Ad-Hoc Wireless MESH Network Implementation for Rescue Robots

Peter D. Turner
BEng (Comp) (Hons)

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Engineering

School of Electrical Engineering and Computer Science
The University of Newcastle
Callaghan, NSW 2308
Australia

April, 2013
The thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to the final version of my thesis being made available worldwide when deposited in the University’s Digital Repository**, subject to the provisions of the Copyright Act 1968. **Unless an Embargo has been approved for a determined period.

Peter D. Turner
Acknowledgments

I would like to acknowledge all those who have helped me during this research. Initially I’d like to thank Dr. Jamil Khan for being my supervisor during this research, his guidance, advice and patience have been greatly appreciated.

I’d also like to thank Dr. Michael Dickinson from the School of Design Communication and IT. Working with Michael to design the robot bears body pieces taught me a great deal about practical Industrial Design techniques. Michael’s advise whilst writing this thesis was also invaluable.

I’d like to acknowledge the technical staff from the Discipline of Electrical and Computer Engineering for their help, particularly in the generation and population of printed circuit boards.

Finally I’d like to thank my family for their love, help and support whilst I pursued this personal goal. I simply could not have done this without them.
Table of Contents

Abstract ........................................................................................................................................... 1
1. Introduction................................................................................................................................... 3
   1.1 Outline of Thesis .................................................................................................................. 5
2. Urban Search and Rescue Robots ............................................................................................. 7
   2.1 Introduction........................................................................................................................... 7
   2.2 Technical Overview of Urban Search and Rescue Robots .............................................. 10
      2.2.1 USAR Robot Types ................................................................................................. 13
      2.2.2 USAR Robot Sensors ............................................................................................ 18
      2.2.3 USAR Robot Processing ....................................................................................... 20
      2.2.4 USAR Robot Communication .............................................................................. 22
      2.2.5 USAR User Interfaces ......................................................................................... 24
   2.3 Urban Search and Rescue Robot Designs ........................................................................... 26
      2.3.1 USAR Quadruped Robot Platform ......................................................................... 28
         2.3.1.1 Mechanical Design ...................................................................................... 31
         2.3.1.2 Processing System Hardware Design ......................................................... 36
         2.3.1.3 Processing System Algorithm .................................................................. 57
      2.3.2 USAR Tank based Robot Platform ......................................................................... 60
         2.3.2.1 Mechanical Design ...................................................................................... 61
         2.3.2.2 Processing System Hardware Design ......................................................... 65
         2.3.2.3 Processing System Algorithm .................................................................. 73
   2.4 Summary ............................................................................................................................... 76
3. Wireless Communication ............................................................................................................. 77
   3.1 Introduction........................................................................................................................... 77
   3.2 Overview of Wireless Network Standards ......................................................................... 87
   3.3 Wireless System for USAR Robots .................................................................................. 108
   3.4 Summary ............................................................................................................................... 119
4. Wireless USAR Robot Guidance System ................................................................................... 121
   4.1 Introduction........................................................................................................................... 121
   4.2 Wireless USAR Robot Guidance System Overview ....................................................... 122
   4.3 Wireless Node Hardware Design ..................................................................................... 128
      4.3.1 Wireless Node Hardware ..................................................................................... 128
Table of Figures

Figure 2.1 Overview of USAR robot functionality ................................................. 12
Figure 2.2 Boston Dynamic’s BigDog Quadruped Robot ..................................... 15
Figure 2.3 Coroware Explorer 4WD Articulated Base Robot ................................. 16
Figure 2.4 Tank based USAR robot with sub-tracks .............................................. 17
Figure 2.5 Early conceptual design of a Quadruped Robot ................................... 29
Figure 2.6 Overview of the tank based USAR robots functionality ....................... 30
Figure 2.7 Skeleton of a Polar Bear, Ursus Martimus ........................................ 31
Figure 2.8 Structural design of robot bear based around Dynamixel modules ....... 32
Figure 2.9 Assembled and dimensioned robot bear .............................................. 33
Figure 2.10 Assembled and dimensioned robot bear ............................................. 34
Figure 2.11 Additional DOF that gave robot bear unique movements .................... 35
Figure 2.12 Block diagram of USAR Quadruped robot processing system .......... 37
Figure 2.13 HyInt processing system ................................................................. 38
Figure 2.14 Mounting of the HyInt processing system ........................................ 40
Figure 2.15 Modules use to acquire sensor data for USAR quadruped ................. 45
Figure 2.16 Packet Structure for HyInt to HySense modules Communications ........ 45
Figure 2.17 USAR quadruped open head view ................................................... 49
Figure 2.18 Overview of Packet Transfer between Control Node and Mobile Node... 51
Figure 2.19 Packet Structure for Mobile Node to FitPC2 Communications ............ 51
Figure 2.20 Packet Structure for Dynamixel Communications ............................ 54
Figure 2.21 Dynamixel location overview .......................................................... 55
Figure 2.22 Algorithm proposed for HyInt ......................................................... 59
Figure 2.23 Overview of the tank based USAR robots functionality ..................... 60
Figure 2.24 MMP-40 manufactured by The Machines Lab .................................. 62
Figure 2.25 USAR Robot Drop Node Deployment System ................................. 63
Figure 2.26 Block Diagram of JN5148-001-Myy based Wireless Node ............... 64
Figure 2.27 Robotic Arm with Pan/Tilt Camera .................................................. 65
Figure 2.28 Block Diagram of Processing System added to MMP-40 tracked robot 66
Figure 2.29 Packet Structure for Mobile Node to FitPC2 Communications .......... 67
Figure 2.30 Overview of Packet Transfer between Control Node and Mobile Node... 69
Figure 2.31 Block Diagram of Motor Control Data Acquisition ......................... 72
Figure 2.32 Algorithm implemented in Fit-PC2 ............................................................. 75
Figure 3.1 RSSI versus received signal power for Jennic JN-5148 .............................. 80
Figure 3.2 RSSI versus transmission range for Jennic JN-5148 ................................. 82
Figure 3.3 Time of Flight measurement technique ....................................................... 82
Figure 3.4 Time of Flight measurement implementation within JN5148 ....................... 83
Figure 3.5 Time of Flight packet transfers for the JN5148 ............................... 84
Figure 3.6 RSSI versus transmission range for Jennic JN-5148 ................................. 85
Figure 3.7 Traditional Cellular network structure ....................................................... 89
Figure 3.8 MESH network structure ........................................................................ 91
Figure 3.9 802.11 Protocol Stack ............................................................................. 94
Figure 3.10 802.11 Collision Free Period Repetition Interval ........................................ 95
Figure 3.11 802.11 Operating Modes ..................................................................... 96
Figure 3.12 Bluetooth Protocol Stack ..................................................................... 97
Figure 3.13 Bluetooth Piconet ............................................................................... 98
Figure 3.14 Bluetooth Scatternet .......................................................................... 98
Figure 3.15 802.15.4 Protocol Stack ....................................................................... 99
Figure 3.16 802.15.4 Network Topologies ................................................................. 101
Figure 3.17 Superframe structure with GTS .............................................................. 102
Figure 3.18 ZigBee Protocol Stack ......................................................................... 103
Figure 3.19 ZigBee Network Topologies ................................................................. 106
Figure 3.20 Wireless guidance system – Initial System ............................................ 109
Figure 3.21 Wireless guidance system – path B1 failing ........................................... 110
Figure 3.22 Wireless guidance system – path B has failed ...................................... 111
Figure 3.23 Wireless guidance system – path C failing, new FFD deployed .......... 112
Figure 3.24 Wireless guidance system – path A1 failing, new FFD deployed .......... 113
Figure 3.25 Wireless guidance system – path C1 failing, new FFD deployed .......... 114
Figure 3.26 Wireless guidance system – result of 3 deployed FFDs ......................... 115
Figure 3.27 Wireless guidance system – Network Topology ..................................... 115
Figure 3.28 Wireless guidance system – partially connected MESH .......................... 116
Figure 3.29 Wireless guidance system – comparison of network coverage ............ 117
Figure 3.30 Wireless guidance system – comparison of initial to final ..................... 118
Figure 4.1 802.15.4 model to USAR robotic guidance system model .................... 122
Figure 4.2 Overview of wireless robotic guidance system implementation ............ 123
Figure 4.3 Initial wireless USAR robot guidance system configuration ................. 125
Figure 4.4 Localisation via trilateration using RSSI and TOF ............................... 127
Figure 4.5 JN5148 Wireless microcontroller block diagram Ref:\[66\] ...................... 129
Figure 4.6 Block diagram of JN5148-001-Myy based node.................................. 130
Figure 4.7 JN5148-001-M03 Wireless microcontroller module block diagram ...... 131
Figure 4.8 JenNet Protocol Stack ........................................................................... 132
Figure 4.9 JenNet application code overview Ref:\[67\] ................................................ 133
Figure 4.10 Wireless USAR robot guidance system structure................................. 136
Figure 4.11 Drop Node Deployment Sequence – Network Perspective ................. 138
Figure 4.12 Drop Node Deployment Sequence – Control Node ......................... 139
Figure 4.13 Drop Node Deployment Sequence – Drop Node ............................... 140
Figure 4.14 Drop Node Failure – Network Perspective ............................................. 141
Figure 4.15 Drop Node Failure – Network Recovery Algorithm ............................. 142
Figure 4.16 Control Node State Machine ................................................................. 143
Figure 4.17 Initial wireless USAR robot guidance system configuration ............... 144
Figure 4.18 Position Node State Machine ................................................................. 145
Figure 4.19 Drop Node State Machine ................................................................. 146
Figure 4.20 Mobile Node 1 (Tank-based USAR robot) State Machine ................... 148
Figure 4.21 Mobile Node 2 (Quadruped USAR robot) State Machine ................... 149
Figure 4.22 Overview of configuration for End-to-End delay test ........................... 150
Figure 4.23 End-to-End Delay test results .............................................................. 152
Figure 4.24 Packet Sniffer screen capture .............................................................. 153
Figure 4.25 Control Node diagnostics Screen Capture ......................................... 153
Figure 5.1 Trilateration to determine N from A, B and C .................................... 160
Figure 5.2 Trilateration using Clustering/Centroids Algorithm .............................. 162
Figure 5.3 Trilateration using Clustering/Centroids ............................................ 163
Figure 5.4 Centroid for 3 Intersecting Circles ...................................................... 164
Figure 5.5 Geometry used to determine intersection of 2 circles ...................... 165
Figure 5.6 C Simulator flowchart for Trilateration using Centroids algorithm ....... 167
Figure 5.7 Simulation scenario using 3 nodes for localisation ............................. 168
Figure 5.8 Simulation scenario using 4 nodes for localisation ............................. 169
Figure 5.9 Successful simulations completed .................................................... 171
Figure 5.10 Distance error vs TOF error .............................................................. 171
Figure 5.11 Plot of calculated \((x, y)\) results for TOF Error of 3m ........................................ 172
Figure 5.12 Jennic 802.15.4 Software Architecture ................................................................. 173
Figure 5.13 Events required to associate a Coordinator and an End Device .................. 174
Figure 5.14 Events required to associate a Coordinator and an End Device .................. 176
Figure 5.15 TOF Measurement results data structure ............................................................... 177
Figure 5.16 LabView Data Logging Graphical User Interface .................................................... 178
Figure 5.17 TOF Data from JN5148-001-M03 indoors – distance 10m .......................... 179
Figure 5.18 RSSI vs Distance for JN5148-Myy in an Outdoor Environment ..................... 182
Figure 5.19 TOF vs Distance for JN5148-Myy in an Outdoor Environment ...................... 182
Figure 5.20 RSSI vs Distance for JN5148-Myy in an Indoor Environment ......................... 184
Figure 5.21 TOF vs Distance for JN5148-Myy in an Indoor Environment ......................... 184
Figure 5.22 Indoor localisation overview ............................................................................. 186
Figure 5.23 Outdoor localisation overview ............................................................................. 187
Figure 5.24 Graphical representation of Outdoor Localisation Tests ............................... 189
Figure 5.25 Position 15 and 18 Clusters and Centroids ...................................................... 190
Figure 5.26 Indoor localisation detailed overview ................................................................. 191
Figure 5.27 Indoor localisation wireless signal path Fixed Node A .................................... 194
Figure 5.28 Graphical representation of Indoor Localisation Tests ..................................... 195
# Glossary

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFH</td>
<td>Adaptive Frequency Hopping</td>
</tr>
<tr>
<td>A2D</td>
<td>Analog to Digital</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BPSK</td>
<td>Binary Phase-Shift Keying</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>CSS</td>
<td>Chirp Spread Spectrum</td>
</tr>
<tr>
<td>CFP</td>
<td>Collision Free Period</td>
</tr>
<tr>
<td>CAP</td>
<td>Contention Access Period</td>
</tr>
<tr>
<td>CFP</td>
<td>Contention Free Period</td>
</tr>
<tr>
<td>CP</td>
<td>Contention Period</td>
</tr>
<tr>
<td>DSSS</td>
<td>Direct Sequence Spread Spectrum</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree of Freedom</td>
</tr>
<tr>
<td>UWB</td>
<td>Direct Sequence Ultra-wideband</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>DTMF</td>
<td>Dual-Tone Multi-Frequency</td>
</tr>
<tr>
<td>EDGE</td>
<td>Enhanced Data Rates for GSM Evolution</td>
</tr>
<tr>
<td>EVO</td>
<td>Evolution-Data Optimisation</td>
</tr>
<tr>
<td>FHSS</td>
<td>Frequency Hopping Spread Spectrum</td>
</tr>
<tr>
<td>FSPL</td>
<td>Free Space Path Loss</td>
</tr>
<tr>
<td>GTS</td>
<td>Guaranteed Time Slots</td>
</tr>
<tr>
<td>GFSK</td>
<td>Guassian Frequency-Shift Keying</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile communications</td>
</tr>
<tr>
<td>HSPA+</td>
<td>Evolved High-Speed Packet Access</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronic Engineers</td>
</tr>
<tr>
<td>IFS</td>
<td>Interframe Space</td>
</tr>
<tr>
<td>IF</td>
<td>Intermediate Frequency</td>
</tr>
<tr>
<td>IR</td>
<td>InfraRed</td>
</tr>
<tr>
<td><strong>Notation</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>IMT</td>
<td>International Mobile Telecommunications</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial, Scientific and Medial radio bands</td>
</tr>
<tr>
<td>kbps</td>
<td>kilobyte per second</td>
</tr>
<tr>
<td>LRF</td>
<td>Laser Range Finder</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MMS</td>
<td>Multimedia Messaging Service</td>
</tr>
<tr>
<td>NWK</td>
<td>Network Layer</td>
</tr>
<tr>
<td>OQPSK</td>
<td>Offset Quadrature Phase-Shift Keying</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency Division Multiplexing</td>
</tr>
<tr>
<td>PSSS</td>
<td>Parallel Sequence Spread Spectrum</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer</td>
</tr>
<tr>
<td>PCF</td>
<td>Point Coordination Function</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indicator</td>
</tr>
<tr>
<td>TDMA</td>
<td>Time Division Multiple Access</td>
</tr>
<tr>
<td>TOF</td>
<td>Time of Flight</td>
</tr>
<tr>
<td>WAP</td>
<td>Wireless Application Protocol</td>
</tr>
<tr>
<td>WPAN</td>
<td>Wireless Personal Area Network</td>
</tr>
<tr>
<td>WWW</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Aerial Vehicle</td>
</tr>
<tr>
<td>USAR</td>
<td>Urban Search And Rescue</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunication Systems</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over Internet Protocol</td>
</tr>
</tbody>
</table>
Abstract

Urban Search and Rescue (USAR) robots have the potential to save the lives not only of trapped victims but also rescuers. Due to the nature of the environment in which USAR robots operate, there are number of challenges that must be overcome to increase their effectiveness. Natural disasters, such as earthquakes, tsunamis and cyclones, can cause wide spread damage to buildings and infrastructure resulting in an environment in which maneuverability is difficult and existing communications infrastructure is inoperable. This study is focused on creating a wireless robotic guidance system for a team of teleoperated USAR robots which are able to maneuver quickly and efficiently over rough terrain within a disaster zone. The research explores robotics and wireless communication systems that will advance the body of knowledge in USAR.

Two robots with differing propulsion systems have been designed as part of this research. The ability of canine search and rescue dogs to maneuver quickly through areas strewn with rubble inspired the design of a quadruped USAR robot. A tank based USAR robot featuring a 5 DOF robotic arm was the second USAR designed. These two USAR robots are able to operate as a team. The quadruped robot performs the initial reconnaissance and may be followed by the tank based robot when the searched area has been deemed structurally sound.

Both USAR robots are teleoperated across an ad-hoc 802.15.4 based wireless robotic guidance system which has been designed as part of this research. The wireless robotic guidance system utilises drop nodes to adapt to its operational environment. When the communication link to the USAR robot deteriorates a drop node is deployed, the wireless network is reconfigured and the USAR robot is able to continue moving deeper into the disaster zone. The resulting multi-hop wireless network overcomes the network coverage issues experienced by traditional single hop wireless networks. Using RSSI and TOF distance measurements and trilateration based localisation algorithms the wireless robotic guidance system is capable of accurately positioning an USAR robot within a disaster zone.
This thesis records the design, experimentation and thinking undertaken to complete this research and delivers a localisation system that advances the state of the art in USAR robotics.
1. Introduction

The collapse of the World Trade Center in 2001 is the first documented disaster where Urban Search and Rescue (USAR) robots were deployed to search for victims [1]. Although no victims were found at the World Trade Center site during this deployment, USAR robots were able to search through the rubble to a depth of 5m to 20m. Prior to the use of USAR robots cameras mounted on poles had been used, this method allowed only 2m of penetration into the disaster zone. Thus, the potential benefits of using USAR robots at disaster sites were realised.

Since 2001 USAR robots have been deployed at twenty six documented disasters [2] but have yet to locate a single survivor. One of the reasons for this is that the chance of trapped victims surviving more than 48hrs are statistically small and it takes on average 4.5 days for an USAR robot deployment to reach the disaster zone. As USAR robots become more widely available to first response rescue teams it is hoped the deployment time will be faster and lead to better results.

USAR operations can be inherently dangerous, potentially placing the lives of rescuers at risks. The use of robots to assist in USAR operations not only has the potential to locate survivors but also reduce the risks faced by rescue workers. As the USAR robot would be performing the search for survivors, the rescuers exposure to the dangerous environment, harmful chemicals and radiation could be minimised. Whilst there are many benefits to be gained from the use of USAR robots, a number of challenges remain to increase their effectiveness.

