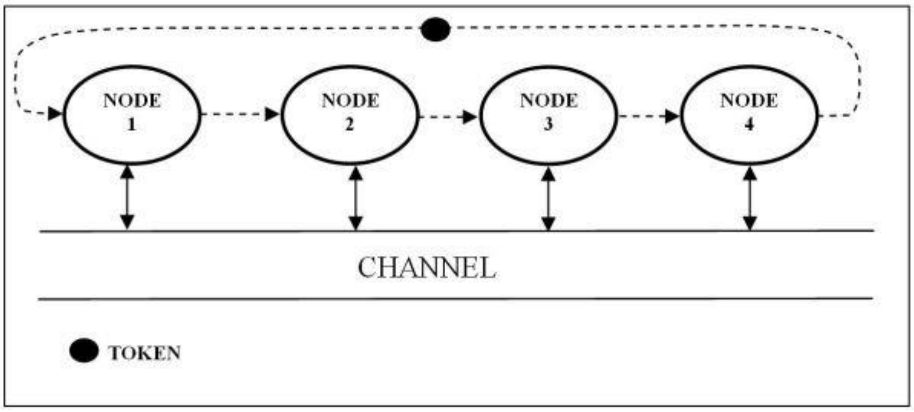


Figure 3.3: Token passing protocol used by the Seeeduino Stalkers[21]

3.2 Using RSS Measurements to Determine Location

Before reliable data can be read, the wireless sensor network must be initialised, as when they are powered on their array which hold RSS values of other nodes is empty. To start the system, a start signal is sent by MATLAB via the connected node. As all the nodes are operating in API mode, additional packets must be added to the transmitting data, and a checksum created so the receiving nodes don't disregard the transmitted frames. Table 3.2 shows the additional parameters that are added for API mode.

Parameters	Hex Values	Integer Values
Start Delimiter	0x7E	126
Address MSB	0x00	0
Address LSB	0x06	6
API identifier – 0x01 means transmitting 16 bit address	0x01	1
Frame ID	0x01	1
Destination Address MSB – 0xFF means broadcast mode	0xFF	255
Destination Address LSB – 0xFF means broadcast mode	0xFF	255
Option – 0x01 Disables ACK	0x01	1
RF Data	0x66	102
Checksum	0x98	152

Table 3.2: Parameters for transmitting in API mode

Once the API frame is transmitted from MATLAB, one round is executed by the wireless network. As this is only to initialize the wireless system, MATLAB doesn't read any values for the initial round.

For the 8 node system, when nodes one to eight are transmitting their payloads and reading RSS values for particular links, node zero is receiving all the transmission data and feeding it into MATLAB via serial. The transmission data received by MATLAB is not only the payload, which contains the RSS values of each link, but it also contains other frames that are introduced for API mode. API mode transmits data in packets of hex values, yet seeing as an integer value is needed for measurements and differences of RSS decibel values, the data received must be converted to integer form. An example of data received by node 0 after a single round of transmission is shown in figure 3.3.

RSSIdata <15x8 double>

	1	2	3	4	5	6	7	8
1	126	126	126	126	126	126	126	126
2	13	13	13	13	13	13	13	13
3	129	129	129	129	129	129	129	129
4	1	2	3	4	5	6	7	8
5	43	41	43	56	47	46	46	42
6	2	2	2	2	2	2	2	2
7	255	34	42	47	42	35	37	34
8	34	255	35	55	36	37	47	44
9	46	35	255	37	38	44	45	56
10	44	53	36	255	34	40	50	43
11	39	35	36	34	255	36	48	42
12	35	37	46	42	38	255	38	36
13	35	43	43	50	49	37	255	36
14	34	44	53	41	42	35	36	255
15	69	56	43	14	49	64	26	39

Figure 3.4: Example RSS values received in MATLAB by node 0

The parameters of interest in figure 3.3 are shown in rows 7-14, which hold the RSS values of links corresponding to certain nodes, and columns 1 - 8 which are the API packets received from nodes 1 – 8. These RSS values will then be used to first create a calibration matrix, then to determine if a links signal has been attenuated by an object.

When an object is within a wireless sensor network the object will attenuate the signals that pass through the object, this will result in a higher RSS value for those specific links. To be able to determine if any given link signal has been attenuated, the system must first be calibrated with initial RSS values. To create a calibration matrix the wireless network must complete a round of transmission with no object present in the system. This will give a calibration matrix with 'unbroken' links.

As the RTI network is now calibrated with RSS values when no object present, once an object enters the tracking area the RSS values of the links which pass through the object will increase as the object will attenuate the transmitted signal. To determine the location of the object the system completes another round of transmissions, and then compares the received RSS values with the calibrated RSS matrix. If the absolute value of the difference between the received RSS matrix and the calibrated RSS matrix is greater than a predefined threshold, the link will be considered 'broken' and be plotted in a graph.

The threshold is a decibel value to determine if the RSS of a link has increased enough to be considered 'broken'. As RSS values are constantly fluctuating within an environment due to multipath, path-loss, and noise interference, the threshold cannot be too low as the RSS values may fluctuate over the threshold, yet it cannot be too high as 'broken' links may not be considered to be 'broken'.

3.3 Implementing K-Means and C-Means Clustering Techniques

K-Means and C-Means clustering use scattered data points to determine a cluster centroid. To implement these two clustering algorithms, for the purpose of locating an object within an RTI network, the 'broken' RSS links in the system must be used to create data points. To do this, data points are created using the intersection of all the 'broken' links within the network.

To help improve accuracy, certain data points are removed from the data set as they would contribute a negative effect to the results gathered. These data points are the ones which lie on the edge and outside of the grid.

Once data points have been created, K-Means and C-Means clustering algorithms can be implemented within MATLAB to determine cluster centres. To determine whether the results of the experiment are able to improve accuracy, these results are compared to Stephen Hart's simulated results [1] , as well as simple localisation techniques, such as the centroid of the data points

Chapter 4. Experimental Results

This chapter aims is to determine experimentally whether K-means and C-means clustering algorithms are able to accurately determine an objects location with minimum error. The first section outlines how a small 8 node network is able to utilize the RSS values of links between nodes to determine the area in which an object is present. Once the links which have been affected by the object are determined, data points are generated, and K-Means and C-means are implemented. Section 2 scales up the original network, to a 28 node system to determine how accurate this method of localisation can be within a room environment. The results collected are then compared to Stephen Hart's simulated results [1], as well as simple localisation techniques, such as the centroid of the data points generated.

4.1 8 Node RTI Network

To determine if utilising RSS values of links between nodes can determine an objects location, three different object locations are tested within a 0.8m x 0.8m, 8 node wireless sensor network. The first location is centred in the middle of the tracking board, the second in the centre of the second quadrant, and the last is directly in front of the top left node, as can be seen in *figure 4.1*.

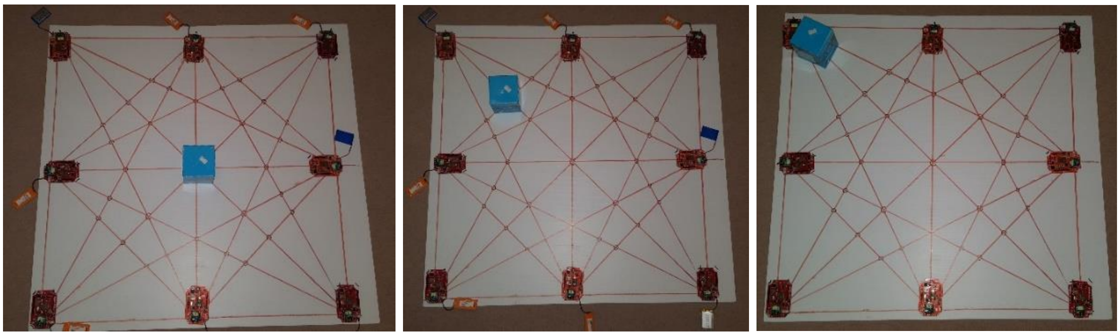


Figure 4.1: Object placed in the centre, centre of second quadrant, and top left of tracking area

4.1.1 Object Location Testing

With all the nodes in the network programmed with the Arduino code shown in Appendix B, and powered by lithium polymer 3.7V batteries, the wireless sensor network is ready to detect object positions. Each node in the system is designated a specific number to be referred to. The number of each node surrounding the system starts from node 1 in the top left corner and rotates around, in a clockwise direction, until node 8 is reached. Node 0 is used for starting the system, as well as receiving all transmitting data. Node 0 is connected to a remote computer via serial, which is running MATLAB scripts. With no object within the tracking area, the start signal, 0x66, is transmitted from MATLAB using the script InitSystem.m, as shown in Appendix C.1, to initialize the wireless system. Once the system has been initialized, MATLAB retransmits the start signal and reads all the RSS values to create the RSS calibration matrix that can be seen in figure 4.2.

	1	2	3	4	5	6	7	8
1	255	35	46	46	51	44	47	45
2	34	255	36	37	44	42	50	50
3	45	37	255	34	41	46	56	59
4	46	37	34	255	32	38	53	54
5	50	44	39	31	255	34	44	53
6	44	43	45	38	35	255	37	49
7	45	49	50	51	45	36	255	52
8	44	47	53	51	53	47	51	255

Figure 4.2 RSS calibration matrix

An object is now placed at the three specified positions on the tracking board to determine whether the wireless system is able to locate objects at different positions by utilising the change of the RSS of specific links. With the object placed in its desired position in the tracking area, the script FindObject.m, shown in Appendix C.2, is executed in MATLAB to gather the RSS values of links to determine which links are affected by the object. Once the RSS values are obtained by MATLAB, a difference matrix is created to determine the change in RSS values of the links affected by the object. The difference matrix for each position can be seen in figure 4.3.

	1	2	3	4	5	6	7	8
1	0	0	0	0	7	0	2	1
2	0	0	0	0	0	5	1	1
3	1	0	0	0	0	2	1	0
4	1	1	0	0	1	0	2	5
5	4	1	0	0	0	1	1	1
6	0	4	0	0	0	0	1	0
7	0	0	1	3	1	0	0	1
8	1	1	0	4	0	1	2	0

	1	2	3	4	5	6	7	8
1	0	0	0	0	4	7	1	0
2	1	0	0	0	0	0	5	5
3	1	0	0	1	0	1	3	4
4	2	1	0	0	1	2	1	2
5	6	0	0	0	0	0	0	2
6	6	0	0	1	0	0	1	2
7	1	4	1	1	1	1	0	1
8	0	7	0	2	1	3	1	0

	1	2	3	4	5	6	7	8
1	0	2	11	6	16	11	8	8
2	1	0	0	0	0	1	0	2
3	9	1	0	0	0	1	1	1
4	7	0	0	0	1	1	0	2
5	20	0	0	0	0	0	2	1
6	10	1	0	0	0	0	1	2
7	7	0	1	0	0	1	0	0
8	9	0	0	1	0	1	1	0

Figure 4.3 Difference matrix for centre, centre of second quadrant, and top left objects (top to bottom)

The difference matrices shown in figure 4.3 prove that the RSS values of links can fluctuate when no object is present within a certain link. The RSS is shown to only fluctuate 1-2 dB for most links. Using this knowledge, the threshold for determining whether a link is to be considered ‘broken’ or not is set to above 3dB. For different systems the RSS of ‘unbroken’ links can fluctuate to larger values as if the system increases in size, path loss and multipath can affect the system further.

To gain a visual representation of the difference matrix, PlodNodes.m, which is shown in Appendix C.3, assesses each value in the difference matrices and plots the links, which have a greater RSS value than the 3dB threshold limit. The generated graphs of the links, which are considered to be affected by the object, can be seen in figure 4.4.

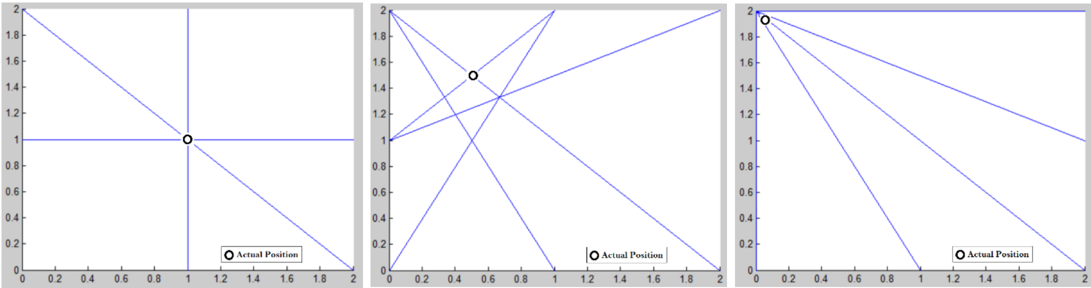


Figure 4.4 Graph of 'broken' links with an object located in the centre, centre of second quadrant, and top left (left to right)

As shown in figure 4.4, by utilising RSS values between transceiver network nodes, a close estimation of an objects location can be constructed.

4.1.2 Clustering and Centroid Techniques

As this 8 node system is only a small system, experiments conducted won't be able to determine conclusively whether K-means or C-means clustering is more accurate than the other. Yet, implementing these techniques on a small system will be able to determine whether results can be generated, even if they are accurate or not. This experiment will be the preliminary testing, before this technique is implemented within the larger 28 node network.

For the three object positions tested within the 8 node network, position 2 (centre of second quadrant), would generate superior results, compared to the other 2 positions, as more intersection of 'broken' links are evident. To generate the data points required for clustering techniques, the intersection of 'broken' links are transferred to data points as shown in figure 4.5.