To operate successfully, USAR robots must have a reliable communication link to rescuers outside the disaster zone. Autonomous USAR robots must have the capability to transfer their observations to rescuers whilst semi-autonomous USAR robots need a reliable communication link for guidance commands. There are two methods by which this communication can be established. A tethered cable may be used between the USAR robot and the rescue team. This results in a reliable communication link but can severely hamper the USAR robots ability to traverse through a disaster zone as the cable...
can easily become entangled within rubble and debris. Wireless communication is the other alternative and removes the mobility restraint imposed by the tethered cable.

However, disaster zones, such as collapsed buildings, create highly variable radio propagation conditions making wireless communications unreliable. In [1] Murphy documents that robots were actually lost in the ruins of the World Trade Centre in 2001 due to a loss of wireless communications. In [2] Murphy also claims that ground based USAR robots are almost always tethered because “wireless networks are unreliable in dense rubble with metal rebar, mines, or inside commercial structures such as nuclear power plants”.

The possible increase in mobility achieved by the removal of the tethered cable has provided the motivation for research in the areas of USAR robot mobility within a disaster area and wireless guidance systems specifically for USAR robots. My research will focus on:

1. Creating highly mobile USAR robots, and
2. Creating a wireless robotic guidance system that is able to:
   (i) Communicate with multiple USAR robots simultaneously within a disaster area, and
   (ii) Accurately determine the locations of USAR robots within a disaster zone.

As well as contributing to the body of knowledge in USAR robotics this research has the potential to save the lives and minimise the risks faced by rescuers.
1.1 Outline of Thesis

The scope of research covered in thesis is spread across two distinct fields, robotics and wireless communications. To aid in following the logic of the arguments and discussions presented in this thesis the robotic research and subsequent results are presented first in Chapter 2 followed by the wireless communication research and results which are presented in Chapters 3 through to Chapter 5.

Within this chapter the challenges faced by USAR robots are stated and the problems to be investigated by the research are addressed. Chapter 2 begins with a literature review that presents the current status of USAR robot research. The detailed design of two USAR robots, a quadruped robot and a tank based robot, are presented in the remainder of the chapter. Chapter 3 begins by reviewing wireless communication systems that could be used for a wireless robotic guidance system. As a result of this review an ad-hoc 802.15.4 based wireless network is selected to form the basis of a robotic wireless guidance system. A method of dynamically building the 802.15.4 ad-hoc wireless network to match the requirements of the disaster zone is then presented in the final section of the following chapter. Chapter 4 then presents the design of an 802.15.4 based wireless guidance system. Within this chapter the hardware design is presented along with the algorithms used at a network and a node level. Chapter 5 then discusses how the wireless robotic guidance system can be used to accurately determine the position of an USAR robot within a disaster zone. A simulation of a selected localisation algorithm is then discussed and results presented before the chapter concludes by presenting the localisation results obtained using RSSI and TOF distance based measurements. Finally Chapter 6 will conclude this thesis by summarising the findings and discussing additional research that could be performed to further enhance the results that have been achieved.
2 Urban Search and Rescue Robots

2.1 Introduction

USAR robots are now commonly used to help search for survivors and assess damage after natural disasters, such as earthquakes, tsunamis and hurricanes. USAR robots were used after the Japanese Tsunami in 2011 to determine the scale of the damage to the Fukushima Nuclear Reactor. Here Unmanned Aerial Vehicles (UAV) were used to gather images from the air [3], whilst iRobot’s Packbot was used to enter the disaster area and obtain images inside the damaged buildings [4], helping engineers to determine what course of action needed to be taken to make the facility as safe as possible. Obviously the use of robots in this situation saved many people from entering the ruins of the Fukushima Nuclear Reactor where they would have been exposed to life threatening high levels of radiation.

Another natural disaster, the Hanshi-Awaji earthquake which hit the Japanese city of Kobe in 1995, inspired the creation of RoboCup [5] Rescue Robot League (RRL) [6]. This competition was first held in 2001, with the aim of the competition being to “encourage the transfer of academic research into the disaster-rescue domain, and to encourage research in a socially significant real-world domain” [7]. “The goal of the urban search and rescue robot competitions is to increase awareness of the challenges involved in search and rescue applications, provide objective evaluation of robotic implementations in representative environments, and promote collaboration between researchers. It requires robots to demonstrate their capabilities in mobility, sensory perception, planning, mapping, and practical operator interfaces, while searching for simulated victims in unstructured environments” [8]. The RRL has continued to grow with 27 teams from 15 different countries participating at RoboCup 2013, to be held in Eindhoven, The Netherlands.

At RoboCup RRL events “teams demonstrate their capabilities in mobility, sensory perception, localisation and mapping, mobile manipulation, practical operator interfaces, and assistive autonomous behaviors to improve remote operator performance
and/or robot survivability while searching for simulated victims in a maze of terrains and challenges” [8]. Robots competing at RoboCup RRL events are controlled wirelessly by remote operators, but there are sections of the arena that are designated ‘Radio Drop-out Zones’ where robots can demonstrate autonomous navigation and are encouraged to do so.

USAR robots are also playing an increasing role in the Military and Law Enforcement. Here these robots are able to enter hostile environments and either acquire data or perform a task considered too dangerous for a person to attempt, e.g. bomb disposal. Aerial search and rescue robots, which are more commonly known as Unmanned Aerial Vehicles (UAV), have also been extensively used by the Military for reconnaissance. With the ability to fly semi-autonomously and broadcast images to remote locations, aerial robots have been a major reconnaissance tool. Recently military UAV’s have been given the ability to deliver weapons on enemy targets, this has intensified debate about whether it is ethical for robots to be armed and able to deliver weapons [9].

A critical component of USAR robots is the communication network between the robot and the control station. The communication network is critical in allowing the operator to control the robot, as well as providing the robot with the ability to send back acquired data in real time. This data may consist of visual images as well as air quality, temperature and radiation levels. Communication between USAR robots and a control station can be classified as either tethered or wireless. Tethered communications links offer a stable communication link but in search and rescue situations the cable can restrict movement and can become entangled in the rubble within a disaster zone. Wireless communications does allow greater movement and flexibility to the USAR in a disaster zone but wireless communications can be unreliable especially when the USAR robot is operating in confined spaces such as collapsed building.

The next section of this chapter takes the form of a literature review and will discuss the current status of USAR robots and outline some of the challenges faced by USAR robots when operating in a disaster zone. The types of USAR robots currently in use as well as the possible sensors that may be fitted to USAR robots will be discussed.
Methods of communicating with USAR robots will then be explored before the issues arising from the interaction between the operator and USAR robot is discussed.

The final section of this chapter will present the designs of two USAR robots that were created as part of this research. The first USAR robot design to be presented is a quadruped robot. It is possible for a quadruped robot to emulate the movements of a canine, such as a sniffer dog, and move carefully and quickly through the disaster zone with minimal impact on the surrounding environment. The second USAR robot design presented is a tank based robot. The tank based robot has good maneuverability over rough terrain but can disturb the surrounding area whilst moving through a disaster zone. The tank based robot though is capable of carrying a large payload allowing additional sensors to be incorporated within the robot.

It is envisaged that these two USAR robots will be capable of operating as a team. After the initial reconnaissance by the quadruped robot the tank based robot could then follow the quadruped robot. As the quadruped robot is able to nimbly move through a search area it would have less impact on the surrounding area reducing the chance that an USAR robots movements could trigger further structural disturbances. When an area had been initially searched by the quadruped robot the tank based robot could then move into the search area and use its additional sensors to complete a thorough search of the area.
2.2 Technical Overview of Urban Search and Rescue Robots

The Center for Robot-Assisted Search and Rescue (CRASAR), based at Texas A&M University, were one of the four teams that attended the World Trade Centre collapse [1] and have now deployed USAR robots at 15 of the 26 documented disasters since 2001. Ten of their USAR robot deployments are discussed in [2]. During these deployments the USAR robots were used for:

- searching for survivors,
- reconnaissance and mapping, and
- inspection of building and bridges.

Paper [2] also outlines CRASAR's observations from their experience attending varying disaster zones, these are:

- ground based USAR robots used in disaster zones are almost always tethered for the following two reasons:
  1. entry into a disaster zone is generally from above, and
  2. wireless networks are unreliable in dense rubble.
- the major problems encountered in a disaster zone has been sensing and not mobility which was previously considered to be the major problem USAR robots could face. “Dirt, mud, water, smoke often interfere with and coat sensors” [2].
- Unmanned Aerial Vehicles (UAVs) must operate within 3m of a structure in order to get reasonable images that can be used by structural engineers.

The collapse of the Municipal Archive College of Cologne City, Germany, in 2009 was also attended by CRASAR and is discussed in [10]. In this disaster the rescue team attempted to deploy two different styles of USAR robots to search for possible survivors and then search for specific documents spread within the buildings ruins. Neither of these deployments were successful because of the nature of the building collapse. Entry points to the collapsed building where the smaller USAR robot could have been
deployed were unstable and deemed unsafe for the rescuers to access whilst the larger USAR robot was unable to maneuver through the internal rubble because of the number of metallic book cases that had collapsed leaving very small operating voids. The rescue team concluded that a marsupial style deployment could have been used successfully here. A marsupial deployment would have involved maneuvering the larger USAR robot over the unstable terrain and deploying the smaller USAR robot in the area that was deemed unsafe for the rescuers themselves.

A need to evaluate and compare the performance of USAR robots prompted the U.S. Department of Commerce’s National Institute of Standards and Technology (NIST) to investigate how to measure the performance of USAR robots. This program was also co-funded by US Governments Federal Emergency Management Agency (FEMA) and the Department of Homeland Security (DHS). The main aim of this investigation was to determine common metrics that could be used to evaluate USAR robots, these common metrics included mobility, sensing capabilities, durability, communications, power, logistics, integration, human-system interaction and terminology [11]. NIST has thus developed some performance standards for USAR robots via the E54 Task Group. Standard test arenas for USAR robots were also specified [12, 13] with additional work being performed to scale these test arenas to a size that could be accommodated in standard laboratories [14].

Figure 2.1 shows an overview of the functionality that an USAR robot may have. This figure breaks these functionalities down into five categories:

1. Types: different mechanical structures,
2. Sensors: methods with which the USAR robot senses it’s environment,
3. Processing: algorithms used within the USAR robots computing system,
4. Communications: methods by which the USAR robot communicates with a control station, and
5. User Interface: ways in which the operators control an USAR robot and perceives the data obtained by the USAR robots sensors.
Figure 2.1 Overview of USAR robot functionality

Each category shown within figure 2.1 is a field of research in its own rite and consists of many sub-categories, many of which are optional and depend entirely on the environment in which the USAR robot is to operate and the task the USAR robot is to perform.
2.2.1 USAR Robot Types

When referring to the type of USAR robot, I am referring to the USAR robot's mechanical structure, which in turn dictates the USAR robot's method of propulsion. In figure 2.1 only four types of USAR robots were mentioned, but USAR robots come in many different physical configurations. The environment in which the USAR robot is to operate will greatly influence the physical structure and propulsion system used. Each type of USAR robot has their niche, but USAR robots can be loosely categorised as either airborne, marine or ground-based.

In pre-USAR robot times, specially trained sniffer dogs proved effective when they were used in disaster zones. Sniffer dogs were able to locate victims buried under rubble as they have an exceptional sense of small, exceptional hearing and are physically agile and nimble on their feet. The Canine Augmentation Technology (CAT) system detailed in [15] takes advantage of the dog's Biological Intelligence (BI). The CAT system allows video and audio to be streamed from a camera and microphone attached to the dog whilst the dog moves through the disaster area under the remote guidance of its trainer. This system allows the dog to locate victims within a disaster zone whilst the dog's handler remains in a safe location.

One of the problems with the CAT system was the images seen by the controller were effectively what had been seen by the rescue dog. When the dog was searching and sniffing through the rubble the controller would only see the ground, whilst when the dog had discovered somebody the dog would look up and the controller would only see the sky and not the victim. To address this issue the Canine Assisted Robot Deployment (CARD) was developed [16]. This system strapped a small wheeled robot to the chest of the rescue dog in such a way that it didn’t interfere with the dog's movements. When the dog located a victim a small wheel robot was deployed from a vest-like device which the dog was wearing. The deployment of a small wheel USAR robot would then allow the controller to see images from the disaster area that they wanted to and not where the dog was looking. This style of USAR robot deployment is known as a marsupial operation and takes advantage of the fact that different robots have different strengths and weaknesses. In this instance the rescue dogs ability to quickly maneuver over the
rough terrain and find victims via its sense of smell is complemented by the smaller wheeled USAR robots stable, controllable operations when the victims have been found.

The deployment of the small wheeled USAR robot was triggered by the controller via a wireless communication link. Thus this system had the same wireless communications constraints as other USAR robot systems, the rescue dog needed to stay within wireless coverage of the controller to ensure that the deployment of the USAR robot could be triggered. It was proposed in [16] that the rescue dog could drop small wireless devices and use an ad-hoc MESH wireless network to ensure that the wireless connection between the rescue dog and the controller was always intact. But this was deemed impractical as it would mean the rescue dog had to carry more equipment whilst searching for survivors which may hinder the rescue dog’s progress.

The systems defined in [15] and [16] are designed to ensure the rescuer remains in a safe location whilst the rescue dog is placed in the disaster zone. Placing a highly trained rescue dog’s life in danger in this manner could be seen as unethical by some and a necessity by others. Legged robots may provide an alternative to rescue dogs but the Biological Intelligence of the rescue dog would be extremely hard to replicate.

Boston Dynamics have developed a range of legged robots that could be adapted to be used as an USAR robot. RHex [17] is a simple hexapod robot with a unique semi-circular leg design that uses only one actuator per leg. RHex is capable of operating in rock fields, mud, sand and up slopes and stairs. BigDog [18] is a large quadruped robot which is 90cm long, stands 75cm tall and weighs 109kg. BigDog’s control system keeps it balanced in all terrain and all weather conditions and it has the ability to walk over rough terrain as would be encountered in a disaster area, as shown in figure 2.2.

The US Defense Advanced Research Projects Agency (DARPA) announced DARPA Robotic Challenge (DRC) in April 2012. The DRC primary goal is “to develop ground robots capable of executing complex tasks in dangerous, degraded, human-engineered environments”. The humanoid robots to be used in this challenge are the Atlas humanoids [19] from Boston Dynamics. Atlas is a human-sized humanoid which is
highly mobile and has been designed to negotiate rough terrain. Humanoids participating in the DRC must be capable of “performing complex mobility and manipulation tasks such as walking over rubble and operating power tools”. When operational the Atlas humanoid will be able to act as an USAR robot within a rough terrain disaster area.

Figure 2.2 Boston Dynamic’s BigDog Quadruped Robot

Wheeled robots are also used as USAR robots but are more suited to predominately flat surfaces. An example a 4WD USAR robot is the Explorer from Coroware, shown in figure 2.3. This Explorer has an articulated base that allows it to move over moderately rough terrain. The onboard PC has a 802.11 b/g/n interface for wireless communications. A pan-tilt camera and a Laser Range Finder (LRF) are also available to aid navigation through the disaster zone. Another unique wheeled USAR robot is the SandFlea [20] from Boston Dynamics. This small wheeled robot has mechanical actuators that enable it to jump up to 10m allowing it to jump walls, onto roofs of houses and through windows.

Tank based robots overcome some of the mobility issues encountered by wheeled robots when traversing rough terrain making them more suitable to USAR style operations. The addition of ‘swing arms’ at the front and/or rear of a tank based robot allows a robot to change its shape to accommodate large obstacles. This style of design is referred to as “variable geometry”.

Paper [21] details a tank based USAR robot with swing arms mounted at the front and rear, this is shown in figure 2.4. The series of images in figure 2.4 show how the swing arms can be manipulated to allow the robot to climb over a large obstacle which in this case is a large step. The swing arms have motorised tracks attached, for this reason they are sometimes referred to as ‘sub-tracks’. Swing arms can generally move through 360° and [21] presents a detailed description of the kinematics required to manipulate a USAR robot over rough terrain by monitoring the Center of Gravity (CG) of the robot.

There has been much research in the area of tank-based USAR design. [22] presents two tank based USAR robots, with one being a pure tank based robot whilst the second has a swing arm connected to the front. Different variations on tank based robots with front and rear swing arms are presented [23-26] whilst [27] adds a motorised pole to the tip of each swing arm to gain additional maneuverability. Tank based USAR robots are now commercially available from companies such as Yujin who offer the RoboHaz DT-3 [28] and iRobot Corporation who manufacture four tank based robots with their PackBot [29] being the most widely used. The PackBot was recently deployed at the Fukushima Nuclear facility [4] in Japan as part of the reconnaissance to determine the damage and radiation levels within the buildings at this facility.
Aerial USAR robots are more commonly known as Unmanned Aerial Vehicles (UAVs). The first documented case of a UAV being deployed as an USAR robot was by CRASAR in 2005 where a miniature helicopter was deployed along the Mississippi coast in the US to look for survivors after Hurricane Katrina [2]. In addition to looking for survivors within disaster zones UAVs are also used to assist structural engineers perform remote structural forensic inspection of damaged buildings and bridges. UAVs used as USAR robots can be either remotely controlled by an operator or autonomous. Most UAVs that are controlled by an operator do operate as a semi-autonomous robot as they use onboard sensor data to control movement and stability whilst taking supervisory movement commands, e.g. move forward, from the operator.

There are many different types of UAVs that can be used in an USAR operation and the selection of which UAV should be used depends greatly on the environment in which the UAV will be operating. [30] discusses the use of two styles of UAVs to support a ground based USAR robot. The first UAV discussed is a miniature blimp. Miniature Blimps have the advantage of being intrinsically stable with low noise, low vibration and the ability to hover. But miniature blimps are very susceptible to wind gusts which
can make them unstable and therefore unusable in breezy locations. The second UAV used in [30] is a Quadracopter. Quadracopters are more resistant to wind gusts and this is their primary advantage over a miniature blimp. Quadracopters are highly maneuverable and are capable of carrying a camera as part of their payload.

Fixed wing aircraft can also be used as USAR robots. The UAV Outback Challenge (UAVOC) [31] is an event that has been created to advance the research into autonomous aerial USAR robots. In the UAVOC autonomous fixed wing aircraft must search a remote location for a survivor, when the survivor has been located the GPS coordinates of the survivor must be recorded and a survival pack dropped near the survivor. Fixed wing UAVs have also been extensively used by the Military over the past decade. With the ability to fly semi-autonomously and broadcast images to remote locations, fixed wing UAVs robots have been a major reconnaissance tool for the military and law enforcement. General Atomics Predator [32] drone is probably the best known military UAV, it has the ability to fly up to 25,000 feet and remain airborne for 40 hours. The Predator drone is controlled wirelessly via a satellite link allowing the operators to be located in a remote secure location.