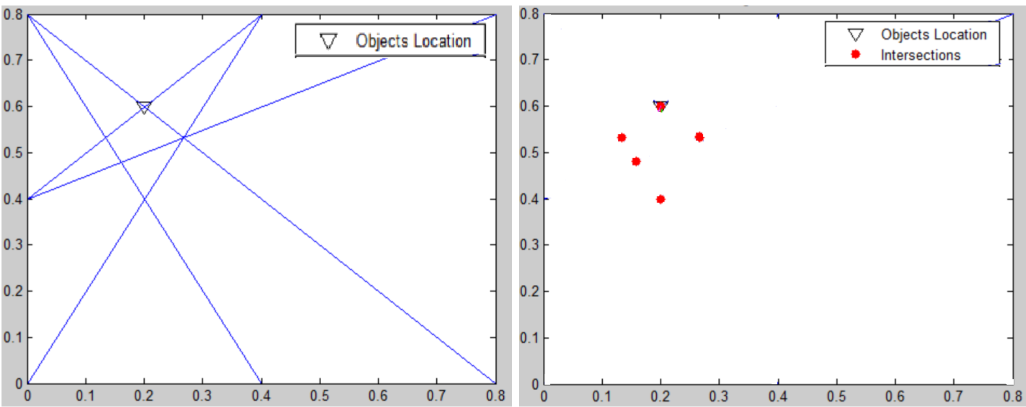


Figure 4.5: Creating data points from the intersection of 'broken' links

As seen in the figure 4.5, only 5 data points can be used to determine K-means and C-means cluster centroid. These 5 data point are implemented within the MATLAB clustering algorithm created to determine if results can be gathered. The results from clustering techniques and centroid of data points can be seen below in figure 4.6 and table 4.1.

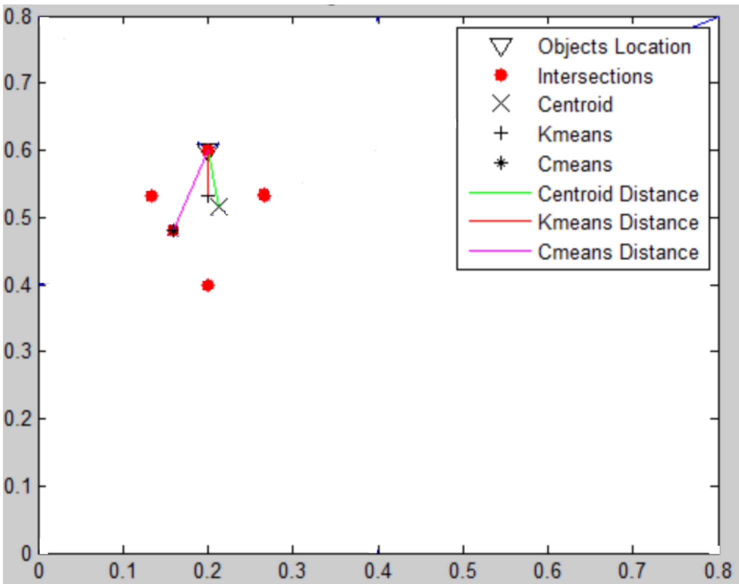


Figure 4.6: Final representation of clustering and centroid data on an 8 node system

	Position (m)	Error (m)
Object	(0.50, 1.50)	-
K-means	(0.50, 1.33)	0.1633
C-Means	(0.40, 1.20)	0.2249
Centroid	(0.53, 1.29)	0.1842

Table 4.1: Error of K-means, C-means, and centroid of data points

4.1.3 Results Summary

For the object placed in the centre location, the link between nodes 3 (0, 0) and 7 (2, 2) should have been considered to be ‘broken’ which it did not. This could be a factor of deconstructive interference of reflected/refracted signals. As some links may not be considered ‘broken’, implementing more nodes within an RTI network can in turn help to reduce this problem as more links will be introduced.

When the object was placed in the centre of the second quadrant more links are considered ‘broken’ than expected. The signals that are passing below the object, but not passing through the object, are also attenuated. This could provide a detrimental effect to the system, as links that should not be considered ‘broken’, will be, which could result in an increase in location error. Yet if an object is not placed directly on an intersection of links, this effect could help to locate the object, as nearby links will still be affected by the object.

The location of the object in front of node 1 (0, 2) worked as desired, all links corresponding to node 1 were considered ‘broken’, and the object position was determined successfully.

The clustering techniques implemented proved that clustering algorithms were able to produce results for the 8 node RTI system. The error in location for K-Means, C-Means and Centroid were 0.2582cm, 0.3557cm, and 0.2913cm respectively. Even though this was a small system, the results shown are still deemed to be fairly accurate. K-Means was shown to provide a higher accuracy compared to C-Means and Centroid.

4.2 28 Node RTI Network

To show whether K-means or C-means clustering is able to provide a higher accuracy compared to simple methods, such as the centroid, the system must be implemented within a larger room environment. For this experiment, a room of dimensions 2.7mx3.9m was used to test a 28 node RTI system with a 0.5m node spacing. This system was created by Daniel George [18], where this thesis utilises the data obtained from the system to test and compare clustering techniques.

Within this testing environment, 33 positions were marked for a person to be standing in whilst the system operated. These marked positions allows the end results to be compared to the actual object position, so an object location error for each algorithm can be obtained. The node positions around the room environment, as well as the 33 test positions can be seen below in *figure 4.7*.

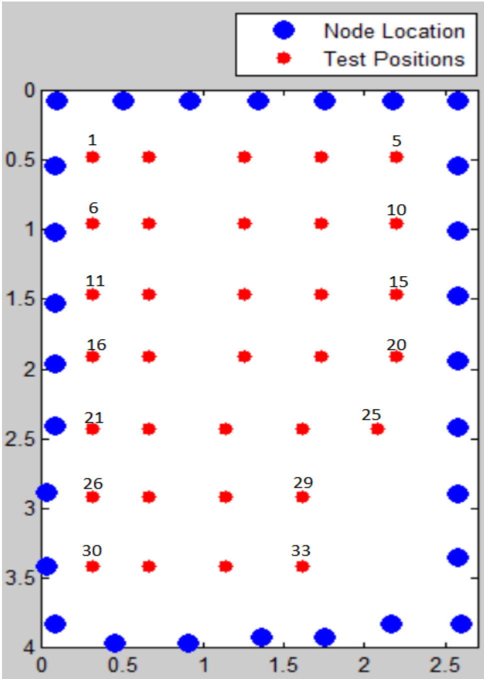


Figure 4.7: Node locations and test positions for 28 node RTI system.

4.2.1 Determining an Objects Position using K-means, C-means, and Clustering Algorithms

As the nodes in this system are spaced further from one another, compared to the first 8 node experiment, the RSS values of the links will fluctuate more due to a greater path loss and more room for interference. With this in mind, and also after some preliminary tests of running the system at several RSS thresholds, the decibel value to determine whether a link is to be considered broken or not, is set to 7dB. With a 7dB threshold set for the node links, and a person standing at one of the thirty-three positions shown above in figure 4.7, the system was activated, and results were gathered.

Shown below is figure 4.8, which gives 4 examples of how the links and clustering algorithms are used to determine a person’s position with a high accuracy. It also gives a visual representation towards where the clustering algorithms and centroid are located compared to the actual position of the objects positions. The links displayed in the figure below are the ones which have a higher RSS difference than the set 7dB threshold. The intersection of these affected links are used to determine the K-means, C-means and centroid location.

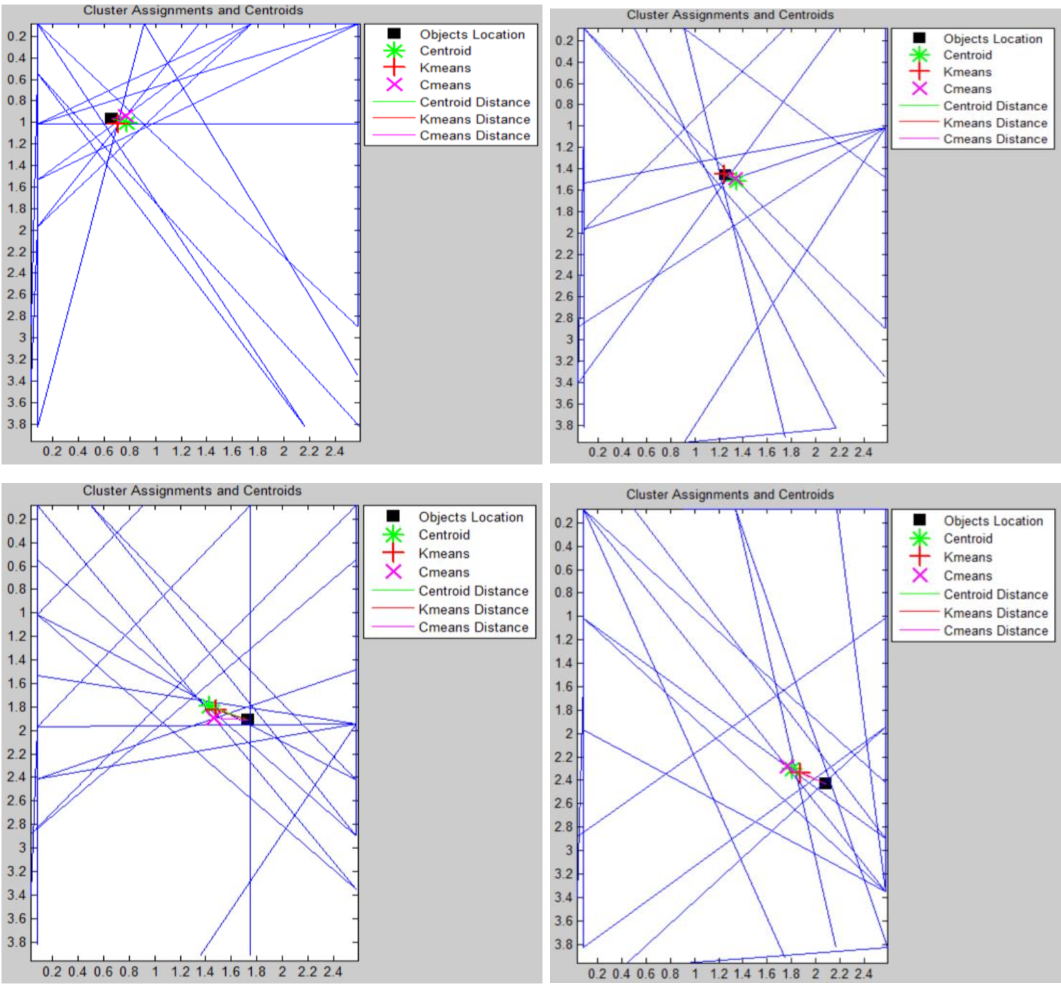


Figure 4.8: K-Means, C-means, and Centroid positions of a person at locations 7, 13, 19 and 25 (top-left to bottom-right)

As shown in the above figure 4.8, most of the links which are above the 7dB threshold are near the person's location. Yet, due to the effects of multipath and path loss, some links which are not near the person have breached the threshold, which will in turn reduce the accuracy of the clustering algorithms. These links which should not be affected by the objects location are shown to be attenuated due to multipath and path loss effects. Also during this experiment it was shown by gathering random samples of data with no object within the system, the RSS values tend to fluctuate approximately 1dB-5dB.

This experiment was conducted over 33 test positions to ensure that an average value for each localisation algorithm could be obtained. With a person standing at position 1 to 33, the system was operated each time to gain the K-means, C-means and Centroid locations for each target position. With all the data collected in MATLAB, the average value for each algorithm was calculated, and the minimum and maximum error was determined.

Table 4.2 compares the average error over all 33 locations for K-means, C-means, and Centroid algorithms. It can be seen from the table that implementing K-means clustering proved to provide the persons location with a lower average error compared to C-Means and Centroid. K-means clustering also gave the highest accuracy of 0.0791m.

Algorithm	Min Error (m)	Max Error (m)	Average Error (m)
K-Means	0.0791	0.3710	0.2445
C-Means	0.1562	0.6331	0.3288
Centroid	0.1320	0.5487	0.3465

Table 4.2: Comparison of K-Means, C-means and Centroid Error for the 28 node system

Stephen Hart's single interference experiment also confirmed, through simulated experiments, that K-Means clustering showed an enhanced capability in determining the closest cluster centroid over C-means clustering [1]. Stephen Hart gained a total simulated K-means and C-means average error of 0.0371m and 0.0404m respectively, whereas this physical experiment showed a K-Means average error of 0.2445m, and a C-means average error of 0.3288m.

Figure 4.9 graphs the relationship between error vs position for K-means, C-means, and Centroid localisation methods. The graph shows that when the target was closer to the edges of the RTI network, the results become less accurate. This is shown at the position 1, 11, 15, 20, 25, and 30, which all lay near the exterior of the system, and are shown by the peaks in the graph below. When the target position are near the middle of the tracking environment, the clustering and centroid techniques acquire a higher location accuracy. This is shown at position 8, 13, and 17, which are all near the centre of the system.

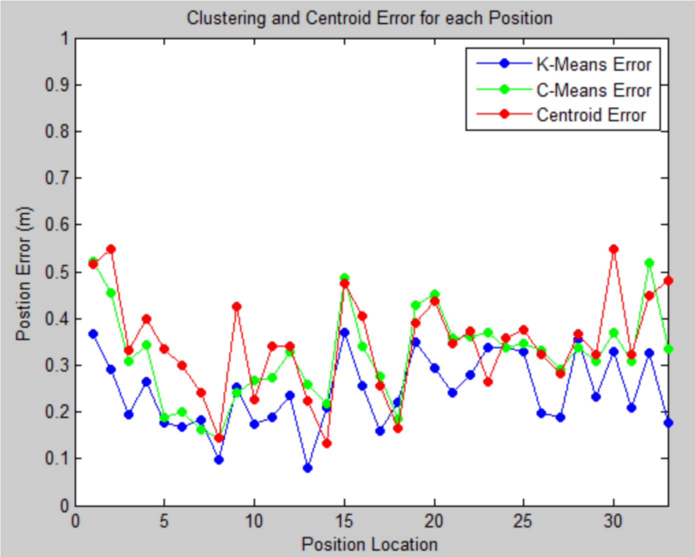


Figure 4.9: K-means, C-means, and centroid plot for the error at each target position

4.2.2 Results Summary

Through the experiment conducted, it can be seen that K-means clustering provided a higher location accuracy for a 28 node system, compared to C-means clustering and Centroid. K-means clustering at target position 13 provided the highest accuracy for the experiment with a location error of 7.91cm. Stephen Hart was able to obtain a K-means minimum error of 1cm, which is much more accurate than the obtained 7.91cm. This major difference in results is due to Stephens's experiment being conducted on a simulation in an ideal environment with no noise interference, and a perfect system. Stephen conducted his experiment by stating that the only wireless links that were affected by the object, were the ones passing directly through it. From the experiment conducted, it was shown that this isn't the case within physical tracking environments. Due to multipath, interference, and path loss, links fluctuate in values over time, resulting in some links being declared 'attenuated' when they are not. Having this affect within the physical wireless network results in the centroid of each localisation algorithm being skewed away from the object.