In 2005 a maritime USAR robot, the Scorpio 45, was used to help free Russian sailors trapped in an AS-28 submarine in the Pacific Ocean. In this particular deployment the Scorpio 45 robot was able to free an entangled Russian submarine by diving to a depth of 200m and severing the entangling cables. CRASAR deployed two maritime USAR robots to investigate structural damage to the Rollover Pass Bridge in Texas USA after Hurricane Ike [2] in 2008. One of these USAR robots was an unmanned marine surface robot whilst the other was an underwater robot. The use of maritime USAR robots in these types of deployments eliminates the need for scuba divers to perform these dangerous tasks.

### 2.2.2 USAR Robot Sensors

The methods with which an USAR robot senses its environment are critical to the USAR robots operation. Sensors can be installed on USAR robots to either aid the
USAR robots mobility or allow the USAR robot to gather information about the environment in which it is operating. The sensors installed on an USAR robot are determined by the environment in which the USAR is operating as well as the task the USAR robot has been asked to perform.

Sensors can be mounted on an USAR robot to provide information to assist mobility by improving obstacle avoidance. Ultrasonic Sensors and Distance Measuring Sensors (DMS) are non-contact methods that can be used to detect objects whilst simple limit switches can be used to detect collisions. Tilt switches and pressure sensors can also be used to verify that the USAR robot remains in contact with the ground.

Localisation within a disaster area is critical for USAR robots. Localisation can be assisted by the use of a Global positioning system (GPS) to provide a two-dimensional (2D) coordinate reading. Airborne USAR robots can use altimeters whilst maritime USAR robots can use a depth gauge to provide data to compliment the GPSs reading and create accurate three-dimensional (3D) coordinates. In cases where ground based USAR robots cannot receive GPS signals the information gained from a 9-axis inertial measurement unit (IMU), consisting of a 3-axis gyroscope, 3-axis accelerometer and 3 axis magnetometer, and an Encoder can be integrated to determine the USAR robots location.

Vision sensing is another method that can be used by the USAR robot for object avoidance and localisation. Data can be collected for vision processing by conventional cameras, thermal cameras or stereo vision cameras such as the Bumblebee [33] from Point Grey. Laser Range Finders (LRF) can provide 3D data to an image processing system. Hokuyo manufacture a range of LRFs that can detect objects up to 30m from the USAR robot, the UTM-30LX [34] has a range of 30m with an accuracy of 10mm to 30mm. Airborne USAR robots can utilise radar whilst maritime USAR robots can utilise acoustics or sonar in a similar manner to detect objects.

As well as the sensors that aid the USAR robots movements within a disaster zone sensors can be mounted on a USAR robot to provide information about the environment. Temperature and humidity sensors may be installed. Sensors that can
determine radiation levels and gas sensors to test the levels of oxygen, hydrogen sulfide, methane and/or carbon dioxide allow the USAR robot operator to determine if the disaster area is safe for rescuers to enter.

2.2.3 USAR Robot Processing

Processing data acquired by the USAR robot is one of the major areas of research for USAR robots. The algorithms implemented are again dependant on the operational environment, the task that needs to be addressed and the number of USAR robots operating in the disaster zone. Algorithms can range from kinematic mobility issues and path planning to image processing and perception of the USAR robots operational environment.

USAR robots can be controlled by an operator, generally referred to as teleoperation, or they can be autonomous and determine their own path through a disaster zone. USAR robots can also be semi-autonomous where the USAR robot can receive supervisory commands from an operator, e.g. move forward, and the USAR robot then determines the best method to do so using motor control and collision avoidance algorithms. Some USAR robots exhibit adjustable autonomy. [35] details a Hierarchical Reinforcement Learning (HRL) algorithm where the USAR robot learns from the environment and the tasks it is asked to perform. Using HRL the USAR robot is then able to determine which tasks it is capable of performing and which tasks require an operator’s intervention. The motivation behind a HRL style algorithm is that operators can find the USAR operation stressful and alleviating pressure from the operator when possible allows the operator to perform at their optimum level when needed.

Victim identification remains one of the tasks that a semi-autonomous or autonomous USAR robot should be capable of doing. Currently victim identification is performed by the USAR robot operator from vision obtained from either a standard camera or thermal camera. [36] presents a technique that may allow victim identification to be performed using an autonomous USAR robot. Hyperspectral Imaging is based on the use of waves in the Near InfraRed (NIR) spectrum. If an object is irradiated in NIR waves then the
Simultaneous Localisation and Mapping (SLAM) is an algorithm or technique used to create a map of an unknown environment. This technique has been widely used in robots since the early 1990’s [37]. SLAM is an iterative process that feeds visual map data to provide localisation and then uses localisation data to reinforce the map. [30] discusses the use of SLAM to estimate the orientation of a robot within a disaster zone. In this particular example data is acquired from two robots. Stereo visual data and IMU data is acquired from a UAV while odometry data is provided by a ground based USAR robot. This data is then used in conjunction with an Extended Kalman Filter (EKF) to provide a 3D map of the disaster zone and show the ground based USAR robots location and orientation within this map.

In an ideal fully autonomous USAR scenario aerial USAR robots would access the disaster area from the air, they would then command large ground based USAR robots to remove rubble before commanding smaller rubble penetrating ground based USAR to search through the debris for survivors. This scale of autonomy is a long time off being realised, but USAR robots can act together to form USAR robot teams where the strengths of one USAR robot complement the other USAR robots within the team. Paper [30] discusses a multi-USAR robot situation where an aerial USAR robot is gaining image data for control purposes, then using teleoperation to control a ground based USAR robot. The algorithms developed within USAR robots for these multi-USAR robot scenarios must allow for the sharing of acquired data and the ability to allocate or share tasks to meet a common goal.

One form of multi-USAR operation is known as a Swarm. In a USAR context swarms are physically similar USAR robots that co-operate together within the disaster zone. The concept of swarm robotics stems from the study of social insects which exhibit a phenomenon known as swarm intelligence. The rationale being that swarms of a large number of simple robots can communicate and cooperate in such a manner that they can
perform the same tasks as a larger robot. There has been a lot of research investigating how USAR robot swarms could cooperate, [38] discusses the issues of how a swarm of USAR robots can enter a disaster zone autonomously via a single entry point. [39-41] discuss how a swarm of USAR robots can cooperate so that they create an ad-hoc MESH wireless network such that all USAR robots have a communication path to an operator.

### 2.2.4 USAR Robot Communication

Communications between USAR robots and a control station are vital for the success of an USAR deployment. There are two forms of communication possible, the first is having the USAR robot tethered via a cable to the control station and the second is wireless communications. There are advantages and disadvantages to both forms of communications though.

Tethered USAR robots have the advantage of having a constant physical communication channel between the USAR and the control station. This physical communication channel is a multi-core cable which provides power, communications and possibly video and audio information. [21] details a tank based USAR robot that operates with a 400m cable interconnecting the USAR robot and the operator. The main disadvantage with a tethered communication link is that it hampers the maneuverability of the USAR robot within the disaster zone. The cable limits the distance that the USAR robot can travel in the disaster zone and the cable becoming entangled in debris is also another major concern. Secondary issues resulting from the use of a tethered USAR robot are more practical, the voltage drop across a 400m cable can be significant. The only way to remedy this voltage drop is to provide larger conductors, which in turn makes the cable bulkier and increases the risk of the cable becoming entangled. Also, if the USAR robot is to operate in the mining industry then Intrinsic Safety (IS) issues must be considered, which could reduce the amount of power available to the USAR robot and its sensors.
Wireless communication similarly has its advantages and disadvantages. Wireless communications allows the USAR robot to move freely in the disaster area. The major disadvantage with wireless communications though is the debris filled environment in which the USAR robot is operating. Collapsed buildings, especially those constructed with reinforced concrete, can severely attenuate the wireless signals between the operator and the USAR robot. [1] documents that robots were actually lost in the ruins of the World Trade Centre in 2001 due to a loss of wireless communications. In [2] the author also claims that ground based USAR robots are almost always tethered because “wireless networks are unreliable in dense rubble with metal rebar, mines, or inside commercial structures such as nuclear power plants”.

Given the challenges faced by wireless connections in USAR deployments the overwhelming advantages they possess has seen significant research in the field. [42] discusses a project where a ground based USAR robot was controlled via a Satellite communication link. Satellite communication can only be performed when the ground based robot is outdoors, so to overcome this problem this project had the satellite communications connected to a LAN and then through to an ad-hoc MESH network to which the USAR robot was wirelessly connected. The satellite communications link presented a narrow bandwidth of 768kbps so a Digital Evaluation Map (DEM) was used to represent the data obtained from the 3D Laser Range Finder.

References [39-41] use teams of USAR robots to ensure that a constant wireless connection is available between the USAR robot and the operator. [39] presents a solution where autonomous USAR robots adopt one of two roles, they are either an Explorer or a Relay. Explorers are USAR robots that are meant to explore at the extremities of the wireless coverage whilst the Relays are purely used to transfer any data gathered by the Explorer towards the control station. As the Explorer and Relay robots move through the disaster area they communicate wirelessly and agree to a position where they can meet and exchange information. After the exchange of data the robots then agree on a new rendezvous location and the Explorer moves further into the disaster area to gather more information whilst the Relay will move towards the control station to exchange data from the Explorer with either another Relay or the control center itself. Using this procedure the Explorer will move deeper into the disaster area.
with each exchange of data and maintains a virtual wireless connection with the control station.

References [40, 41] presents a more dynamic approach to that presented in [39] as the USAR robots are designated as Search Robots (SR) or Relay Robots (RR) dynamically. The USAR robot that finds itself at the end of a communication link, i.e. greatest number of ‘hops’ from the control station, becomes the SR and other USAR robots act as RR. [40, 41] also differs from [39] because the USAR robots always remain connected to their ad-hoc MESH network. SRs and RRs monitor their Received Signal Strength Indicator (RSSI) from adjacent robots and move in such a way that the RSSI between robots never drops below set thresholds. Should the RSSI fall below the threshold the network will reconfigure and the reallocation procedure to determine SR and RR will occur.

2.2.5 USAR User Interfaces

Two critical factors that can affect the operator in teleoperated USAR robots deployment are the methods used by the operator to control the USAR robot and the way in which the data obtained by the USAR robot is perceived by the operator. Human Robot Interaction (HRI) is thus another field of USAR robotics that has been heavily researched.

One of the reasons that operators find USAR situations stressful is the lack of perception they have of the environment in which the USAR robot is operating. The way in which data is acquired and then presented to an operator can improve their situation awareness. Situation awareness is defined in [43] as “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future”. [30] presents a method aimed at improving the operators situational awareness by using a ground based USAR robot and an aerial USAR robot. In this proposal images are obtained from both USAR and merged to form a 3D representation of the environment in which the USAR is operating. This image is then displayed in the operators Head Mounted Display
(HMD). The HMD has a 3-DOF tracking device which allows the operator to then issue commands to the ground based robot via head movements.

When the operator is using a narrow bandwidth communication link, as in [42], it was found that issuing coordinate based guidance commands gave much better results than issuing commands that changed operational parameters, e.g. changing motor speeds. If the coordinates are issued frequently enough then new coordinates will be issued before the USAR robot has reached the current coordinates, thus resulting in continuous movement of the USAR robot. This system has the advantage that if the wireless communications is momentarily lost or there is a delay in the image being presented to the operator then the USAR robot will move to the next waypoint and stop, as opposed to the operational parameter method which would result in the USAR robot travelling in the current direction until new operational parameters were received. Therefore the use of coordinate based commands reduces the number of collisions within the disaster zone.

Research into user-friendly remote controllers for USAR robots is presented in [44]. Here tests were done with operators to see if a joystick-based controller or a touch screen Tablet PC achieved better guidance performance. In this series of tests, video was streamed to a display and the operator used only this information to guide an USAR robot through a set of obstacles. Results based on the time taken to traverse the obstacle courses and the number of collision resulted in the joystick-based controller recording significantly better results. When surveyed 78% of the operators rated the joystick controller as ‘moderate’ or greater whilst only 53% of the operators rated the Tablet PC control as ‘moderate’ or greater.
2.3 Urban Search and Rescue Robot Designs

This research involved the design of two Urban Search and Rescue (USAR) robots. A quadruped was designed to emulate the speed and maneuverability of canine rescuers. This design was based around the biomimetic features of a bear [45, 46] which created a robot capable of operating as a quadruped with the ability to stand on its hind legs. The ability to stand can increase the robot’s field of view and thus its perception of the environment in which it is operating. A tank based robot with the ability to deploy wireless Drop Nodes (DN) was also designed. The tank based robot was also fitted with a pan-tilt camera on a 3 Degree of Freedom (DOF) robotic arm, allowing an operator to obtain a 360° view from within the disaster zone.

These two robots, whilst differing in their mechanical structure, complement each other’s functionality. A quadruped is able to move nimbly and quickly over rough terrain, such as rubble, with minimal disturbance to the surrounding area. This makes a quadruped robot ideal for performing an initial search where it is a necessity not to disturb the surrounding area in a way that may cause further damage, i.e. create further structural collapses. The tank based robot also has the ability to move quickly over rough terrain but its tracks can cause collateral damage to the surrounding environment when moving. When moving it is also possible for the tank tracks to slip, this slippage can cause non-planned sideways movement and/or spray debris around the disaster zone. The advantage of the tank based robot though is its ability to carry a large payload, meaning it is capable of carrying additional mechanical apparatus, such as the Drop Node Deployment System (DNDS) which is detailed in section 2.3.2.1.

These two USAR robots which were designed as part of this research can either work individually or as a team. When working as a team the quadruped can perform the initial reconnaissance for survivors and/or victims within an area of the disaster zone before the tank robot is permitted to enter that area. This allows an area to be inspected with minimal structural impact. Once the area is deemed safe then the tank robot could follow the quadruped. In this scenario the tank based robot would be responsible for maintaining the wireless communication channel to the Control Station (CS) by deploying DNs when the RSSI of wireless connection between the tank robots Mobile
Node (MN) and its parent node faded below a predefined value. A complete description of the wireless communication system implemented in this research and the algorithms used to ensure the wireless communication network remains functional are contained in sections 3 and 4 of this thesis.

Both the quadruped robot and the tank based robot could also act as individual USAR robots. The main disadvantage for the quadruped when operating independently is the depth that the quadruped can penetrate the disaster area is limited to the coverage of a single hop of the wireless communication system. Operating in this manner may result in the CS losing communication with the quadruped which could result in the robot being lost within the disaster area. The tank based robot could operate individually with a greater penetration depth within a disaster area as it is the robot responsible for keeping the wireless communication channel operational by deploying DNs. But having the tank based robot operate individually would mean that the initial reconnaissance for survivors and victims’ would be performed by a robot that may create enough vibrations to see conditions within the disaster area deteriorate further endangering any survivors.

The following sub-sections of this thesis will present the complete designs of both the quadruped and tank based USAR robots. As part of the design process a set of design criteria for USAR robots have been developed. These design criteria have been included in both the designs that have been completed as part of this research and are defined as follows:

1. As a minimum the robot should be capable of some simple semi-autonomous operation, to allow for intermittent communications between the robot and the operator.
2. The robot should be designed such that the operational time within the disaster zone is maximised. The key factors here are the mechanical structures weight and the energy source. The chosen energy source must be capable of moving the robot within the disaster zone for a reasonable amount of time before the energy source needs to be replenished.
3. The robot should be mobile enough to maneuver quickly and efficiently through all anticipated terrains, but at the same time be robust enough to withstand the conditions within the disaster zone.

4. The robot should have a reliable and energy efficient way of communicating data back to an operator.

5. The robot should have sufficient on-board sensors to allow it to:
   (i) traverse the terrain with minimal impact on the environment,
   (ii) determine an accurate position within the disaster zone, and
   (iii) be capable of relaying environmental conditions, such as oxygen levels and/or radiation levels to a control station.

The design of both the quadruped and tank based USAR robot is now presented in the following sub-sections.

### 2.3.1 USAR Quadruped Robot Platform

In January 2006 Sony Corporation announced that it was ceasing production of their robotic pet the AIBO. The AIBO had been the basis of RoboCup’s Four-Legged League (4LL) since 1999. This prompted the RoboCup Federation to release a “Call for Tenders: A Standard Robot Platform for Robot Soccer” which was issued in November 2006. The University of Newcastle’s NUBots [47] had been proactive in the search for a standard robot platform to enable RoboCup’s 4LL to continue and presented a proposal [48, 49] in June 2006. Figure 2.5 shows an early conceptual design for a quadruped robot that could replace the AIBO as the robotic platform in the 4LL. This quadruped was based on the same physical attributes as the AIBO as well as having the biomimetic characteristics of a terrier-style dog.

Initial work on the quadruped robot design lead to the concept of a robot based around a bear instead of a terrier-style dog. The advantages of designing a quadruped around the biomimetic features of a bear are:
1. A bear is a physically big, strong and bulky animal. Thus basing the structure of a quadruped around a bear would allow more room to house a complex electronic system and sufficient batteries to allow the quadruped to be operational for a minimum of an hour.

2. A bear is capable of standing on its hind legs enabling the bear to obtain a higher viewing perspective and thus a greater perception of its surrounding environment.

3. A bear can perform both quadrupedal & bipedal motions.

Figure 2.5 Early conceptual design of a Quadruped Robot

The ability to use a robot bear for purposes other than RoboCup’s 4LL was another motivation that led to the development of a functional robot bear. One of these motivations was the possible use of a quadruped as an USAR robot. Canines have demonstrated their ability to maneuver quickly and nimbly through a disaster zone, so the concept of a quadruped robot to emulate this maneuverability has considerable merit. Thus a quadruped USAR robot based on the biomimetic features of a bear was designed and named HyKim [45, 46].

The implementation of a quadruped robot as an USAR is based on both the overview of USAR robot functionality presented previously in figure 2.1 and the design criteria
presented in the introduction to section 2.3. Figure 2.6 shows the functionality that the USAR quadruped robot that I have designed possess.

In this design the type or structure is a legged. The legged base will give the USAR robot good maneuverability over rough terrain and the agilely required to maneuver without disturbing the surrounding environment, making it ideal for an USAR robot performing the initial reconnaissance of a disaster area. The legged structure is based around Dynamixel modules, which are in essence serially controlled servo motors, with the necessary custom designed aluminum frames to interconnect the motors as well as safely house the electronic systems.
The USAR quadruped robot has a custom designed energy efficient processing system. The processing system consists of three major electronic modules, a main processing system and two modules that interface to the USAR quadruped robots sensors. The onboard processing system will implement an algorithm that is predominantly manually controlled but may exhibit some semi-autonomous features, like obstacle avoidance. The communication between the USAR quadruped robot and the CS will be wireless and a complete discussion on the ad-hoc MESH wireless communication system is presented in sections 3 and 4 of this thesis. The teleoperation control at the CS will be performed by a PC or laptop, initially using the keyboard to issue movement commands but this may be extended to joy stick control once the system is operational.