This experiment also showed that when the person was towards the centre of the tracking environment, the localisation accuracy increased compared to when the person was closer to the edge. This is shown by the minimum error obtained via K-means clustering being 7.91cm at position 13, and the maximum value being 37.1cm at position 15. Where position 13 is in the centre of the tracking environment, and position 15 is towards the edge. This effect happens as when the link is towards the centre of tracking environment, there are more links, resulting in a higher amount of links being attenuated. Whereas if the target is on the edge of the tracking area, there is a substantially lower amount of links, meaning that a wrong link being attenuated will result in a larger error.

C-means clustering still proved to be more accurate over a simple localisation techniques such as the centroid of the intersecting data points, with its average accuracy for the 28 node system being 32.88cm. Even though the clustering algorithm is not as accurate as K-means clustering for the single interference test, does not mean it isn't a viable option for locating multiple targets within a tracking environment. During Stephens simulated experiments he showed that C-means clustering provided a higher location accuracy compared to K-means when 2 or more people were within the tracking environment.

Chapter 5. Conclusion and Future Work

The purpose of this section is to deliver a summary of knowledge attained through research of current literature and to review the experiments and results shown in this thesis. This chapter also contains suggestion on improvements which can be made to the hardware and software components to help further the research, as well as techniques to improve the accuracy of RTI imaging.

5.1 Concluding Remarks

With the review of current literature, it is evident that RTI imaging is a growing field of interest within many applications such as security, health care, ‘smart’ homes etc. Compared to current wireless imaging techniques, RTI is dominating as it does not require the object to wear any intrusive devices/tags and it can be implemented with low cost and very little hardware. Another benefit towards RTI imaging over other localisation methods such as UWB or MIMO is that it is relatively cheap to construct. The RTI system being constructed is able to utilise low cost, low power, and cheap nodes for the system.

This thesis aimed to answer two main question, whether the RSS of signals is suitable for determining an object location within a defined environment with a high accuracy, and whether clustering techniques, such as K-means and C-means, are able to provide a higher location accuracy compared to simpler methods.

The first question was evaluated by placing an object in 3 different positions within a small 0.8mx0.8m 8 node testing environment. The results gained through the conducted experiments determined that by taking advantage of the attenuation of signals passing through objects, an RTI network is able to determine an objects position anywhere within the wireless network with a small location error.

To determine whether clustering algorithms are capable of finding an objects location with a high accuracy, the data from a 2.7mx3.9m 28 node RTI system was utilised. With implementing the clustering algorithms and centroid location MATLAB code, K-means, C-means and Centroid produced an average position error of 24.45cm, 32.88cm, and 34.65cm respectively. Using these results it can be shown that the K-means clustering algorithm provided the highest location accuracy compared to C-means and Centroid for a 28 node RTI system.

5.2 Future Work

The main objective of this annual thesis is to determine whether clustering techniques such as K-Means or C-Means is able to improve the accuracy of results gathered through RTI. As this objective has been met through the experiments conducted in this thesis the future work could possibly be based around implementing a more accurate and faster system to see if the objects location error can be reduced.

Other than implementing a faster system, future work, with close relation towards this thesis, could be conducted by introducing multiple targets to the RTI network. Stephen Hart simulated an RTI system using multiple targets and clustering algorithms. Future work could be to use the system developed in this thesis, or to redevelop the RTI system to include multiple targets and compare the clustering algorithms results obtained to Stephens simulated results.

5.2.1 Software Improvements

To determine whether a link is being considered ‘broken’ this thesis uses a threshold value. This method may cause errors within certain environments as RSS values are shown to be able to fluctuate due to multipath, noise and path loss. Another means of showing an object location is to use a weighted matrix, which determines weights for every link within a system. Smaller links will be weighted more than longer links as they will be able to give a more accurate representation of the true RSS. Implementing a weighted matrix will be able to make the system more reliable, which is needed for larger RTI systems as they are more prone to interference.

The Seeeduino code used for all the nodes in the system is shown to be able work as expected. Yet, as the size of the system increases more signals will be transmitting, and the distances travelled will be larger. To overcome any issues delays may have to be implemented within the Seeeduino code, and Acknowledgments (ACK) could be utilised to determine if every node has received the broadcasted signal.

During the course of these experiments, MATLAB proved to not be an ideal software for this thesis. MATLAB would occasionally drop packets of data, resulting in the incoming data being useless. To combat this detrimental effect, other software from reading incoming XBee data could be researched into, or even developed.

References

[1] S. Hart, "Comparison of Clustering Techniques for RTI positioning using Graph Theory," Honors Thesis, School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, 2013.

[2] J. Wilson and N. Patwari, "Radio Tomographic Imaging with Wireless Networks," *Mobile Computing, IEEE Transactions on*, vol. 9, pp. 621-632, 2010.

[3] M. Bocca, O. Kaltiokallio, and N. Patwari, "Radio tomographic imaging for ambient assisted living," in *Evaluating AAL Systems Through Competitive Benchmarking*, ed: Springer, 2013, pp. 108-130.

[4] A. Goldsmith, *Wireless communications*: Cambridge university press, 2005.

[5] L. Bao Hua, B. Otis, S. Challa, P. Axon, C. Chun Tung, and J. Sanjay, "On the Fading and Shadowing Effects for Wireless Sensor Networks," in *Mobile Adhoc and Sensor Systems (MASS), 2006 IEEE International Conference on*, 2006, pp. 51-60.

[6] O. Katircioglu, H. Isel, O. Ceylan, F. Taraktas, and H. B. Yagci, "Comparing ray tracing, free space path loss and logarithmic distance path loss models in success of indoor localization with RSSI," in *Telecommunications Forum (TELFOR), 2011 19th*, 2011, pp. 313-316.

[7] J. Olenewa, *Guide to Wireless Communications*: Cengage Learning, 2013.

[8] F. Viani, P. Rocca, G. Oliveri, D. Trinchero, and A. Massa, "Localization, tracking, and imaging of targets in wireless sensor networks: An invited review," *Radio Science*, vol. 46, 2011.

[9] G. Deak, K. Curran, and J. Condell, "A survey of active and passive indoor localisation systems," *Computer Communications*, vol. 35, pp. 1939-1954, 2012.