2.3.1.1 Mechanical Design

The actuators that were chosen for the robot bear design were Dynamixel modules, a range of serially controlled servo motors manufactured by Robotis. The dimensions and specifications for the Dynamixel modules can be found in Appendix A. The mechanical design challenge was to create a robot constructed primarily from these Dynamixel modules whose shape resembled a real bear. The first step of my mechanical design was to examine the skeleton of a bear. Figure 2.7 shows the skeleton of a Polar Bear, Ursus Martimus. The bear skeleton was used as the basis of my design to ensure that the ratio of limbs, backbone, neck and head were all consistent with that of a real bear.

Figure 2.7 Skeleton of a Polar Bear, Ursus Martimus
Each of the robot bears limbs requires three DOF, one of these is provided by the ‘shoulder’ module so two DOF must be incorporated within each limb itself. This was achieved by mounting two Dynamixels ‘back-to-back’. This is shown in figure 2.8 in both the upper front and upper rear limbs. The size of these upper limbs was therefore fixed and as such dictated the minimum ratio that could exist when the bears skeleton was compared to the quadrupeds mechanical structure. It was decided to use the minimum ratio for the design, as one of the initial design goals was to create a quadruped that was approximately the same size as an AIBO. Therefore the length of two Dynamixels, 80mm, became the basis upon which all aluminum frames and body parts would be based and scaled accordingly. Table 2.1 details the dimensions of the resulting robot bear along with a comparison to the dimensions of the AIBO. Figure 2.9 shows these dimensions overlaid onto the completed design.

Figure 2.8 Structural design of robot bear based around Dynamixel modules

The actuators that were selected as the basis for the robot bear where the RX series Dynamixels from Robotis, the three different models within that series are the RX-10, RX-28 and the RX-64. These modules have holding torque characteristics of 10 kg.cm, 28 kg.cm and 64 kg.cm respectively. RX-64 modules were used in four locations where additional torque was required. These were to drive the back legs, at the hip-pivot point and at the base of the neck. The leg drive modules needed the extra torque to enable the robot bear to go from a quadrupedal standing position to a bipedal stand, whilst the hip-pivot module needed the extra torque to move the torso when operating in bipedal
mode. The neck module required extra torque to move the robot head which weight 1.5kg and was located 120cm from the robots torso. RX-28 modules were used in all other locations. In all twenty one Dynamixels were used within the robot bear, giving the robot bear 21 DOF. The DOF are shown in table 2.2.

<table>
<thead>
<tr>
<th>l</th>
<th>Description</th>
<th>HyKim</th>
<th>Sony Aibo</th>
<th>Size Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>l2</td>
<td>Neck</td>
<td>120</td>
<td>100</td>
<td>1.20</td>
</tr>
<tr>
<td>l3</td>
<td>Head Length</td>
<td>130</td>
<td>130</td>
<td>1.00</td>
</tr>
<tr>
<td>l4</td>
<td>Upper Front Leg</td>
<td>80</td>
<td>70</td>
<td>1.14</td>
</tr>
<tr>
<td>l5</td>
<td>Lower Front Leg</td>
<td>130</td>
<td>75</td>
<td>1.73</td>
</tr>
<tr>
<td>l6</td>
<td>TOTAL Front Leg</td>
<td>210</td>
<td>145</td>
<td>1.45</td>
</tr>
<tr>
<td>l7</td>
<td>Front Width</td>
<td>180</td>
<td>130</td>
<td>1.38</td>
</tr>
<tr>
<td>l8</td>
<td>Upper Back Leg</td>
<td>80</td>
<td>70</td>
<td>1.14</td>
</tr>
<tr>
<td>l9</td>
<td>Lower Bag Leg</td>
<td>90</td>
<td>75</td>
<td>1.20</td>
</tr>
<tr>
<td>l10</td>
<td>Ankle Back Leg</td>
<td>60</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>l11</td>
<td>TOTAL Back Leg</td>
<td>230</td>
<td>145</td>
<td>1.58</td>
</tr>
<tr>
<td>l12</td>
<td>Back Width</td>
<td>190</td>
<td>130</td>
<td>1.45</td>
</tr>
<tr>
<td>l13</td>
<td>Back Length</td>
<td>180</td>
<td>130</td>
<td>1.38</td>
</tr>
</tbody>
</table>

Table 2.1 Dimensions of the robot bear and comparison to the AIBO dimensions

Figure 2.9 Assembled and dimensioned robot bear
<table>
<thead>
<tr>
<th>Location</th>
<th>DOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front Left Leg</td>
<td>3</td>
</tr>
<tr>
<td>Front Right Leg</td>
<td>3</td>
</tr>
<tr>
<td>Rear Left Leg</td>
<td>3</td>
</tr>
<tr>
<td>Rear Left articulated Ankle</td>
<td>2</td>
</tr>
<tr>
<td>Rear Right Leg</td>
<td>3</td>
</tr>
<tr>
<td>Rear Right articulated Ankle</td>
<td>2</td>
</tr>
<tr>
<td>Horizontal Shoulder Roll</td>
<td>1</td>
</tr>
<tr>
<td>Horizontal Hip Pivot</td>
<td>1</td>
</tr>
<tr>
<td>Head</td>
<td>2</td>
</tr>
<tr>
<td>Neck-</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.2 Robot Bear

To complete the mechanical design ABS plastic body pieces were printed on a 3D printer, some of these pieces are shown in figure 2.10. The pieces designed provided one of two functions. Some pieces were designed to act as structural pieces of the robot bear whilst other pieces were designed purely to cover the electronic systems in an aesthetic way. Figure 2.10 shows the lower limbs of the front legs, these ABS plastic pieces were designed to act as part of the mechanical structure. Whilst the head pieces, shown on the assembled robot bear in figure 2.9, were designed to protect the circuitry that was located in the head.
Conventional quadrupeds, such as the AIBO, provide no DOF within the body to accurately emulate the movement of a quadruped. The motion of bears was closely examined to determine where extra DOF could be added to more accurately imitate their motions. Based on these observations two additional DOF were added. The first DOF was created to allow for a rolling movement of the front shoulders whilst the second DOF was added at the base of the backbone in the hip region to allow lateral movement to complement the shoulder roll. Figure 2.11 (a) shows the extra two DOF that were added that may be used to provide advanced quadrupedal motion. Figure 2.12 (b) shows how the Hip DOF can be used to provide a unique walking gait for the robot bear when acting as a biped.

![Quadruped Top View](image1)

(a) Quadruped mode

![Quadruped Front View](image2)

(b) Biped mode

Figure 2.11 Additional DOF that gave robot bear unique movements

Table 2.3. summarises the physical characteristic of the robot bear that has been designed to act as an USAR quadruped robot. This design should provide agile movement within a disaster zone, minimising the impact to the surrounding environment.
2.3.1.2 Processing System Hardware Design

The processing system for the USAR quadruped robot consists of a custom design energy efficient processing system. A modular, distributed processing system was achieved by having the following five modules:

1. HyInt: this is the core component of the processing system and is based around an energy efficient Geode microcontroller module. This module interfaces to the other modules with the processing system by either USB or serial communications.

2. HySense: this is an ATMega128 based system which is mounted on the torso of the USAR quadruped robot and uses an RS485 connection to communicate with the HyInt module. This module is used to gather the sensor data on the USAR quadruped robots body and transfer this to the HyInt via the RS485 link.

3. HySense Lite: similar to the HySense module but with reduced functionality, this is an ATMega128 based system and is mounted in the head of the USAR quadruped robot. This module is used to gather the sensor data from the USAR quadruped robots head and transfer this to the HyInt via the RS485 link.

4. Mobile Node: this is the interface module to the 802.15.4 based control network. This module communicates serially to the HyInt module.

5. Dynamixel network: this network consists of 21 Dynamixel modules which communicate to the HyInt module via a daisy chained RS485 network. Each Dynamixel has its own ATMega8 microcontroller and as such can perform a number of tasks that would otherwise be performed by the HyInt module.

### Table 2.3 Robot Bear Physical Characteristics

<table>
<thead>
<tr>
<th>Body</th>
<th>Degrees of Freedom (DOF): 21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height</td>
<td>0.59m</td>
</tr>
<tr>
<td>Weight</td>
<td>5.0kg</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Battery</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Lithium Polymer</td>
</tr>
<tr>
<td>Voltage</td>
<td>14.4V</td>
</tr>
<tr>
<td>Capacity</td>
<td>2400mAh</td>
</tr>
</tbody>
</table>
A block diagram of the processing is shown in figure 2.12. The three Dynamixels highlighted in red are the three Dynamixels in the USAR quadrupeds head and neck. In the default configuration, all Dynamixels are connected in a daisy-chain arrangement to the HyInt controller. An optional configuration, and the one highlighted by the red Dynamixels, is to have the head and neck Dynamixels controlled directly by the HySense Lite module which is mounted in the USAR quadrupeds head.

Figure 2.12 Block diagram of USAR Quadruped robot processing system

A 3-axis accelerometer and the 3-axis gyro form part of the HySense Lite module, thus having the HySense Lite control the neck and head angles allows a control system to be implemented that attempts to keep the head stationary whilst in motion. The camera for the USAR quadruped is located in the head, so the ability to prevent unnecessary movement of the head whilst in motion will provide a more stable image and thus ease
the burden on any image processing algorithms that may be implemented in HyInt. The concept of keeping the head as stable as possible whilst in motion is a feature displayed by the cat family, as a cat stalks its prey its head remains completely stable allowing the cats eyes to remain focused in its prey. This feature is not implemented in the USAR quadruped robot but is an extension of my work that will be investigated in future research.

**HyInt**

The HyInt processing system is the core processing system for the USAR quadruped robot. Figure 2.14 shows the HyInt processing system which consists of three boards:

1. Compulab CM iGLX,
2. Motherboard Interface (PB0801), and
3. Power Supply board (PB0803).

![Figure 2.13 HyInt processing system](image)

The CM-iGLX is a computer module supplied Compulab. To complement this computer module two additional boards were designed and manufactured, these were an intelligent power supply board and a motherboard interface upon which the CM-iGLX was mounted. The CM-iGLX module was selected as the basis of HyInt for two reasons:
1. The CM-iGLX is based around AMD’s Geode LX800. At the time of design the Geode LX800 microcontroller was one of the most energy efficient microcontrollers available based on comparisons of Million Instructions Per Second (MIPS) and Million Floating Point Operations Per Second (Mflops) versus device power consumption. The GeodeLX800 is capable of 990MIPS plus 270Mflops for a power consumption of 3~5W. If the mean power consumption is used for comparison purposes, i.e. 4W, the result is 248 MIPS/W and 68 Mflops/W. This gave superior performance compared to other available options, such as Intel’s XScale PXA255 ARM microcontroller. The XScale PXA255 specifications were 250MIPS at a power consumption of 1.6W, which is 156 MIPS/W and no floating point capability.

2. The CM-iGLX’s Geode LX800 is designed around a 32-bit X86 compatible Central Processing Unit (CPU). As the target operating system for the HyInt system (OS) was a linux based OS, having an X86 compatible CPU eliminated the additional step of cross compilation in the development phase. Cross compilation is a time consuming process and necessary if software is developed on an X86 compatible CPU and the target microcontroller is not an X86 compatible device, such as an ARM processor. Thus, selecting an X86 compatible device would reduce software development time.

The CM-iGLX’s Geode LX800 microcontroller required some form of cooling as the microcontroller reached temperatures of 90°C during testing before eventually shutting down. Cooling was provided by mounting the HyInt system in the body of the USAR quadruped as shown in figure 2.14. HyInt was mounted in such a way that the Geode LX800 was physically in contact with the main aluminum torso piece, thus using the body structure of the USAR quadruped robot as a heat sink. When mounted in this way the Geode LX800 reached temperatures in the range of 40°C to 50°C which were well below its specified maximum operating temperature of 85°C.
An intelligent power supply board was developed for the HyInt processing system, Appendix B contains the schematic for this board as well as the IO allocation of its ATMega8 microcontroller. The Intelligent Power supply board was designed to take the 14.4V 4 cell 7.2Ah LiPo battery and create 3 different power supply rails:

1. a 3.3V supply for the on board ATMega8 microcontroller,
2. a 3.3V supply for the CM-iGLX and other electronics within HyInt, and
3. a 5V supply for the CM-iGLX and other electronics within HyInt.

A separate 3.3V supply was created for the ATMega8 by using a LD1117V33, this is a simple 3 terminal low dropout regulator capable of delivering 1.3A. This was done so that the power supply controlling the microcontroller was independent of the power supplies being used within HyInt and therefore capable of turning HyInt’s power supplies off when necessary whilst remaining operational. The main 3.3V and 5V supplies used within the HyInt module were based on the VRAE-10E1A0 DC-DC converters from Bel Power, these highly efficient DC-DC converters are capable of delivering 10 A with an efficiency of 91%. The output voltage of these DC-DC converters was controlled by the value of resistance placed between the TRIM pin and 0V.
The intelligent power supply boards microcontroller monitors the power usage of the USAR quadruped robot. It is able to do so by monitoring the battery voltage and the current supplied by the 14.4V 2.4Ah LiPo battery. The battery voltage is sampled by a simple resistor voltage divider circuit which scales the input voltage down to a maximum of 3.3V. A 3.6V Zener diode is connected across the microcontrollers analog input to offer over voltage protection. Two current monitors are then used to determine the current flowing to the Dynamixel modules as well as the current being drawn by the remaining electronic systems within the USAR quadruped. The ACS713 Hall-Effect based linear current sensor from Allegro MicroSystems was chosen as the current sensor. This sensor allows current to pass through the device and produces an output voltage which is proportional to the current flow. Again a 3.6V Zener is placed across the microcontrollers analog input for over voltage protection.

Monitoring the power consumed by the UASR quadruped allows the intelligent power board to take action when it deems the power left within the batteries has been reduced to a level where future operation cannot be guaranteed. There are two ways that the microcontroller can determine when the LiPo battery is running out of charge. The first is to monitor the battery voltage. A characteristic of LiPo batteries is the output voltage is constant and then rapidly drops when the battery requires recharging. Monitoring the battery voltage and looking for the transition from constant to falling voltage is one method of determining a battery has reduced charge. The other method is known as Coulomb Counting and involves integrating the current usage over time. As the LiPo battery is rated at 2.4Ah the number of Coulombs which are potentially available is known, so by monitoring the current draw allows a prediction to be made as to when the battery may become inoperable.

Knowing when the battery is about to fail allows the intelligent power supply board to perform a controlled shutdown. Two FETs are used as switches that are able to control the current flow to the Dynamixel modules and the remaining electronics within the USAR quadruped. When the microcontroller deems that the LiPo battery is about to run out of charge the microcontroller will initially turn off power to the Dynamixel modules. This is done so that any remaining charge within the battery can be used by
the CM-iGLX to perform a controlled shutdown and avoid corruption of the operating system which is stored in the CM-iGLX’s NAND FLASH. An output from the ATMega8 microcontroller is connected to the CM-iGLX’s power healthy input, when the CM-iGLX sees this signal go HI it will immediately instigate a shutdown of the operating system in a controlled manner, saving any potential damage to the NAND FLASH memory.

A serial communication link is also used between the CM-iGLX and the ATMega8 microcontroller. This allows the current and voltage readings from the ATMega8 to be transferred to the CM-iGLX as well as giving the CM-iGLX the ability to turn power off the Dynamixel network should it ever need to do so. For greater flexibility additional functionality may be provided by the ATMega8 microcontroller unused GPIO. The unused GPIO are assigned the following features incase these may be needed for future projects.

1. Three on board pushbuttons connected to ATMega8’s PB3, PB4 and PB5,
2. Three dual position DIP switches connected to ATMega8’s PB6, PB7 and PD4,
3. Two on board LEDs connected to the ATMega’s PB1 and PB2, and
4. Four GPIO, that could be assigned as inputs or outputs, available at 10-pin IDC header.

Thus the intelligent power supply board provides the following features for the HyInt processing system:

1. Creates three separate power supplies,
2. Monitors battery input voltage and current levels,
3. Capable of determining the charge status of the LiPo battery,
4. Ability to turn power off to the Dynamixel modules as well as the electronic systems within the USAR quadruped,
5. Communicates serially with the CM-iGLX, and
6. Provides additional GPIO capabilities should these be required in future projects.
Both the CM-iGLX and the intelligent power supply board connect to the motherboard interface, the 3rd board that constitutes the HyInt processing system. This board provides additional functionality to the HyInt processing system and the schematics for this board can be found in Appendix B. The CM-iGLX provides only two USB ports. A four port USB hub integrated circuit was added to expand the available number of USB ports to five. The integrated circuit used was the USB2504 from SMSC, the only additional circuitry needed to support the four port USB hub integrated circuit was a 24MHz crystal. Thus the allocation of ports available on the motherboard interface were:

- USB1.0: RS485, two channels for Dynamixel modules,
- USB1.1: RS485, one channel for HySense and HySense Lite modules,
- USB1.2: Keyboard accessible from expansion board,
- USB1.3: Logitech web camera, and
- USB2.0: FLASH Drive with operating system.

It was important that the USB port used by the FLASH drive be connected directly to the CM-iGLX and not through the 4 port hub. This allowed the FLASH drive to contain the operating system as well as the file system and thus allowed the system to boot from the external FLASH drive. USB1.0 was connected to an FTDI FT2232L integrated circuit. This integrated circuit is a dual UART/FIFO but importantly has transmit enable (TXDEN) outputs. The TXDEN signals allow the MAX485 integrated circuits to implement the necessary two-wire half-duplex RS485 network needed by the Dynamixel modules. Similarly, USB1.1 was connected to an FTDI FT232RL integrated circuit which is a single channel UART/FIFO. The FT232RL was connected to another MAX485 to provide an RS485 channel that would be used to communicate with the HySense and HySense Lite modules. The final two USB ports, USB1.2 and USB1.3, were used to connect directly to a keyboard and the Logitech web camera. The Logitech web camera was mounted in the head of the USAR quadruped and provided vision capability to the robot.

Two serial ports are made available from the CM-iGLX on the motherboard interface. The first of these was used to connect to the intelligent power supply boards
microcontroller, whilst the second was a spare serial channel that could be used for debug purposes or uploading new BIOS images to the CM-iGLX.

An expansion board was also designed that connected to a 20-pin IDC ribbon cable header on the motherboard interface. The expansion board was a board specifically designed to plug into the USAR quadruped during software development. Through the connector on the motherboard interface the following connections were available on the expansion boards:

1. Ethernet port,
2. USB port which was used for a keyboard,
3. Serial port which was used for debugging and re-imaging the CM-iGLX’s BIOS,
4. VGA terminal.

**Hysense and HySense Lite**

All the USAR quadruped robots sensors are mounted on or connected to the HySense and the HySense Lite modules, these modules are shown in figure 2.15. The HySense module is mounted on the main torso piece of the USAR quadruped and acquires data from sensors located around the body. The HySense Lite module is mounted in the head of the USAR quadruped and as well as gathering data from sensors in the head region the HySense Lite module also controls the USAR quadruped’s eyes. The schematic of HySense and HySense Lite are presented in Appendix B.

The HylInt processing system communicates with the HySense and HySense Lite modules via a multi-drop RS485 network. In this communication network the HylInt module is the master, initiating all communication exchanges between itself and the HySense and HySense Lite modules. Command Packets are sent from HylInt. HySense and HySense Lite modules may then respond with a Return Packet. Figure 2.16 shows the packet structure for HylInt to HySense and HySense Lite communication packet whilst table 2.4 defines the function of each byte used within this packet structure. Both Command Packets and Return Packets share the same packet structure.
Byte 0 of a packet contains the Start of Header (SOH) ASCII character 0x02. In a Command Packet Byte 1 contains the ID of the destination. In this protocol:

- ID = 0x00 is a broadcast packet and as such is meant for all connected modules,
- ID = 0x01 is a packet for HySense, and
- ID = 0x02 is a packet for HySense Lite.

The ID byte in a Return Packet is the ID of the transmitting modules, allowing the HyInt to determine the source of the return packet. Byte 2 is the length of the packet and
in this protocol is equal to the number of data bytes $N$. Byte 3 is used to define the OpCode for the packet, this OpCode specifies what type of data is being sent or requested. Appendix C has a full listing of the available OpCodes defined as part of this protocol. There are many possible options specified by these OpCodes, ranging from simply requesting the reading from a Distance Measurement Sensor to selecting an image to be displayed on the OLEDs connected to the HySense Lite module.

The data or payload is then contained in bytes 4 to $N+4$. The number of data bytes within the payload is dependent upon the OpCode that is being performed. The CheckSum used within this packet structure is a 16-bit Cyclic Redundancy Check (CRC16). The HI byte of the CRC16 is placed in byte $N+5$ while the LO byte of the CRC16 is placed in byte $N+6$.

HySense and HySense Lite use an ATMega128 as their microcontroller. A bootloader has been written for this application and allows the HySense and HySense Lite modules to be reprogrammed over their serial port UART1. This feature allows the firmware to be updated in HySense and HySense Lite without the removal of any body pieces.

The RS485 communication is implemented on UART1 for both the HySense and HySense Lite modules using a MAX485 integrated circuit. An additional IO pin configured as an output is needed to enable the MAX485’s transmitter when the ATMega128 needs to transmit data. The algorithm used for the RS485 network is to disable the MAX485’s transmitter by default and only when data needs to be transmitted does the MAX485’s transmitter become enabled. In this way all connected RS485 devices on the network ‘listen’ for data and only when they have ‘heard’ information for them do they respond. This algorithm is easily implemented by setting the transmit enable (TXDEN) output at the start of the transmit Interrupt Service Routine (ISR) and when the transmit buffer is empty resetting the TXDEN output.

The ATMega128 has a maximum of eight analog inputs. The sensors used on the HySense and HySense Lite modules required eleven so a Texas Instruments ADS7844 integrated circuit was used to expand the number of analog inputs. The ADS7844 is a 12-bit eight channel serial output analog to digital converter. This was added to the
ATMega128’s Serial Peripheral Interface (SPI) bus. The SPI bus allows devices to pass data to the microcontroller in a standardised serial format. The SPI bus consists of four signals:

1. SCK, SPI bus serial clock.
2. MISO, SPI bus Master Input/Slave Output.
3. MOSI, Master Output/Slave Input.
4. SS, Slave Select Input.

Thus, the ATMega128’s SPI bus was used in conjunction with the ADS7844 to increase the number of possible analog inputs from eight to sixteen.

Both the HySense and HySense Lite modules contain a 6-axis Inertial Measurement Unit consisting of a 3-axis accelerometer and a 3-axis gyro. The 6-axis IMU is provided as feedback for the USAR quadruped robots balance and gait generation. The 3-axis accelerometer used is +/-3g ADXL330 from Analog Devices. The ADXL330 provides three analog outputs which are connected directly to the ATMega128’s analog inputs which has a voltage reference of 3.3V. The inputs from the ADXL330 sit at mid-voltage range of 1.5V when they are experiencing 0g. As force is detected by the accelerometer the voltage from the ADXL330 will increase or decrease depending on the direction of the force.

The 3-axis gyro is made up of two devices. An IDG-300 from InvenSense is used for the x and y axis whilst the ADXRS150 from Analog Devices is designed specifically for the z axis. The IDG-300 is a 3.3V device with has a sensitivity of +/-500°/s. The IDG-300 has three outputs, an output for both axes and a reference output. The gyro outputs sit mid-voltage range at 1.5V and are connected directly to the ATMega128’s analog inputs. The ADXRS150 z-axis gyro is a 5V device and the ADS7844 SPI bus 8-channel Analog to Digital converter has been configured to accept a maximum 5V input voltage. The ADXRS150 has a sensitivity of +/-150°/S and has an output of 2.5V when detecting no rate of change. There are two additional outputs, a 2.5V voltage reference that can be used to compare the variable analog input to and an analog input that indicates the devices temperature.
HySense and HySense Lite modules both use similar Distance Measuring Sensors (DMS). Both modules are capable of accepting inputs from two DMSs and use a FDG6342L integrated load switch to supply power to the DMSs. The load switches are used to conserve energy and turn the DMSs off when distance readings are not required and are controlled by two of the ATMega128’s digital outputs. The DMS that are connected to the HySense module is a Sharp GP2D120 which has a 3.1V to 0.3V voltage range, this corresponds to a range of 3cm to 40cm, but this relationship is non-linear. The GP2D120 requires a 5V supply but as the maximum output voltage from this device is 3.1V it is connected directly to the ATMega128’s analog input which has a voltage reference of 3.3V. The GP2D120 is mounted on the USAR quadrupeds chest such that when the robot is in a quadruped standing position the GP2D120 is pointed at the ground approximately 20cm in front of the robot. This GP2D120 sensor is provided to allow the USAR quadruped to determine it has come to the edge of a surface, e.g. the top of a flight of stairs, and prevent the robot from falling over this edge.

The DMS that is connected to the HySense Lite module is a Sharp GP2D12. This device has a 2.6V to 0.4V voltage range which corresponds in a non-linear relationship to a range of 8cm to 80cm. As with the GP2120, the GP2D12 requires a 5V supply but as the maximum output voltage from this device is 2.6V it is connected directly to the ATMega128’s analog input. The GP2D12 is mounted on the USAR quadrupeds head, at the end of the nose just above the camera. This DMS was included here so that the distance to objects detected by an image processing system could be verified.

In addition to the sensors previously mentioned the HySense module has a 3-axis Magnetic Sensor Module. The MicroMag3 from PNI was selected as the tilt-compensated digital compass. This module is based around PNI’s ASIC 3-axis Magneto-Inductive Sensor Driver and Controller, this device uses magneto-inductive sensors that change inductance with changes in the magnetic field parallel to the sensor. PNI’s Application Note [50] describes how magneto-inductive sensors mounted on 3-axes can be used as an accurate tilt-compensated digital compass. The ASIC has an SPI bus interface. HySense has two SPI devices so a 74LVC125 Quadruple Bus Buffer with
Tri-State Outputs is used to select which SPI device has access to the bus and thus a connection to the ATMega128 microcontroller.

The HySense Lite module has two features that aren’t available on the HySense module. These are the ability to control two Organic Light-Emitting Diode (OLED) displays (PB0701) and an interface to a board with three capacitive touch sensors with LED feedback (PB0702). Schematics to both PB0701 and PB0702 are available in Appendix B.

Two UG-2828GDEDFO1 128 by 128 colour OLED modules were used to emulate eyes for the USAR quadruped. This was done to allow the robot to display emotion, but it was also possible to display text as well as images on these displays. These displays were configured so they were controlled by the ATMega128’s SPI bus. This meant that three devices were connected to the SPI bus for the Hysense Lite module so additional digital outputs were required to ensure the correct devices were enabled when required.

Figure 2.17 shows the USAR quadrupeds head, the HySense Lite module is visible in this image as are the OLED displays. The camera can also be seen as can the DMS mounted above the camera.

The touch sensors were created using a Quantum Research QT240. Three touch pads were created on the printed circuit board (PCB). A number of resistors and capacitors
were used to select the sensitivity of the touch pad so that the change in capacitance of the touch pad could be detected through the ABS plastic head pieces. The QT240 provided digital outputs to indicate that a touch sensor had been activated, these digital outputs were connected directly the HySense Lite’s ATMega128. Feedback that a touch had been detected was provided by LEDs controlled by outputs from the ATMega128. For these LEDs to be activated the signals from the QT240 had to be detected by the ATMega128, thus the activation of the LEDs was proof that the inputs had been detected by the ATMega128.

**Mobile Node**

A Mobile Node (MN) is located within each of the USAR robots that have been designed as part of this thesis. Both these MNs use the same hardware design and the same communication protocol, the only major difference being the data allocation within the payload of the data packets. To make the description of our USAR robots standalone as complete descriptions there may be some information presented in this sub-section describing the MN which may be repeated in the tank based USAR robot design section.

The Mobile Node (MN) hardware design is presented and discussed in section 4.3.1 of this thesis. The MN mounted within the quadruped USAR robot is designated as MN2 and is configured as a router within the 802.15.4 based wireless guidance system whilst the Control Node (CN) is the coordinator within the network. The CN establishes a connection to the MN2 through the 802.15.4 network, section 4 of this thesis details the 802.15.4 based wireless robotic guidance system. Figure 2.18 shows the entire data flow from the Control Station to the HyInt module.
Figure 2.18 Overview of Packet Transfer between Control Node and Mobile Node

A standard packet was defined to enable software written in C to be reused in all core processing systems and all connected microcontrollers, this is shown in figure 2.19. Table 2.5 defines the function of each byte used within this packet structure.

<table>
<thead>
<tr>
<th>0xFF</th>
<th>0xFF</th>
<th>RSVD</th>
<th>LENGTH</th>
<th>RSVD</th>
<th>Data[0]</th>
<th>............</th>
<th>Data[N]</th>
<th>CHECK SUM</th>
</tr>
</thead>
</table>

Figure 2.19 Packet Structure for Mobile Node to FitPC2 Communications

*Byte 0* and *Byte 1* of the standard packets are headers and defined as 0xFF. The next byte, *Byte 2*, is allocated as an ID byte and is required to identify the packets destination. This is required when more than one device is connected to the communication bus, such as the multi-drop RS485 network used for the Dynamixel modules upon which the robot arm is based. When packets are only being exchanged two devices then *Byte 2* is not used.

Packets with variable lengths are accommodated for by this packet structure as *Byte 3* defines the length of the packet. The length of the packet is the number of data bytes \( N \) plus two, i.e. \( N+2 \). To allow packets to have different functionality *Byte 4* is provided to define the type of packet that has been transmitted or received. *Byte 5* to *Byte \( N+4 \) are allocated for \( N \) bytes of data, or that packets payload as it is commonly known. Finally
the last byte within the packet $Byte^{N+5}$ is the packet CheckSum. The CheckSum is calculated using the following equation:

$$CheckSum = \sim (ID + Length + Type + Data[0] + ... + Data[N-1])$$

where $\sim$ is the NOT operator.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Function</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 0</td>
<td>Header</td>
<td>0xFF</td>
</tr>
<tr>
<td>Byte 1</td>
<td>Header</td>
<td>0xFF</td>
</tr>
<tr>
<td>Byte 2</td>
<td>ID</td>
<td>None</td>
</tr>
<tr>
<td>Byte 3</td>
<td>Length</td>
<td>$N+2$</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Type</td>
<td>None</td>
</tr>
<tr>
<td>Byte 5</td>
<td>Data [0]</td>
<td>None</td>
</tr>
<tr>
<td>Byte $N+4$</td>
<td>Data [N-1]</td>
<td>None</td>
</tr>
<tr>
<td>Byte $N+5$</td>
<td>CheckSum</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2.5 Byte definition within Packet Structure

This is an efficient CheckSum as the CheckSum can be calculated as a packet is being received. The following algorithm can is used to implement this:

```c
// both Header Bytes have been received – process the Rxed packet  
PacketValid=FALSE;  
loop=TRUE;  
length=0;  
CheckSum=0;  
for (i=0;loop==TRUE;i++)  
{  
  CheckSum+=RxedByte;  
  if (i==1) length= RxedByte;  
  else  
  {  
    if (i>1) length--;  
    if (length==0) loop=FALSE;  
  }  
}  

CheckSum~=CheckSum;  
if (RxedByte) == CheckSum) PacketValid=TRUE;
```

There are two types of packets that are used within this system, Command Packets and Acknowledge Packets. Command Packets are issued by the Control Station (CS) and flow through the system to the HyInt module. These packets are used to issue commands to the USAR quadruped robot motion controller. Acknowledge Packets serve a dual purpose, they acknowledge that the Command Packet has been received as
well as transferring operational variables, such as sensor readings to the CS. Table 2.6 shows the data allocation within these packets.

<table>
<thead>
<tr>
<th>Command Packet</th>
<th>Acknowledge Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte</td>
<td>Function</td>
</tr>
<tr>
<td>Byte 0</td>
<td>Header</td>
</tr>
<tr>
<td>Byte 1</td>
<td>Header</td>
</tr>
<tr>
<td>Byte 2</td>
<td>ID</td>
</tr>
<tr>
<td>Byte 3</td>
<td>Length</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Type</td>
</tr>
<tr>
<td>Byte 5</td>
<td>Image Source</td>
</tr>
<tr>
<td>Byte 6</td>
<td>Move Speed</td>
</tr>
<tr>
<td>Byte 7</td>
<td>Move x</td>
</tr>
<tr>
<td>Byte 8</td>
<td>Move y</td>
</tr>
<tr>
<td>Byte 9</td>
<td>CheckSum</td>
</tr>
</tbody>
</table>

Table 2.6 Byte definition within Packet Structure

The Command Packet from the CN to the MN2 contains the following commands:

- Wireless network to be used by the video stream, 802.11 or 802.15.4,
- Movement commands for the USAR robot, and

Byte 5 within the Command Packet is used to select which network, 802.11 or 802.15.4, is used to transfer image data from the MN2 to the CN. The operator of the CS is able to select either the 802.11 or 802.15.4 network to transfer images from the USAR robot. The 802.11 network has the bandwidth to stream video images from the USAR robot, but should this become inoperable the operator can choose to use the 802.15.4 network and see image captures as opposed to a continual image stream.

Bytes 6 to 8 are movement commands for the USAR robot. As discussed in [42] there is an advantage in commanding an USAR robot to a new set of \( x \) and \( y \) coordinates as opposed to simply giving a direction in which the USAR must move. The advantage is that if communications is lost between the MN and the CN the USAR robot will stop at the required coordinates as opposed to continuing to move in the specified direction until communications is re-established.
The Acknowledge Packet has two functions. Firstly it acts as an Acknowledge that the Command Packet has been received. Secondly, operational parameters are returned to the CN to aid the CS operator. Byte 5 confirms to the CN which wireless network is being used to transfer image data to the CS. Byte 6 and 7 return the values from the Infra Red (IR) Distance Measurement Sensors (DMS) that are mounted on the USAR robots body.

**Dynamixel Network**

The Dynamixel modules are a range of serially controlled servo motors with each Dynamixel module based around an ATmega8 microcontroller. Dynamixels use a half-duplex multi-drop RS485 network for communications. Each Dynamixel is programmed with an ID, this ID is equivalent to the address of the network device. The Dynamixel network is a Star Topology and the Dynamixel modules act as slaves in the communication network, only responding to commands when requested.

Figure 2.20 shows the packet structure used to communicate with the Dynamixels and table 2.7 details the bytes allocation within the packets.

```
| 0xFF | 0xFF | ID | LENGTH | TYPE | Data[0] | ... | Data[N] | CHECK SUM |
```

Figure 2.20 Packet Structure for Dynamixel Communications

The USAR quadruped has twenty one Dynamixels from the RX range, the physical location of these Dynamixels is shown in figure 2.21.
Byte 0 and Byte 1 are headers bytes and defined as 0xFF. The next byte, Byte 2, is allocated as an ID byte and is required to identify the destination Dynamixel. Dynamixel communication packets have varying lengths, this is facilitated by having Byte 3 define the packet length. The length of the packet is the number of data bytes $N$ plus two, i.e. $N+2$.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Function</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 0</td>
<td>Header</td>
<td>0xFF</td>
</tr>
<tr>
<td>Byte 1</td>
<td>Header</td>
<td>0xFF</td>
</tr>
<tr>
<td>Byte 2</td>
<td>ID</td>
<td>None</td>
</tr>
<tr>
<td>Byte 3</td>
<td>Length</td>
<td>$N+2$</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Instruction</td>
<td>None</td>
</tr>
<tr>
<td>Byte 5</td>
<td>Data [0]</td>
<td>None</td>
</tr>
<tr>
<td>Byte $N+4$</td>
<td>Data [N-1]</td>
<td>None</td>
</tr>
<tr>
<td>Byte $N+5$</td>
<td>CheckSum</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2.7 Byte definition within Packet Structure
Byte 4 is defined as the Instruction byte and specifies what type of packet is being transmitted. There are seven possible packet types, these are:

1. 0x01 PING
2. 0x02 READ_DATA
3. 0x03 WRITE_DATA
4. 0x04 REG WRITE
5. 0x05 ACTION
6. 0x06 RESET
7. 0x83 SYNC WRITE

Positioning the Dynamixel can be done in one of three different ways, the first moves Dynamixels individually whilst the second and third methods allow the Dynamixel movements to be synchronised:

1. The WRITE_DATA command can be used to simply command an individual Dynamixel to a target position.
2. The REG WRITE command can be used to transmit a new position to a Dynamixel, this position is buffered until an ACTION packet is received. Using this method it is possible to send positions to as many Dynamixels as necessary, then when the ACTION command is received all Dynamixels will move in synchronisation.
3. The SYNC WRITE packet is a single command packet which is structured such that it contains positioning data for multiple Dynamixels. This packet is sent as a broadcast packet, thus all packets will read the SYNC WRITE packet and determine if data within the packets payload is requesting the Dynamixel to be moved.

Byte 5 to Byte N+4 are allocated for N bytes of data which is commonly known as the packets payload. The protocol definition for the Dynamixel specifies the allocation of bytes within the packet payload. This depends greatly on the type of Instruction that is being transmitted. Finally the last byte within the packet ByteN+5 is the packet CheckSum. The CheckSum is calculated using the following equation:
\[ \text{Checksum} = \neg(\text{ID} + \text{Length} + \text{Type} + \text{Data}[0] + \ldots + \text{Data}[N-1]) \]

where \( \neg \) is the NOT operator.