[10] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, pp. 54-69, 2005.

[11] H. Liu, Z.-h. Wang, X.-y. Bu, and J.-p. An, "Image reconstruction algorithms for radio tomographic imaging," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*, 2012, pp. 48-53.

[12] O. Kaltiokallio, M. Bocca, and N. Patwari, "Enhancing the accuracy of radio tomographic imaging using channel diversity," presented at the Mobile Adhoc

and Sensor Systems (MASS), 2012 IEEE 9th International Conference on Las Vegas, NV 2012.

[13] A. Pal, "Localization algorithms in wireless sensor networks: Current approaches and future challenges," *Network Protocols and Algorithms*, vol. 2, pp. 45-73, 2010.

[14] J. Wilson, N. Patwari, and F. G. Vasquez, "Regularization methods for radio tomographic imaging," in *2009 Virginia Tech Symposium on Wireless Personal Communications*, 2009.

[15] P. Berkhin, "A survey of clustering data mining techniques," in *Grouping multidimensional data*, ed: Springer, 2006, pp. 25-71.

[16] O. Maimon and L. Rokach, *Data mining and knowledge discovery handbook* vol. 2: Springer, 2005.

[17] M. Pacula. *blog.mpacula.com*.

[18] D. J. George, "The impact of node failure on localisation accuracy in Radio Tomographic Imaging networks," May 20165 2015.

[19] Digi, "XBee / XBee-PRO RF Modules," 2015.

[20] Seeed. (2015). *http://www.seeedstudio.com/wiki/Seeeduino_Stalker_v2.3*.

[21] InTechOpen. (2010). *http://www.intechopen.com/source/html/10843/media/image5.jpg*.

Appendices

Appendix A: ECTE451 Project Review

A.1 Overview

Locating a person/object within an indoor environment can be beneficial in many applications such as security, health care, ‘Smart’ homes etc. The main aim of this project is to implement a system in an indoor environment which can locate and track moving objects with a high accuracy. Radio Tomographic Imaging (RTI) is the technology which is used with the system as it can determine the attenuation of signals caused by the moving object. By setting up static nodes around the tracking environment, which transmit and receive radio frequencies, the system can use the Received Signal Strength (RSS) of the static nodes to determine the position of an object. This project also compares various algorithms, which are used to compute the location, to determine which algorithm will provide the most accurate position.

A.2 Project Description

This project aims to increase the accuracy of indoor localisation using Radio Tomographic Imaging. Different algorithms/techniques have been implemented in the past to try and improve location accuracy. My goal for this project is to review most of these algorithms and techniques to try and determine which one can give the highest accuracy, and if possible improve on this result.

The need for indoor localisation covers many areas. Tracking of elderly people within a health care environment is one of these areas. With indoor localisation we can monitor their whereabouts and track their movements to ensure that they are safe from harm. Security is another area where knowing if a person/persons is in a specific area can be beneficial.

Accuracy is the main objective of this project. Many different systems have been implemented for indoor tracking, all with the main goal of locating an object with the highest possible accuracy. By using RTI I plan on achieving a high positioning accuracy.

A.3 Project Plan

The main intention of this project is to be able to increase the indoor localisation accuracy using Radio Tomographic Imaging. To do this, certain milestones must be met to ensure that by the end of ECTE458 I am able to provide reasonable results which can confirm my thesis objective.

During ECTE451 my main milestones are based around research, and implementing a Radio Tomographic Imaging system which is able to output results, even if the results are inaccurate. Even though I have decided to use an RTI system to determine a person/objects location in an indoor environment, I will have to research into other methods used for indoor localisation. By doing this, I will have a greater understanding of other localisation methods, meaning I can compare and contrast these methods to determine why my system is better/worse.

Research will also go into which hardware/software I will have to utilize to create an efficient and accurate system. Even though I have already decided to use Seeeduino Stalkers as the nodes in the system due to budget constraints, other devices will still be researched into to again determine if mine was the most suitable.

Even though the research will continue throughout the whole project, the Hardware/Software will have to start being implemented/created once I have a basic idea of how it needs to be constructed. Asides from research, building an RTI system will take up the majority of ECTE451. I intend to implement a system which produces results by the end of first session. Doing this will allow me to be able to focus more on implementing and testing algorithms/techniques which can increase the systems accuracy in ECTE458.

The system will consist of Seeeduino Stalker nodes equipped with Xbees to receive and transmit 2.4Ghz radio frequencies. Seeing as the nodes will be placed around a room with 1 metre of separation, the amount of nodes in the system would be around 25+. As I have never worked with Xbees before, I will first have to research forums and online code for how to allow 2 Seeeduinos with Xbees to talk to each other. Once I can get 2 nodes to send packets to each other, next step is for them to be able to retrieve the Received Signal Strength (RSS) of the sending node. The RSS will be used to determine the position of an object as the signal will be attenuated when the signal passes through the object. To be able to apply algorithms to the data the computer receives from the nodes, a software will be needed.

There are different options for the software which can be used to read the xbee packets. I could either write a desktop application which has an Xbee API to handle all the packets, or use matlab which reads the data over a serial link. Research will have to go into the different software options to determine which one can generate a working model fast, yet also accurate. Once a software has been chosen, the next step is to set up the network of the 25+ nodes and feed the data into the software to determine a position of an object.

ECTE458 will be all about improving the accuracy of the object position. This may involve trying new algorithms, implementing different techniques, or even using different hardware/software. It will also entail more research into this topic to be able to determine which algorithms/techniques can be used with RTI. Also a Graphical User Interface will have to be developed and shown within the software to provide a live stream of the person's path and movements.

Radio Tomographic imaging is used as it doesn't require the use of any tags or sensors the user has to carry. This is called Passive Localisation. The RSSI-based localisation technique is being used over Time of Arrival, Time Difference of Arrival, and Angle of Arrival due to its simplicity and robustness in environments affected by multipath [1].Seeeduino stalkers have been chosen to be the nodes in the RTI system not only because they are able to talk with Xbees, but also because of the project budget. Needing 25+ nodes in a system would require more money than the given budget. Yet, since the Seeeduino Stalkers are readily available from the University of Wollongong, borrowing 25+ nodes will not take any money from my budget.

To validate my experimental results I will have a person walk a predefined path within the tracking environment. The person will walk around the environment at different speeds as well to test how reliable the system can be. To see if the results are accurate and reliable I will match them with the path the person took. This method of validation will be used with every algorithm implemented to determine which algorithm/technique provides the most accurate results.

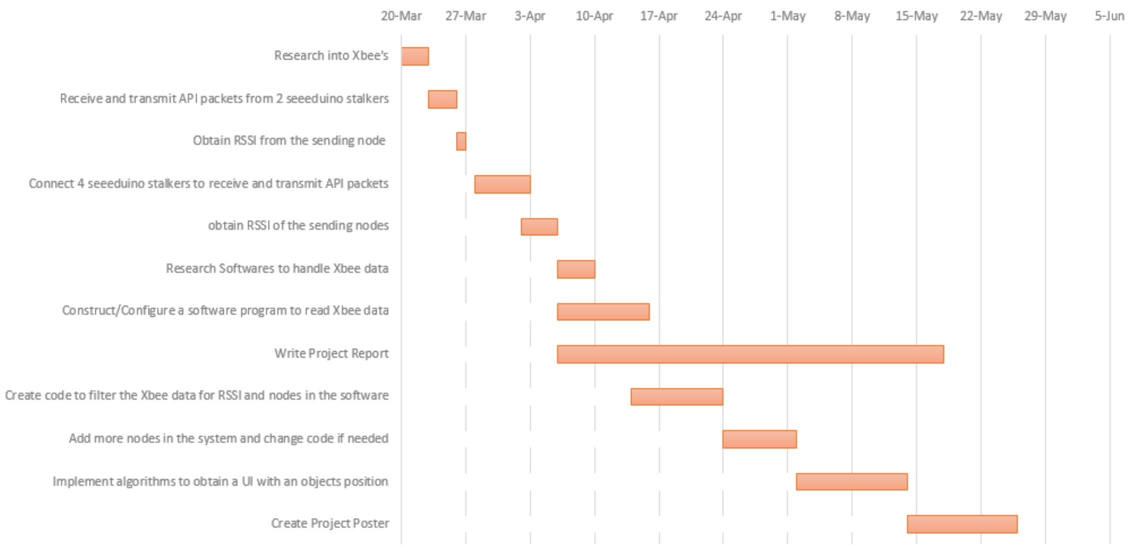
A Gantt chart can be seen in Appendix A.5 which shows all milestones which are planned to be reached in ECTE451.

A.4 Resources Required

Hardware: 25+ Seeeduino Stalkers will be needed for the nodes in the RTI system. Each of these nodes must also be equipped with an Xbee, resulting in 25+ Xbee's needed as well. To power on the Seeeduino Stalkers I can either run power cables to each of nodes, or power each on using 5V Lithium Polymer Batteries. To form a connection between the nodes and the computer I will have 1 UartsBee with another Xbee module attached to receive the systems RSS values.

Software: To read packets of data that are being sent to the UARTsBee, I will either write a desktop application to read the incoming packets, or use Matlab. Matlab would be preferred as I have basic knowledge of how to use it from my degree, yet research is still needed to determine which would be best.

A.5 ECTE451 Gantt Chart



Appendix B: ECTE458 Project Review

B.1 Overview

Locating a person/object within an indoor environment can be beneficial in many applications such as security, health care, ‘Smart’ homes etc. The main aim of this project is to implement a system in an indoor environment which can locate objects with a high accuracy. Radio Tomographic Imaging (RTI) is the technology which is used with the system as it can determine the attenuation of signals caused by the moving object. By setting up static nodes around the tracking environment, which transmit and receive radio frequencies, the system can use the Received Signal Strength (RSS) of the static nodes to determine the position of an object. This project introduces a new algorithm for determining the object position within a tracking environment. This new method is also compared to existing methods to determine if the new algorithm can achieve a higher accuracy.

B.2 Project Description

The need for indoor localisation covers many areas. Tracking of elderly people within a health care environment is one of these areas. With indoor localisation we can monitor their whereabouts and track their movements to ensure that they are safe from harm. Security is another area where knowing if a person/persons is in a specific area can be beneficial.

This project aims to increase the accuracy of indoor localisation using Radio Tomographic Imaging and Clustering Techniques. Different algorithms/techniques have been implemented in the past to try and improve location accuracy. ECTE451 required me to review most current algorithms and techniques to try and determine which one can give the highest accuracy. Within ECTE458 my goal is to introduce a new localisation algorithm known as ‘Clustering Algorithms’. These algorithms will be implemented on hardware to determine if they are able to provide a high localisation accuracy compared to currently used techniques.

ECTE458 has introduced some minor changes compared to the ECTE451 project proposal. The project plan for ECTE451 required me to research into current localisation techniques to determine which could provide a higher accuracy. After researching the topic thoroughly, I discovered that Clustering techniques have yet to be implemented within RTI localisation systems. This changed my project goal, as the main objective is to now implement clustering algorithms on a hardware platform to compare the accuracy obtained to existing algorithms.

B.3 Project Plan

The main intention of this project is to be able to increase the indoor localisation accuracy using Radio Tomographic Imaging (RTI) and Clustering Techniques. To do this, certain milestones must be met to ensure that by the end of ECTE458 I am able to provide reasonable results which can confirm my thesis objective.

RTI is used as it doesn't require the use of any tags or sensors the user has to carry. This is called Passive Localisation. The RSSI-based localisation technique is being used over Time of Arrival, Time Difference of Arrival, and Angle of Arrival due to its simplicity and robustness in environments affected by multipath. Seeeduino stalkers have been chosen to be the nodes in the RTI system not only because they are able to talk easily with Xbees, but also because of the project budget. Needing 20 nodes in a system would require more money than the given budget. Yet, since the Seeeduino Stalkers are readily available from the University of Wollongong, borrowing 20 nodes will not take any money from the allocated budget.

The wireless system developed in ECTE451 consists of 8 Seeeduino Stalker nodes equipped with Xbees to receive and transmit 2.4Ghz radio frequencies. These nodes are used to construct a wireless sensor network which is able to determine an objects location by measuring the attenuation of the transmitted signals. If a transmitted signal is attenuated, it shows that an object is affecting the link between the transmitter and receiver node. Once all the transmitted signals are analysed, all the attenuated links are plotted on a graph to determine the objects position

ECTE458 will focused towards improving the accuracy of the object position as well as increasing the systems size. This will involve implementing different clustering techniques through MATLAB to determine if they are capable of increasing the location accuracy. Seeing as the nodes will be placed around a room with 1 metre of separation, the amount of nodes in the system would need to be increased from 8 to 20 nodes

The previous outcome for ECTE451 was to implement various indoor tracking algorithms such as the least square solution, and linear back projection, to determine which algorithm gave the highest accuracy. After thorough research, it was evident that clustering algorithms such as K-Means and C-Means have yet to be incorporated and tested within RTI systems. This finding changed my initial thesis strategy to implement clustering algorithms instead of already existing algorithms to improve location accuracy.

To validate my solution I will run two different experiments with an object within the same location. The first experiment will locate the object by finding the object positon by using the centroid of the intersection of attenuated links. Whereas the second experiment will locate the object by implementing clustering algorithms of the points of intersections of attenuated links. These two experiments will be able to verify whether clustering techniques are capable of increasing the location accuracy compared to simple localisation techniques.