The RS485 network is therefore used to change the Dynamixel modules operational parameters as well as reading the Dynamixels operational variables in real time. This allows the master of the communication network, in this application the HyInt module, to read variables such as the motors shaft position and the current being drawn by the Dynamixels motor. Writing operational parameters such as the required goal position and the required motor speed allow great flexibility in the way the USAR quadruped’s limbs are sequenced to create a walking gait.

### 2.3.1.3 Processing System Algorithm

The HyInt module is the main processing system within the USAR quadruped robots processing system. Figure 2.22 shows a proposed algorithm which could be implemented in HyInt. Due to the volume of work within this research it was not possible to accomplish the implementation of this algorithm as the ‘Quadruped Motion Algorithm’ was not completed. Whilst some preliminary research was performed on a quadrupedal walking gait for this robot the developed walking gait was very rudimentary and only adequate to test the functionality of the robot. Further research will be conducted to refine the walking gait to incorporate quadruped and bipedal motion utilising the 2 DOF highlighted in figure 2.11.

The HyInt module interfaces with all the sub-modules of the USAR quadruped robot by either serial or USB connections. Thus the Interrupt Service Routines (ISR) are critical to ensure data flow is maintained without loss of packets.

Every process cycle the HyInt should send several Command Packets to both the HySense and HySense Lite modules. The data within the Return Packets will provide the readings of the 6-axis Inertial Measurement Unit (IMU), the Distance Measurement Sensors (DMS) and the 3-axis tilt-compensated digital compass. These variables should then be feed into the Quadruped Motion Algorithm, this algorithm is used to generate
the motion sequences of the USAR quadruped. The movement generated by this algorithm will depend on whether the quadruped has reached its most recent target position. If the target position has been reached then the sensor readings will be used to maintain the current quadruped stance. If, however, the target position has not been met then the algorithm will determine what motion sequences are necessary to reach the target position.

The next task that should be performed by the HyInt module is to determine if a Command Packet has been received from the Mobile Node (MN). If no Command Packet has been received then the HyInt module will continue to poll the HySense and HySense Lite modules until a Command Packet is received via the MN.

When a Command Packet is received from the MN an Acknowledge Packet is immediately issued to the Control Node (CN). Data from the Command Packet is then extracted and processed. The first task is to determine if the wireless network over which image data is being transferred has been changed by the operator. If this is the case then the image data is switched across to be transferred on the newly selected wireless network, 802.11 or 802.15.4.

The proposed algorithm that could be executed within the HyInt module, as shown in the figure 2.22, is quite basic and uses only a small percentage of the HyInt modules processing power. There is scope for more advanced algorithms to be developed for the HyInt module. Such algorithms could include image recognition and semi-autonomous navigation through a disaster zone and will be explored in further projects associated with this hardware.
Figure 2.22 Algorithm proposed for HyInt
2.3.2 USAR Tank based Robot Platform

The design of the tank based USAR is based on both the overview of USAR robot functionality presented in figure 2.1 and the design criteria presented in the introduction to section 2.3. Figure 2.23 shows the functionality that the tank based USAR robot I have designed possesses.

In this design the type of structure is a tank. The tank base will give the USAR robot good maneuverability over rough terrain, the ability to climb stairs and has the capacity to accommodate the hardware that will need to be fitted to automate the deployment of network nodes.
The USAR tank robot will have an onboard PC with custom designed electronics to interface to wireless communication hardware and the USAR robots motors. The onboard computing system will implement an algorithm that is predominantly manually controlled but some semi-autonomous features, like obstacle avoidance, may be implemented. The communication between the tank based USAR robot and the CS will be wireless and a complete discussion on the ad-hoc MESH wireless communication system is presented in sections 3 and 4 of this thesis. The teleoperation control at the CS will be performed by a PC or laptop, initially using the keyboard to issue movement commands but this may be extended to joy stick control once the system is operational.

The tank based USAR robot has been designed in such a way that additional functionality and hardware can be added for future projects. A 2-DOF Pan-Tilt camera has been mounted on a 3-DOF robot arm to increase the USAR robots ability to act as an inspection robot and gives the robots camera the ability to view 360° around the robot.

### 2.3.2.1 Mechanical Design

The attributes that were considered when selecting a tank base for an USAR robot were maneuverability and the ability to operate from a battery based power supply for an extended period. With these attributes in mind the MMP-40, which is manufactured by The Machines Lab, was chosen as the basis of the tank based USAR robot. This MMP-40 tank base weighs 18.1kg, has a top speed of 76 cm/s and can operate continuously for 2 hours. The tank base is 70cm in length, allowing it to traverse stairs and has a payload capacity of 13.6kg. Figure 2.24 shows the MMP-40 as supplied from the manufacturer.

A 24V$_{DC}$ 7.2Ah NiCd battery is supplied with the MMP-40 tank base. The manufacturer specifies that the tank base can be operational for 2hrs from the 7.2Ah batteries. To achieve this, the average current draw from the power source would be 3.6A. The MMP-40 tank base is fitted with 2 100W 24V$_{DC}$ electric motors which are fitted with 500 Cycle Per Revolution (CPR) 3 channel quadrature optical encoders. The encoders
are desirable as they can be used for dead reckoning positioning. Dead reckoning positioning could be used in addition to trilateration calculations from the wireless guidance system to provide accurate positioning within the disaster zone. Full specifications for the MMP-40 tank base can be found in Appendix D.

Figure 2.24 MMP-40 manufactured by The Machines Lab

**Drop Node Deployment System**

Figure 2.25 shows the linear actuator which has been designed to deploy DNs from the USAR robot. The Linear Actuator is based around a threaded rod that passes through the centre of the DN Deployment System (DNDS). The threaded rod passes through the center of the DNs and as the threaded rod turns the DNs are moved linearly towards the deployment location. Two guides are provided within the DNDS to ensure that the DNs cannot rotate upon the threaded rod and thus ensure linear motion.

The linear actuator uses a Dynamixel RX-28 module to turn the threaded rod. The Dynamixel module is able to determine the angle of its motor shaft and thus can accurately control the linear movement of the DNs. The Linear Actuators’ movements are controlled by the Fit-PC2 via an RS485 connection. When a DN moves to the end of the DNDS the DN will then pass through an opening in the floor of the USAR robot and on to the ground.
The linear actuator has also been designed in such a way that the DNs are able to determine when they have been deployed. This is critical as the DNs transceivers must only be enabled once the DN has left the USAR robot. This functionality has been achieved by connecting the power supply for the USAR robot to the guides within the DNDS. Figure 2.25 shows the power senses that have been included in the design of the DN enclosure. These power senses make contact with the power supply provided along the guides and the presence of power on these sensors confirms that the DN is still within the DNDS. Circuitry is added to the DN, as shown in figure 2.26, to monitor the power supply and pass a conditioned TTL signal to the DNs microcontroller.

When the DN detects that there is no power supply connected to its power supply sensors the DN realises that it has been deployed and the JN5148 will turn on its 802.15.4 transceiver. The discovery of a new 802.15.4 network node will then trigger a reconfiguration of the 802.15.4 based network, as detailed in section 4.3.1. Using the USAR robots power supply as an input to the DN also allows for the possibility of the USAR robots power supply to supply power to the DNs whilst within the DNDS. This would allow the DNs to conserve energy whilst housed in the DNDS and thus maximize their operational time once deployed.
Robotic Arm

A 3-DOF robotic arm with a 2-DOF Pan-Tilt camera was added to the top of the MMP-40. This robotic arm is shown in figure 2.27. The robot arm was designed around the RX series of Dynamixels from Robotis. Dynamixel modules are serially controlled servo motors which rely on a daisy chained RS485 to communicate back to the main controller. Each Dynamixel is assigned an ID and this ID is used to address the module when the main controller requires the Dynamixel to perform a function. To achieve an equivalent design using traditional PWM controlled servo motors would have required 5 PWM signals be generated and manipulated by an additional microcontroller. The fact that Dynamixel modules can be told to go to a target position at a defined speed eliminates the need for this extra microcontroller, simplifying the hardware within the USAR robot. The specifications for the RX series Dynamixels are available in Appendix A.

This robotic arm been designed so that when the robot arm is in its home position, i.e. at its minimum height, the Pan-Tilt camera can still be operated through its entire range of motion. This was done so that when the robot is traveling, the camera can be maneuvered by the operator to view the entire surrounding, not just what is in front of the robot. The robot arm has a span of 93 cm when fully extended, and by manipulating
the robot arm and the Pan-Tilt camera it is possible to see the entire 360° around the USAR robot.

An off the shelf Logitech C500 Web Cam is used for acquiring image data, this was chosen due to the availability of device drivers for Windows and Linux operating systems. The C500 is a colour camera with a maximum resolution of 1280 x 1024 pixels. This camera also has an inbuilt microphone, which could be useful to communicate with survivors within the disaster area.

2.3.2.2 Processing System Hardware Design

A design for the processing system of the tank based USAR is required as the MMP-40 tank base is only supplied with a 24V 7.2Ah NiCd battery, 2 24V DC motors and a motor controller. Figure 2.28 shows the block diagram of the processing system which was added to the MMP-40 tank base to allow it to function as an USAR robot.
As the processing system is powered from a battery, an energy efficient solution must be selected. Thus the core of the processing system is a fan-less Fit-PC2 computer. The Fit-PC2 has a low power consumption drawing a maximum 8W (0.67A @ 12V DC) when the CPU is operating at full load. The Fit-PC2 is based around Intel's Atom Z530 processor and US15W chipset and is capable of running either Windows or Linux based operating systems. For this research Ubuntu 9.10 was used as the operating system. The hardware specifications for the Fit-PC2 are shown in Appendix E.

The Fit-PC2 is connected with the sub-systems of the tank based USAR Rescue robot using the following interfaces:

- USB1: Mobile Node (MN).
- USB2: Logitech C5000 Web Cam, and
- USB3: Robotic Arm RS485 network,
- RS232: Motor Control – DAQ MCU module,

A standard packet was defined to enable software written in C to be reused in the FitPC2 and all connected microcontrollers, this is shown in figure 2.29 Table 2.8 defines the function of each byte used within this packet structure.
Byte 0 and Byte 1 of the standard packets are headers and defined as 0xFF. The next byte, Byte 2, is allocated as an ID byte and is required to identify the packets destination. This is required when more than one device is connected to the communication bus, such as the multi-drop RS485 network used for the Dynamixel modules upon which the robot arm is based. When packets are only being exchanged two devices then Byte 2 is not used.

<table>
<thead>
<tr>
<th>Byte</th>
<th>Function</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 0</td>
<td>Header</td>
<td>0xFF</td>
</tr>
<tr>
<td>Byte 1</td>
<td>Header</td>
<td>0xFF</td>
</tr>
<tr>
<td>Byte 2</td>
<td>ID</td>
<td>None</td>
</tr>
<tr>
<td>Byte 3</td>
<td>Length</td>
<td>N+2</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Type</td>
<td>None</td>
</tr>
<tr>
<td>Byte 5</td>
<td>Data [0]</td>
<td>None</td>
</tr>
<tr>
<td>Byte N+4</td>
<td>Data [N-1]</td>
<td>None</td>
</tr>
<tr>
<td>Byte N+5</td>
<td>CheckSum</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2.8 Byte definition within Packet Structure

Packets with variable lengths are accommodated for by this packet structure as Byte 3 defines the length of the packet. The length of the packet is the number of data bytes $N$ plus two, i.e. $N+2$. To allow packets to have different functionality Byte 4 is provided to define the type of packet that has been transmitted or received. Byte 5 to Byte $N+4$ are allocated for $N$ bytes of data, or that packets payload as it is commonly known. Finally the last byte within the packet Byte$N+5$ is the packet CheckSum. The CheckSum is calculated using the following equation:

\[
\text{CheckSum} = \neg (\text{ID} + \text{Length} + \text{Type} + \text{Data[0]} + \ldots + \text{Data[N-1]})
\]

where $\neg$ is the NOT operator.
This is an efficient CheckSum as the CheckSum can be calculated as a packet is being received. The following algorithm can is used to implement this:

```c
// both Header Bytes have been received – process the Rxed packet
PacketValid=FALSE;
loop=TRUE;
length=0;
Checksum=0;
for (i=0;loop==TRUE;i++)
{
  CheckSum+=RxedByte;
  if (i==1) length= RxedByte;
  else
  {
    if (i>1) length--;
    if (length==0) loop=FALSE;
  }
}
Checksum-=Checksum;
if (RxedByte) == CheckSum) PacketValid=TRUE;
```

**Mobile Node**

The Mobile Node (MN) hardware design is presented and discussed in section 4.3.1 of this thesis. The MN mounted within the tank based USAR robot is designated as MN1 and is configured as a router within the 802.15.4 based wireless guidance system whilst the Control Node (CN) is the coordinator within the network. The CN establishes a connection to the MN1 through the 802.15.4 network, section 4 of this thesis details the 802.15.4 based wireless robotic guidance system. Figure 2.30 shows the entire data flow from the Control Station to the Fit-PC2.

There are two types of packets that are used within this system, Command Packets and Acknowledge Packets. Command Packets are issued by the Control Station (CS) and flow through the system to the Fit-PC. These packets are used to issue commands to the USAR tank robots motor controller, the robotic arm and to select the network over which image data will be sent. Acknowledge Packets serve a dual purpose, they acknowledge that the Command Packet has been received as well as transferring operational variables, such as sensor readings and encoder pulse counts to the CS. Table 2.9 shows the data allocation within these packets.
The Command Packet from the CN to the MN contains the following commands:

- Wireless network to be used by the video stream, 802.11 or 802.15.4,
- Movement commands for the USAR robot, and
- Movement commands for the USAR robots 5 DOF Robot Arm.

<table>
<thead>
<tr>
<th>Command Packet</th>
<th>Acknowledge Packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte 0</td>
<td>Header</td>
</tr>
<tr>
<td>Byte 1</td>
<td>Header</td>
</tr>
<tr>
<td>Byte 2</td>
<td>ID</td>
</tr>
<tr>
<td>Byte 3</td>
<td>Length</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Type</td>
</tr>
<tr>
<td>Byte 5</td>
<td>Image Source</td>
</tr>
<tr>
<td>Byte 6</td>
<td>Move Speed</td>
</tr>
<tr>
<td>Byte 7</td>
<td>Move x</td>
</tr>
<tr>
<td>Byte 8</td>
<td>Move y</td>
</tr>
<tr>
<td>Byte 9</td>
<td>Robot Arm #1</td>
</tr>
<tr>
<td>Byte 10</td>
<td>Robot Arm #2</td>
</tr>
<tr>
<td>Byte 11</td>
<td>Robot Arm #3</td>
</tr>
<tr>
<td>Byte 12</td>
<td>Robot Arm #4</td>
</tr>
<tr>
<td>Byte 13</td>
<td>Robot Arm #5</td>
</tr>
<tr>
<td>Byte 14</td>
<td>CheckSum</td>
</tr>
<tr>
<td>Byte 0</td>
<td>Header</td>
</tr>
<tr>
<td>Byte 1</td>
<td>Header</td>
</tr>
<tr>
<td>Byte 2</td>
<td>ID</td>
</tr>
<tr>
<td>Byte 3</td>
<td>Length</td>
</tr>
<tr>
<td>Byte 4</td>
<td>Type</td>
</tr>
<tr>
<td>Byte 5</td>
<td>Image Source</td>
</tr>
<tr>
<td>Byte 6</td>
<td>Encoder</td>
</tr>
<tr>
<td>Byte 7</td>
<td>Front DMS</td>
</tr>
<tr>
<td>Byte 8</td>
<td>Left DMS</td>
</tr>
<tr>
<td>Byte 9</td>
<td>Right DMS</td>
</tr>
<tr>
<td>Byte 10</td>
<td>Camera DMS</td>
</tr>
<tr>
<td>Byte 11</td>
<td>Robot Arm #1</td>
</tr>
<tr>
<td>Byte 12</td>
<td>Robot Arm #2</td>
</tr>
<tr>
<td>Byte 13</td>
<td>Robot Arm #3</td>
</tr>
<tr>
<td>Byte 14</td>
<td>Robot Arm #4</td>
</tr>
<tr>
<td>Byte 15</td>
<td>Robot Arm #5</td>
</tr>
<tr>
<td>Byte 16</td>
<td>CheckSum</td>
</tr>
</tbody>
</table>

Table 2.9 Byte definition within Packet Structure
Byte 5 within the Command Packet is used to select which network, 802.11 or 802.15.4, is used to transfer image data from the MN to the CN. The operator of the CS is able to select either the 802.11 or 802.15.4 network to transfer images from the USAR robot. The 802.11 network has the bandwidth to stream video images from the USAR robot, but should this become inoperable the operator can choose to use the 802.15.4 network and see image captures as opposed to a continual image stream.

Bytes 6 to 8 are movement commands for the USAR robot. As discussed in [42] there is an advantage in commanding an USAR robot to a new set of x and y coordinates as opposed to simply giving a direction in which the USAR must move. The advantage is that if communications is lost between the MN and the CN the USAR robot will stop at the required coordinates as opposed to continuing to move in the specified direction until communications is re-established. Byte 9 to 13 specify the positions for the 5 Dynamixel modules that constitute the robot arm, this enable the CS to have complete control over the movement sequence of the Dynamixels within the Robot Arm structure.

The Acknowledge Packet has two functions. Firstly it acts as an Acknowledge that the Command Packet has been received. Secondly, operational parameters are returned to the CN to aid the CS operator. Byte 5 confirms to the CN which wireless network is being used to transfer image data to the CS. Byte 7 to 10 return the values from the Infra Red (IR) Distance Measurement Sensors (DMS) that are mounted on the USAR robots body and robotic arm. The DMS sensor used are Sharp GP2Y0A21YK, these devices give a non-linear analog output between 0.4V and 3.1V which represents 5cm to 80cm. This output from these sensors allows the operator to see how close the USAR robot is to any obstacles it encounters in the disaster zone. Bytes 11 to 15 of the Acknowledge Packet are used to confirm to the operator the current position of the robotic Arms Dynamixel modules.
Motor Control Data Acquisition module

The Fit-PC2 is the core module in the USAR tank robots processing system, but the Fit-PC2 has no digital or analog Input Output (IO) capability. So the Motor Control Data Acquisition (MCDAQ) module was designed to act as an interface between the Fit-PC and all sensors and actuators associated with the USAR Tank robot. Figure 2.31 shows the block diagram for the MCDAQ and the schematic is available in Appendix F.

A serial communication channel between the Fit-PC2 and MCDAQ is used to retrieve the sensor data and control the actuators. The standard packet structure, as shown previously in figure 2.29 is used for Fit-PC to MCDAQ communications. The Fit-PC2 will send a Command Packet to the MCDAQ module when it requires the USAR tank robot to move. The MCDAQ will acknowledge this with an Acknowledge Packet that will also contain the current values of the sensors. Table 2.10 displays the data allocation within the Command and Acknowledge packets.

Byte 0 of the Command Packet is used to initiate the Drop Node Deployment System (DNDS) which is documented in section 4.3. Byte 0 of the Acknowledge Packet displays the status of the DNDS and will be a 0x00 when it is idle and a 0x01 when the DNDS is active. Bytes 6 to 8 are the commands to move the USAR tank robot, these parameters tell the USAR tank robot where to move to (x, y coordinates) and the speed required. The MCDAQ must process this data and create two PWM signals that are feed to the Motor Controller. The Motor Controller is a Scorpion XXL v2 Speed Controller [51], this speed controller takes two PWM signals as it’s input, one for motor speed and the other for robot direction. This PWM signal is the standard PWM signal used by commercially available remote control products. The pulse is issued every 20ms, with the pulse being HI between 1ms and 2ms. A pulse width of 1.5ms will make the motors stop rotating, any signal above 1.5ms and below 2ms will make the motor rotate in the forward direction whilst a signal above 1ms and below 1.5ms will make the motor rotate in the reverse direction.
To ensure that the USAR tank robot moves to the right destination the inputs from the MMP-40’s 500 Cycle Per Revolution (CPR) 3-channel quadrature optical encoder are connected to the MCDAQ module. Each signal has a 10k pull-up resistor to pull the signal to 12V, then an RC filter circuit consisting of a 10k resistor and a 10nF capacitor is used to remove any high frequency noise. A 4504 is then used to step the signal down.
from 12V to 5V before the signal is passed through a 74HC14 Schmidt Trigger before terminating at a microcontroller input pin. The microcontroller input pin was selected such that an interrupt is generated each time the signal changes as either a positive going or negative going signal. Thus all encoder signal are conditioned by circuitry before generating interrupts within the microcontroller, which can be determine the distance moved by the USAR tank robot. The Encoder pulse count is also feedback to the Fit-PC as Byte 6 in the Acknowledge Packet.

The MCDAQ module has 8 digital outputs which use NPN transistors to drive status LEDs, these can be controlled by Byte 9 of the command packet. The MCDAQ module has four Distance Measurement Sensors (DMS) connected, these are Sharp GP2Y0A21YK and give a non-linear analog output between 0.4V and 3.1V which represents 5cm to 80cm. The DMSs are positioned on the front, right and left hand side of the MMP-40 while a 4th is mounted at the tip of the robotic arm on the Pan-Tilt camera mechanism. The reading of the DMSs are transferred to the Fit-PC2 in bytes 7 to 10 of the Acknowledge Packet. Eight Digital inputs are also provided on the MCDAQ module, these inputs are pulled HI to 12V via the MCDAQ module then a 4504 is used to step the signal down from 12V to 5V before the signal passes to the microcontroller input pins. The status of these eight digital inputs is transferred to the Fit-PC within the Acknowledge Packet as byte 11.

### 2.3.2.3 Processing System Algorithm

The Fit-PC2 is the main processing system within the USAR tank robots processing system. Figure 2.32 shows the algorithm which is implemented in the Fit-PC2. The Fit-PC2 interfaces with all the sub-modules of the USAR tank robot by either serial or USB connections. Thus the Interrupt Service Routines (ISR) are critical to ensure data flow is maintained without loss of packets.

Every process cycle the Fit-PC will send a Command Packet to the MCDAQ. This Command Packet will not contain any new commands but act as a request for the MCDAQ to send an Acknowledge Packet containing current operational data. This operational data is then buffered within the Fit-PC2. This enables the Fit-PC2 to
respond immediately to Command Packets that are received via the MN. After the MCDAQ has responded and the data within the Acknowledge Packet has been buffered the Fit-PC2 will then check to see if a Command Packet has been received from the MN. If no Command Packet has been received then the Fit-PC2 will continue to poll the MCDAQ module for operational data until a Command Packet is received via the MN.

When a Command Packet is received from the MN an Acknowledge Packet is immediately issued to the CN. Data from the Command Packet is then extracted and processed. The first task is to determine if the wireless network over which image data is being transferred has been changed by the operator. If this is the case then the image data is switched across to be transferred on the newly selected wireless network, 802.11 or 802.15.4.

Commands that are to be issued to the MCDAQ are then extracted from the MNs Command Packet and transferred to the MCDAQ. The MCDAQ will then send an Acknowledge Packet to acknowledge the new commands have been received, this Acknowledge Packet will again carry the latest operational parameter data which will be buffered within the Fit-PC2.

The tasks being performed by the Fit-PC2, shown in the algorithm in figure 2.32, are quite basic and use a small percentage of the Fit-PC2’s Intel's Atom Z530 processor power. The Fit-PC2 was chosen for this design as it allows for future expansion of the capabilities of the USAR tank robot. Such capabilities could include image recognition and semi-autonomous navigation through a disaster zone and will be explored in further projects associated with this hardware.
Figure 2.32 Algorithm implemented in Fit-PC2
2.4 Summary

The initial sections of this chapter took the form of a literature review that discussed the current status of USAR robot research. The final section then presented the designs for two USAR robots that were built as part of this research. The first USAR robot was a quadruped which has the agility to move quickly through a disaster zone with minimal disturbance to the surrounding area. The second USAR designed was a tank based robot which included a camera mounted at the end of a 5 DOF robot arm.

Chapters 3 through to Chapter 5 will now focus on the wireless robotic guidance system that was created to control these two robots. The reliability of the wireless guidance is enhanced by dropping wireless nodes from the tank based USAR robot when the wireless network is deteriorating. The dropped wireless nodes would be incorporated into the wireless robotic guidance system and allow the USAR robots to continue moving through the disaster zone. The mechanism for deploying these dropped wireless nodes resides in the tank based robot and was presented in the previous section of this thesis.

The next chapter will begin by reviewing wireless communication systems that could be used for a wireless robotic guidance system before concluding that an ad-hoc wireless network based on the 802.15.4 specification would be an ideal platform for such a system. A method of dynamically building the 802.15.4 ad-hoc wireless network to match the requirements of the disaster zone is then presented in the final section of the following chapter. The subsequent chapter will then present the design of an 802.15.4 based wireless guidance system, detailing the hardware design as well as the algorithms used at a network and a node level. The chapter following the design of the wireless robot guidance system will then discuss how this system can be used to localise an USAR robot. This chapter will also contain results of simulations and test results demonstrating the level of accuracy to which an USAR robot can be positioned within a disaster zone.
3 Wireless Communication

3.1 Introduction

Robots can be classified as autonomous, semi-autonomous, or teleoperated. Autonomous robots are capable of independently making complex decisions to complete a set of tasks based on the environment in which they are operating. Tele-operated robots are controlled remotely by an operator and make no decisions in relation to the tasks they are undertaking. In this case decisions are communicated by external entities. Whilst semi-autonomous robots are capable of making some decisions relating to their tasks but are also able to be controlled by an operator, e.g. a semi-autonomous robot may be told to move to a specific location by an operator but the robot could then determine the best path to follow to reach the destination.

Robots operating as either semi-autonomous or manually controlled require a transmission medium to allow the operator to communicate with the robot. There are two possibilities for this communication process. The first is a tethered arrangement where a fixed cable is connected between the operators Control Station (CS) and the robot. The second possibility is the use of a wireless link between the CS and the robot. Due to the nature of tasks which they undertake, Urban Search and Rescue (USAR) robots generally operate in hostile, rugged environments. The use of a tethered cable in these situations limits the distance that an USAR robot can enter the disaster zone and can result in the connecting cable becoming constantly entangled with objects within the disaster zone. Both these reasons result in movement through the disaster area being limited, slow and cumbersome. A wireless guidance system between the CS and the USAR robot would alleviate these problems and is the major reason why wireless communications is the preferred method of guidance for an USAR robot.

Even though autonomous robots operate independently of an operator sometimes they do need to return collected data to a CS for archiving and processing as well as accessing data from a remote database. This could be done by having the robot return to the CS and physically downloading the recorded data or have a method to download this
data remotely. Given the rugged terrain in which USAR robots operate having them return to download collected data could be time consuming and in some cases impractical. Wireless communications could offer a solution that allows an autonomous USAR robot to transfer recorded data to a CS whilst within the disaster zone.

Given there are significant advantages in having a wireless communication system between an USAR robot and a CS this presents two major technical challenges, both of which directly influence the distance that an USAR robot can travel in a disaster zone. The first of which is the finite power source of an USAR robot. Thus any wireless communication system installed on an USAR robot must be as energy efficient as possible.

The second technical challenge that needs to be overcome is the wireless transmission range of both the CS and the USAR robot. This transmission range of a wireless signal has a finite operational distance in ideal operating conditions, this is known as Free Space Path Loss (FSPL). FPSL defines the loss in signal strength in an ideal environment, i.e. the transmitter and receiver have a direct line of sight through free space. The FSPL equation is based on two effects:

1. Spreading of electromagnetic energy in free space which is given by the formula:

   \[ S = P_r \frac{1}{4\pi d^2} \]

   Rearranging this formula for \( P_r \) gives:

   \[ P_r = S4\pi d^2 \] \hspace{1cm} [3.1]

2. The receiving antenna’s aperture, which for an isotropic antenna is given by the formula:

   \[ P_r = S \frac{\lambda^2}{4\pi} \] \hspace{1cm} [3.2]
The equation for FSPL is given by ratio of transmitted power to received power:

$$FSPL = \frac{P_T}{P_R} = \left(\frac{4\pi d}{\lambda}\right)^2 \quad [3.3]$$

where:
- $P_T$ is the power within the transmitted signal,
- $P_R$ is the power within the received signal,
- $d$ is distance between the transmitter and receiver, and
- $\lambda$ is the wavelength of the signal.

When the wavelength, $\lambda$, is constant the equation for the FSPL shows that the loss in signal strength is proportional to the distance between the transmitter and receiver squared, i.e. $FSPL \propto d^2$. Thus, in an ideal situation where there is a line of sight path between transmitter and receiver there is a finite range over which a wireless link can remain operation.

As an USAR robot could be operating amongst debris and rubble, so it is also important to consider how this environment could affect the distance covered by wireless communications. In this instance the selection of the frequency of the wireless signal is important. Wireless signals with lower frequencies are preferred for operating in an indoor or hostile disaster environment as they have stronger penetration through concrete and rubble.

It is also possible for wireless signals to arrive at a receiver having travelled multiple paths. This can be caused by reflections from objects between the transmitter and receiver which can include the ground and mountains as well as manmade structures such as buildings. It is possible for the reflected signals to destructively interfere at the receiver resulting in an attenuated received signal. This phenomenon is known as multi-path fading and can also affect the transmission distance of wireless signals. In an USAR scenario debris within a disaster zone can be a cause of multi-path fading. An example of an environment that could create multi-path fading would be scattered debris within a collapsed structure made with re-enforced concrete. After the structure has collapsed the metal used within the re-enforced concrete would be arranged randomly thus creating many reflecting surfaces between the transmitter and receiver.
One method that can be used to determine when a wireless link is deteriorating is the Received Signal Strength Indicator (RSSI). RSSI is a metric used by radio receivers and represents the power present in a received radio signal. The RSSI is normally determined in the Intermediate Frequency (IF) stage of the receiver circuitry. An analog DC voltage output from the IF stage may be used to represent the RSSI. This analog DC voltage is passed to an Analog to Digital (A2D) converter resulting in the RSSI being represented by an arbitrary number. This number is then stored in a specified register within the receiver and can be accessed by a host microcontroller for analysis.

Manufacturers of radio receivers will specify the relationship between the arbitrary RSSI reading and the power in the received signal for their products. As an example Jennic manufacture a range of 802.15.4 compliant wireless transceiver modules. For their JN-5148 [52] module Jennic specify that “the RSSI represents the power of the received signal in the radio with 1dB resolution” [53]. Figure 3.1 shows the relationship between RSSI and the received signal power. For the JN-5148 the maximum RSSI value is 108 whilst “the intrinsic noise floor is the receiver limits the minimum reported RSSI value to 20” [53]. As can be seen in figure 3.1 the relationship between the RSSI and the received signal power is linear between the bounds of 20dB to 108dB.

![Figure 3.1 RSSI versus received signal power for Jennic JN-5148](image-url)
The implementation of a wireless guidance system for an USAR robot not only improves the mobility of the USAR robot within a disaster zone but can also provide accurate positioning information. This can be achieved by using the Received Signal Strength Indicator (RSSI) and Time of Flight (TOF) information of the wireless signals. As mentioned previously, the RSSI represents the power present in a received signal. The RSSI value decreases as the distance between the transmitter and receiver increases, therefore the RSSI can also be used to determine the distance between a transmitter and a receiver if the power of transmission is known. However, due to fading the linear relationship displayed in figure 3.1 may not be achievable in a real environment.

If the Jennic’s JN-5148 is again used as an example, an inverse square law based model is used to determine the relationship between RSSI and the transmission distance in free space. Figure 3.2 displays the relationship defined in equation [3.4].

\[
\text{Range} = 0.02 \times 10^{\left(\frac{108 - \text{RSSI}}{20}\right)} \quad [3.4]
\]

Distances calculated by RSSI readings give poor accuracy above 10m as the RSSI readings change in very small increments after 10m as shown by figure 3.2. In an USAR scenario RSSI readings will be affected as rubble and debris will be prevent a direct Line of Sight (LOS) between the wireless devices. In this case the power loss exponent would be larger than 2 and difficult to estimate. It is for this reason that RSSI based distance measurements are not ideally suited to distance determination in indoor or hostile disaster environments as the power loss formula cannot be accurately applied.

Results derived from TOF techniques are another method that can be used by wireless systems to determine the distance between the transmitter and receiver. The principle that is used to determine the TOF is illustrated in figure 3.3. The transmitter will send a POLL packet to the receiver and will record the time it takes for an ACK packet to be received, this is shown in figure 3.3 as \(T_{\text{TOTAL}}\). The time for the receiver to receive the POLL packet and issue a response is shown in figure 3.3 as \(T_{\text{ACK}}\).
Thus the TOF is given by the following equation:

\[ TOF = \frac{T_{TOTAL} - T_{ACK}}{2} \]  

[3.5]
As there is no direct connection between the transmitter and receiver it is difficult for the transmitter to know the exact time it has taken the receiver to respond to the POLL, i.e. $T_{ACK}$. There are two methods that could be used to determine $T_{ACK}$. The first would involve the value of $T_{ACK}$ being transmitted as data within the ACK packet, whilst the second method would involve the transmitter estimating $T_{ACK}$ and using this estimate in its TOF calculation. The second option here is possible if the ACK packet is generated consistently by receiver. One method of achieving this consistency would be for the transceivers hardware to create the ACK.

Jennic’s JN-5148 has a TOF ranging engine incorporated within its Baseband design to provide accurate TOF data. Figure 3.4 shows the TOF measurement implementation within the JN-5148.

![Figure 3.4 Time of Flight measurement implementation within JN5148](image)

Here $t_{TOF}$ is calculated using the following formula:

$$t_{TOTAL} = 2t_{TOF} + t_{TX1} + t_{RX2} + t_{PROC2} + t_{ACK2} + t_{TX2} + t_{RX1} + t_{PROC1}$$

$$\therefore t_{TOF} = \frac{t_{TOTAL} - (t_{TX1} + t_{RX2} + t_{PROC2} + t_{ACK2} + t_{TX2} + t_{RX1} + t_{PROC1})}{2}$$ \[3.6\]

where: $t_{TXx}$ is the time it takes to transmit a packet, $t_{RXx}$ is the time it takes to receive a packet, $t_{PROCx}$ is the time it takes to process a packet, and $t_{ACKx}$ is the time it takes to acknowledge a packet.
Jennic provide an Application Programming Interface (API) for their software protocol stacks to ensure that valid TOF information is acquired. When called the API will initiate the packet sequence shown in figure 3.5.

Initially a PRIME packet is sent to inform the receiver that TOF data is to be recorded, the receiver will then respond to this with a MAC generated ACK packet. A series of POLLS are executed, each being responded to by a MAC generated ACK. The transmitter will then send a DATA RQST packet asking for all the TOF information to be transmitted, which the receiver does by transmitting a DATA packet. This DATA pack is again acknowledged by a MAC generated ACK packet. The DATA packet contains the TOF readings for the receiver and allows the transmitter to determine the average TOF. The packet sequence shown in figure 3.5 takes 18ms. This consists of 4ms for each POLL and MAC ACK pair and 6ms for the remaining packets.

The ideal relationship between the TOF result and the distance between the transmitter and receiver in free space can be seen in the equations below. Assuming the fact that waves of a frequency of 2.4GHz travel at the speed of light \(c=299,792,458\) m/s, then the distance traveled \(d\) can express as shown in [3.7]. Substituting in the speed of light
constant and noting the fact that the TOF readings from the JN-5148 are in picoseconds results in equation [3.8]. This equation defines a linear relationship as shown in figure 3.6.

\[ d = \frac{vt}{c} \text{ or } d = ct \]  

\[ d = (3 \times 10^8)\left( TOF \times 1 \times 10^{-12} \right) \]

\[ d = 0.0003 TOF \]

Distance determination by TOF based calculations will lose accuracy in an environment where there is no Line of Sight (LOS) between the wireless devices and when multipath fading is present. In these cases the TOF readings will tend to bias the distance estimate as the signal path between wireless devices will be longer than an ideal LOS situation. This will be the case in an indoor or a disaster zone in which our USAR robots will be operating and must be considered when the results are calculated and analysed.
Regardless of whether the distance between transmitter and receiver was determined by RSSI or TOF these variables can then be used in trilateration calculations to accurately determine the co-ordinates of the USAR robot within the disaster zone. Having accurate co-ordinates for the USAR robot would then allow rescuers to accurately position any survivors found and thus determine the most efficient way to mount a rescue operation.

The following section will discuss possible wireless communication systems that could be used for the guidance of an USAR robot in a disaster zone. This will then be followed by a discussion detailing how a wireless USAR robot guidance system could be dynamically deployed to match the environment in which the USAR is operating.
3.2 Overview of Wireless Network Standards

There are many possible options to be considered when selecting a suitable platform for a wireless robotic guidance system. One possibility would be to select a wireless system that was based around existing cellular networks which are widely available for mobile phone coverage. In the 1980’s the first generation (1G) of wireless communication devices were released and were predominately used for voice communications. 1G networks used analog signals to transfer data between radio towers and mobile communications devices. In 1G systems voice data was modulated to a higher frequency, which was generally 150MHz. 1G networks offered little data capability and where available the downloads speeds varied between 2.9kbps and 5.6kbps.

There were no standards defined for 1G which resulted in 6 incompatible 1G systems operating across Western Europe alone. This prompted the creation of a committee, known as the Groupe Special Mobile (GSM), whose goal was to create a standard for a second generation (2G) mobile system that was rolled out across Europe and eventually all over the world. The resulting standard was known as GSM (which was renamed Global System for Mobile communications). The main improvements of the GSM standard was replacing analog transmission systems with digital air interface for all voice traffic and adding dedicated data connections for mobile devices.