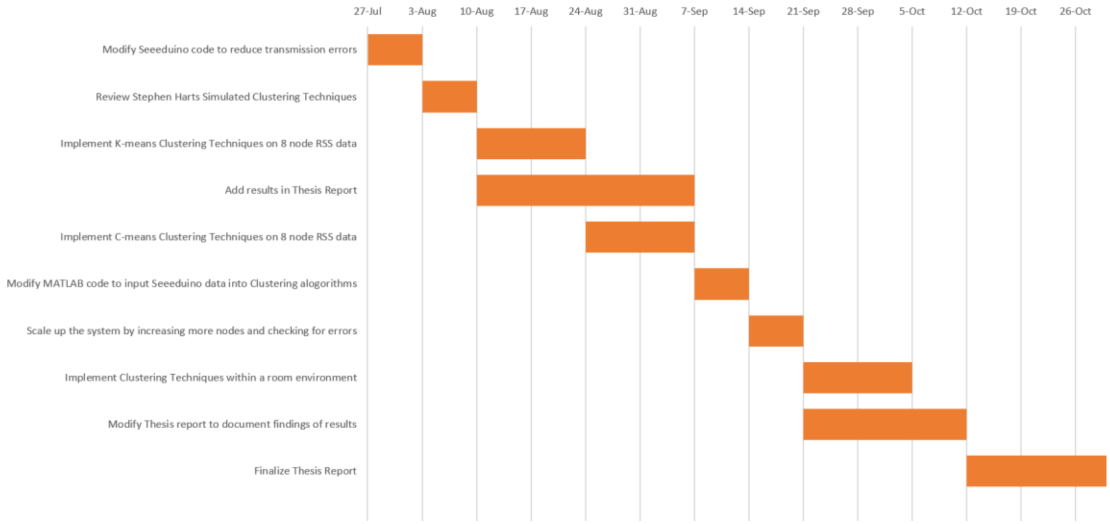
All milestones and deliverable to be achieved by the end of ECTE458 can be seen in B.5

B.4 Adaption of Supervisor and Examiners Feedback in the ECTE451 Report

The examiner provided some excellent feedback with regards to my ECTE451 report. Careful review will go towards the areas of the report which should be changed. Grammar errors and image referral errors were also brought to my attention, which will need to be modified throughout this final semester.

A more elaborate Gantt chart will also need to be constructed within ECTE458, which will show a more accurate break down structure of what experiments and report changes must be made.

B.5 Gantt Chart



Appendix B: Seeeduino Code

```
#include <XBee.h>
#include <SoftwareSerial.h>

SoftwareSerial com(8, 9); //debugging
XBee xbee = XBee();

uint8_t rssData[8] = {};     //stores [Rss1, Rss2, Rss3]  8 for 8 nodes
uint8_t xbeeData[8] = {};    //stores xbee payload   8 for 8 nodes
uint8_t myCmd[] = {'M','Y'}; // AT Command used to obtain address of xbee
uint8_t xbeeAddress;         // stores the address of the connected xbee

AtCommandRequest atRequest = AtCommandRequest(myCmd);
AtCommandResponse atResponse = AtCommandResponse();
Rx16Response rx16 = Rx16Response();

void setup()
{
  memset(rssData,0xFF,sizeof(rssData));
  Serial.begin(57600); //connects to xbee at a baud rate of 57600 (can try increasing value later)
  delay(1000);
  sendAtCommand();     //gets Xbee MY address
}

void loop()
{
  xbee.readPacket(); // reads incoming xbee packets
    if (xbee.getResponse().isAvailable()) //if a packet is available
    {
       //delay(200);

       if (xbee.getResponse().getApiId() == RX_16_RESPONSE) //if the packet recieved is a RX 16 
response type
       {
          xbee.getResponse().getRx16Response(rx16); // rx16 stores the xbee API data

          if(rx16.getData(0) == 0x66 && xbeeAddress == 0x01)//if start signal sent from matlab (0x66), and 
xbee address = 1
          {
            Tx16Request tx16 = Tx16Request(0xFFFF, rssData, sizeof(rssData));
```

```
          delay(400);
          xbee.send(tx16);
        }
        else if (rx16.getRemoteAddress16() > 0) // if sending address is above 0x00
        {
         rssData[rx16.getRemoteAddress16() - 1] = rx16.getRssi(); // rss[address-1] = RSSI value of recieved address

         if( xbeeAddress == (rx16.getRemoteAddress16() + 1)) // if it is the next seeeduino to send (goes from 1 to 2 to 3... etc
         {
           Tx16Request tx16 = Tx16Request(0xFFFF, rssData, sizeof(rssData));
           delay(400);
           xbee.send(tx16);
         }
        }
      }
    }
}

//Function which finds MY address of connected xbee
void sendAtCommand()
{
  xbee.send(atRequest);
  // wait up to 5 seconds for the status response
  if (xbee.readPacket(5000))
  {
    if (xbee.getResponse().getApiId() == AT_COMMAND_RESPONSE)
    {
      xbee.getResponse().getAtCommandResponse(atResponse);

      if (atResponse.isOk())
      {
        //Only need first value of address as i wont have anymore than 255 nodes
        xbeeAddress = atResponse.getValue();
      }
    }
  }
}
```

Appendix C: MATLAB Code

C.1: InitSystem.m : Initializing the Wireless System

```
N = 8; %amount of nodes
L = (N^2 - N)/2; %amount of links
RssObject = zeros(N);

%Creates a Serial port object named XbeeCOM
XbeeCOM = serial('COM4');

%Sets all parameters for Serial Communication
set(XbeeCOM, 'BaudRate', 57600);
set(XbeeCOM, 'Parity', 'none');
set(XbeeCOM, 'StopBits', 1);
set(XbeeCOM, 'FlowControl', 'Software');
set(XbeeCOM,'Timeout',5);

startSignalHex =
hex2dec({'7E','00','06','01','01','FF','FF','01','66','98'});

%To calculate CHECKSUM: Not including frame delimiters and length, add
all bytes keeping only the lowest 8 bits
%of the result and subtract from 0xFF.
%To verify: Add all bytes (include checksum, but not the delimiter and
length). If the checksum is
%correct, the sum will equal 0xFF.

fopen(XbeeCOM);

fwrite(XbeeCOM, startSignalHex);
pause(5);
flushinput(XbeeCOM);
pause(1)

fwrite(XbeeCOM, startSignalHex);

transmitStatus = fread(XbeeCOM,5,'uint8');
XbeeData = fread(XbeeCOM, [N+7,N], 'uint8');

fclose(XbeeCOM);

RSSdata = XbeeData(7:N+6,1:N); %stores only the RSS values of the xbee
data in RSSIdata
```

C.2 FindObject.m : Detecting an Object

```
fopen(XbeeCOM);

fwrite(XbeeCOM, startSignalHex);
pause(5);
flushinput(XbeeCOM);
pause(1);
fwrite(XbeeCOM, startSignalHex);
```

```
transmitStatus = fread(XbeeCOM,5,'uint8');
XbeeData = fread(XbeeCOM, [N+7,N], 'uint8');

fclose(XbeeCOM);

RssObject = XbeeData(7:N+6,1:N);

LinkBreak = abs(RssObject - RSSdata);
```

C.3 PlotNode.m : Plotting Broken Links

```
nodeLoc = [0 2;1 2;2 2;2 1;2 0;1 0;0 0;0 1];

hold on
for i=1:N
    for j=1:N
        if LinkBreak(i,j) > 3
        plot ([nodeLoc(i,1) nodeLoc(j,1)],[nodeLoc(i,2) nodeLoc(j,2)])
        end
    end
end
```