GSM was first launched in Europe in 1991 using 900MHz spectral band. In 1993 GSM was expanded to operate within the 1800MHz spectral band. Where these two spectral bands were unavailable, such as the United States, spectral bands 850MHz and 1900MHz were reserved for GSM. Regardless of the spectral band used GSM systems uses Time Division Multiple Access (TDMA) and achieves a data rate of 9.6kbps. Base Stations are used to interface between mobile devices and infrastructure of the telecommunication provider and the transmission range between base stations and mobile devices can vary from hundreds of meters to tens of kilometers depending on the location and characteristics of the base stations antenna.
The evolution of 2G continued in 2000 with the introduction of General Packet Radio Service (GPRS) which could provide data rates from 56kbps to 115kbps. This enabled mobile devices to utilize data services such as Wireless Application Protocol (WAP), Multimedia Messaging Service (MMS) as well as email and World Wide Web (WWW) access. The introduction of Enhanced Data Rates for GSM Evolution (EDGE) in 2003 further improved the data rates achieved by GPRS by a factor of 4, with a maximum achievable data rate of 473.6kbps.

The use of a 2G system as a robot guidance system was proposed in [54] and [55]. A teleoperated robot guidance system over GPRS was presented in [54]. This paper detailed how a control station issued movement commands to a semi-autonomous robot over a GPRS communication link. A GSM guidance system was proposed in [55]. This paper proposed using a conventional mobile phone as the guidance system. Dual-Tone Multi-Frequency (DTMF) signaling from the mobile phone were used as guidance commands, these DTMF tones were then decoded by the robots processing system and the appropriate movement sequence initiated. This paper also investigated returning images from the robot via MMS over the GSM network.

Third Generation (3G) mobile systems are based on the International Mobile Telecommunications-2000 (IMT-2000) standard and have been commercially available since 2001. The IMT-2000 standard requires that a mobile 3G system have a minimum data rate of 200kbps but many 3G systems have data transfer rates up to 2Mbps. 3G systems can be either based on the Universal Mobile Telecommunication Systems (UMTS) or the CDMA2000 standard. The latest UMTS release Evolved High-Speed Packet Access (HSPA+) allows theoretical peak data rates of 56Mbps for downloaded data and 22Mbps for uploaded data, whilst the latest release of CMDA2000 Evolution-Data Optimisation (EVO) allows for 3.1Mbps for downloaded data and 1.8Mbps for uploaded data. These high data rates have allowed mobile broadband to become a feature on mobile devices, e.g. smartphones.

In 2008 the International Mobile Telecommunications Advanced (IMT-Advanced) specified the requirements for the Fourth Generation (4G) mobile system. One of the major changes in the 4G specifications was that 4G systems would be purely packet-
switched networks. This meant that voice communications, which had previously been circuit-switched systems, would be purely Voice over IP (VoIP) for 4G systems. Long Term Evolution (LTE) is one standard that has been rolled out commercially as a 4G system. LTE is based on GSM/EDGE and UMTS/HSPA and the first commercial LTE based 4G system was launched in late 2009. LTE has a theoretical download data rate of 299.6 Mbps and an upload data rate of 75.4 Mbps.

Basing a robotic guidance system on 2G, 3G or 4G technology would have some advantages. The data rates available on these cellular networks would be sufficient to send guidance commands and the transmission range from base stations to mobile devices would be more than adequate to give coverage to the CS and the USAR robot. Figure 3.7 shows how the existing infrastructure for a cellular network could be used. In this illustration the disaster zone is shown in red and spans four different cells. Movement of a mobile USAR robot in such a disaster area would result in the USAR robots network interface having to perform handovers to appropriate base stations to ensure network coverage, but the USAR robot would remain connected to the CS and able to receive guidance commands.

![Figure 3.7 Traditional Cellular network structure](image)

The use of the cellular networks infrastructure would mean that the only hardware that needed to be designed would be network interfaces for the CS and the USAR robot. Telit wireless solutions produce a range of modules which cover GSM/GPRS, UMTS and LTE standards [56]. These modules could easily be integrated with an embedded
microcontroller based system to provide the functionality required to implement a robotic guidance system at the CS and USAR robot over a cellular network.

However, whilst the existing 2G, 3G or 4G infrastructure could be seen as an advantage for a robot guidance system it can also be seen as one of the major disadvantages. Events such as earthquakes, tsunamis, cyclones, tornados or explosions can cause widespread damage. It is highly likely that such a catastrophic event would also severely damage or destroy the cellular infrastructure. If the robot guidance system were based on these technologies then the cellular base stations would need to be repaired or replaced before the USAR robot could be deployed. If the disaster zone were in a remote location where there was limited or no cellular network coverage then mobile base stations would need to be deployed to provide the wireless infrastructure needed for a cellular based robotic guidance system.

Additionally, cellular systems are designed to work predominately over a large outdoor area but cellular signals can penetrate to some degree indoors. It is envisaged that the disaster zone in which an USAR robot would operate would more closely resembled a confined indoor environment. A wireless network that is able to expand its network coverage as the disaster zone is explored would be highly advantageous [40, 41]. This scalability is a feature of ad-hoc MESH wireless networks. Ad-hoc wireless MESH networks can be created such that there is more than one possible communication path between a CS and an USAR robot. This provides a high level of redundancy to the wireless network, should one path become inoperable for some reason then it may be possible to establish a new path within the MESH networks topology. Thus a portable short range wireless communication system that created an ad-hoc MESH network would be better suited for an USAR application.

Figure 3.8 shows the topology of a MESH network. In a MESH network the network nodes must be capable of not only receiving and processing data but also capable of retransmitting data if the received data was intended for another network node. The MESH network shown in figure 3.8 shows how a MESH network might be used in an USAR operation. In this particular scenario a Coordinator node (CN), a Routing Node (RN) and a Mobile Node (MN) are shown. In a robot guidance scenario the data source
is the CN and the destination is the MN. Data is transmitted from the CN and travels through a network of RNs before reaching the MN. Each time a packet is passed through a RN towards the MN a ‘hop’ has occurred. The number of ‘hops’ between the CN and the MN is one of the parameters that can be used in determining the optimal route between CN and MN in a MESH network.

![Figure 3.8 MESH network structure](image)

In a MESH network there are many possible paths that data may traverse when travelling from the source node to the destination node. This is one of the features of MESH networks that make this topology so resilient. If a network node were to fail then it is possible for the network to reconfigure and ensure that a path still exists between the source node and the destination node. A network that can perform this type of reconfiguration is referred to as a ‘self healing’ network.

The path which the data travels from the source node to the destination node within a MESH network depends on the routing strategy implemented. There are numerous routing strategies that may be implemented which can be either static or dynamic, and change based on network traffic. The selected routing strategy depends largely on the specific needs of the application, some applications will require that the speed of data
from source to destination must be optimised whilst other applications may require a guarantee that data from source to destination is delivered.

Paper [57] discusses how communication delays within a wireless network can affect the performance of a robotic guidance system. Here a satellite communication link is used between a control station and an USAR robot. Findings revealed that sending the robot updated target coordinates resulted in improvements in the USAR robots maneuverability. Conventional methods of sending specific movement commands to USAR robots, such as motor speeds, developed problems when the wireless link was delayed or intermittent. Given that satellite communication links have a minimum delay of 270ms there is an inherent delay incorporated into all movement commands. Using specific movement commands an USAR robot would be told to set its motor speeds to head in a certain direction and do so until commanded not to. This creates a problem should command packets be delayed or lost entirely as the USAR robot would continue along the set path until communications could be reestablished. This could cause the USAR to bump into obstacles or worst case drive itself out of the wireless networks coverage. This specific problem highlights the problems associated with delayed and lost packets within a wireless guidance system and the operational implications these can have for the USAR robot.

Thus, the reliability of ad-hoc MESH networks and their deployment flexibility makes this network topology ideal for communications between the USAR robot and the CS within a disaster zone. In the paper by Freeman, J. [58] the use of a 802.15.4 based MESH network within a hazardous environment is presented. In this paper an autonomous USAR robot searches randomly through a disaster zone using a range of on board sensors to detect injured people. When an injured person is detected the USAR robot will determine its position using Received Signal Strength Indicator (RSSI) measurements from two nodes that have fixed reference positions. After positioning itself via trilateration calculations the USAR robot will then use the MESH network to transmit the co-ordinates of the injured person to closest rescuers.

An additional paper by Freeman, J. [59] discusses the use of a ZigBee based Wireless Sensor Network (WSN) that also acts as a robotic guidance and localisation system in
an Indoor Environment. Here a semi-autonomous robot uses a WSN to receive movement commands and determine its position via trilateration calculations from RSSI readings. The calculated position is then used as one of the parameters for the semi-autonomous robots path planning algorithm.

The EUROPCOM project [60-63] details how a Ultra-WideBand (UWB) ad-hoc network MESH can be used within a disaster zone. EUROPCOM allows human rescuers to remain in constant wireless communications with a control station. This is achieved by having the rescuers carry a wireless communication device known as a Mobile Unit (MU) which communicates to the control station via a Control Unit (CU).

As the rescuer moves through the disaster zone they monitor the signal strength to their MU, when this gets low they deploy a Drop Unit (DU). The deployment of a DU triggers a reconfiguration of the ad-hoc MESH network with the DU inserting itself in the wireless network between the MU and the unit whose signal strength was weakening. Thus deploying DUs in this manner enables the wireless network to self-heal and in doing so guarantees that the rescuer remains in constant contact with the control station.

EUROPCOM project also uses trilateration between the network nodes to accurately determine the position of the rescuers within the disaster node. To achieve this an additional unit known as a Base Unit (BU) are used. BUs have GPS fitted and these network nodes act as reference positions that the trilateration calculations are based on.

As mentioned previously, the EUROPCOM system relies on human rescuers deploying DUs to create the ad-hoc MESH network. My system will use an USAR robot to create the ad-hoc MESH network. This will be achieved by allowing the USAR robot to monitor the RSSI and when necessary deploy a DU. There are three possible candidates that could be considered as a platform for such a robot guidance systems ad-hoc network, these are Wi-Fi, Bluetooth or a network built upon the 802.15.4 standard such as ZigBee. All three of these options are capable of forming ad-hoc wireless networks in the Industrial, Scientific and Medial (ISM) radio bands (2.45Ghz and 5.8GHz).
Wi-Fi is the common terminology for a Wireless Local Area Network (WLAN) that meets the specifications of the Institute of Electrical and Electronic Engineers (IEEE) 802.11 standards. The 802.11 standards detail the WLAN Medium Access Control (MAC) and the Physical Layers (PHY) as shown below in Figure 3.9.

The original 802.11 standard specified that the PHY could be implemented as either Frequency Hopping Spread Spectrum (FHSS) or Direct Sequence Spread Spectrum (DSSS) radio transmissions or as InfraRed (IR) signals. The first 802.11 standard was proposed in 1997 and since then a range of amendments have resulted in subsequent standards as shown in table 3.1. Orthogonal Frequency Division Multiplexing (OFDM) was introduced to the PHY layer in 802.11a and increased the maximum data rate to 54Mbps. The transmission range of the protocols listed in Table 3.1, excluding 802.11, is 35m to 70m which would be more than adequate in an ad-hoc MESH network.
### Table 3.1 802.11 Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Release</th>
<th>Freq (GHz)</th>
<th>Maximum Data rate (Mbps)</th>
<th>Modulation</th>
<th>Indoor Range (m)</th>
<th>Outdoor Range (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11</td>
<td>1997</td>
<td>2.4</td>
<td>2</td>
<td>DSSS, FHSS</td>
<td>20</td>
<td>100</td>
</tr>
<tr>
<td>802.11a</td>
<td>1999</td>
<td>5</td>
<td>54</td>
<td>OFDM</td>
<td>35</td>
<td>120</td>
</tr>
<tr>
<td>802.11b</td>
<td>1999</td>
<td>2.4</td>
<td>11</td>
<td>DSSS</td>
<td>35</td>
<td>140</td>
</tr>
<tr>
<td>802.11g</td>
<td>2003</td>
<td>2.4</td>
<td>54</td>
<td>OFDM, DSSS</td>
<td>38</td>
<td>140</td>
</tr>
<tr>
<td>802.11n</td>
<td>2009</td>
<td>2.4, 5</td>
<td>600</td>
<td>OFDM</td>
<td>70</td>
<td>140</td>
</tr>
</tbody>
</table>

802.11’s MAC layer consists of two co-existing operational modes, the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). Figure 3.10 illustrates how the two functions use a cyclic structure that allocates a time period when the two modes can transmit data packets. This cyclic structure is known as the Collision Free Period (CFP) and a beacon is used to define the start of the CFP. Within the CFP two periods are defined. The first is the Contention Free Period (CFP) for devices operating in the PCF mode whilst the second is the Contention Period (CP) for devices operating in the DCF mode. Devices operating in DCF mode transmit data packets in accordance with the CP protocol but collisions may occur between data packets as the CP protocol does not assign specific timeslots in which devices can transmit. PCF mode devices are assigned specific timeframes in which they can transmit data packets and as such applications that require a guaranteed Quality of Service (QoS) must use the CFP protocol.

![Figure 3.10 802.11 Collision Free Period Repetition Interval](image)

Most commercial devices only implement the DCF mode, thus “the fundamental channel access mechanism for the IEEE 802.11 MAC is DCF” [64]. DCF is also known as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) which is a random access mechanism that enables sharing of wireless media. Before a device operating in DCF mode can transmit it must first sense the wireless medium to see if it
is currently in use. Once the device has deemed the wireless medium is free it will wait a specified timeframe before transmitting, this timeframe is referred to as the Interframe Space (IFS). Once data has been transmitted the destination device must issue an ACK immediately it successfully receives the data packet. If a data packet is lost and a retransmission is required the transmitting device must wait an additional random backoff time before retransmitting the data, this is done to minimise the chances of a collision on the subsequent retransmission.

Figure 3.11 shows the operating modes for 802.11. The infrastructure mode shown in figure 3.11 (a) allows a wireless device to connect to a Local Area Network (LAN) via a Wireless Access Point (WAP). In this mode the wireless connection allows the wireless device to perform all the tasks it normally would do if it had a physical connection to the LAN.

Figure 3.11 (b) shows 802.11’s ad-hoc mode which allows wireless devices to create a point-to-point connection over which data can be shared. This ad-hoc mode could be used in an USAR robot scenario but this would require the wireless device to be capable of creating and maintaining a point-to-many connection which would require the wireless device to act as a router node.

Another option for a platform for a wireless ad-hoc network would be Bluetooth, a wireless standard which meets the 802.15.1 Wireless Personal Area Network (WPAN) specification. The Bluetooth protocol stack is shown in figure 3.12. Bluetooth was
originally conceived by Ericsson in the mid-1990’s as a wireless replacement for RS232 and USB type cables and is now widely used in Laptops, Smart Phones, Tablet PC’s and wireless mice and keyboards.

Bluetooth uses 70 1MHz channels operating over the 2.400.0GHz to 2.483.5Ghz spectral band. Frequency Hopping Spread Spectrum (FHSS) is implemented with 800 frequency hops per second performed when Adaptive Frequency Hopping (AFH) is enabled. Bluetooth v1 uses Guassian Frequency-Shift Keying (GFSK) and is able to achieve a maximum application throughput of 0.7Mbps. The introduction of Bluetooth v2 saw the introduction of $\pi/4$-DQPSK and 8DPSK modulation which increased the maximum application throughput of 2.1Mbps. Bluetooth v3 has also been defined, but these specifications only use Bluetooth to negotiate and establish a communications link over a collocated 802.11 link. Bluetooth devices were designed to be either Class 1, Class 2 or Class 3 devices with the transmission range of each 1m, 10m, 100m respectively.
The Bluetooth specification implemented a Master-Slave structure, where each Bluetooth network has a single master and can have a maximum of seven slaves. Such a network was referred to as a Piconet and is shown on figure 3.13 which is a traditional Star network topology. Due to the Master-Slave structure Slave nodes must synchronise their clocks with that Master and can only communicate with other Slave nodes via the Master.

![Figure 3.13 Bluetooth Piconet](image1)

The Bluetooth specification allows a node to act as a Master and Slave node simultaneously. Such a device allows piconets to be combined to form a larger network known as a Scatternet and is shown in figure 3.14.

![Figure 3.14 Bluetooth Scatternet](image2)
Whilst a Scatternet could be used as a robot guidance system one major problem would be the Master-Slave structure. A Scatternet requires multiple Masters which would mean there would be no standardised or centralised control. It may be possible to implement a Scatternet where there was a Master that could control all the other Masters but this would add another layer of complexity to the software implementation of the scatternet. Also Master/Slave nodes are the only nodes permitted to connect Piconets. In a Search and Rescue scenario this could mean a failure of a Master/Slave node could result in a loss of communication between the CS and the USAR robot. The fact that a Scatternet topology is a group of interconnected Star networks means that it may not be as resilient as a MESH network. Again, this could be compensated for in the software implementation by having Slaves waiting to become Master/Slave nodes should an existing Master/Slave node fail but this would add another layer of complexity to the software implementation.

The need for a specification that could create and easily maintain dynamic ad-hoc MESH wireless networks was one of the primary motives that prompted the Institute of Electrical and Electronic Engineers (IEEE) to develop the 802.15.4 standard. 802.15.4 defines a Low-Rate Personal Area Network (LR-WPAN) with the aim to provide a standard for low-cost, low-speed and low-power communication devices. The protocol stack for 802.15.4 contains a Medium Access Control (MAC) layer and a Physical (PHY) layer as shown in figure 3.15.

The standard PHY layer defined three frequency bands in which 802.15.4 could broadcast, these are detailed in table 3.2. The original 2003 specification based the PHY layer on Direct Sequence Spread Spectrum (DSSS). The revised 2006 specification
specified four different PHY layers, there of these were based on DSSS with different modulation techniques whilst a Parallel Sequence Spread Spectrum (PSSS) was introduced for the 868MHz and 915MHz frequency bands. A further revision in 2007 saw the number of defined PHY layers increased from four to six with a Direct Sequence Ultra-wideband (UWB) and a Chirp Spread Spectrum (CSS) being introduced.

<table>
<thead>
<tr>
<th>Frequency Range</th>
<th>2003</th>
<th>2006</th>
<th>Geographical Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>868.0 – 868.6 MHz</td>
<td>1 channel 20 &amp; 40 kbps</td>
<td>3 channels 100 &amp; 240 kbps</td>
<td>Europe</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>902.0 – 928 MHz</td>
<td>10 channels 20 &amp; 40 kbps</td>
<td>30 channels 100 &amp; 240 kbps</td>
<td>Australia and North America</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2400 – 2483.5 GHz</td>
<td>16 channels 240 kbps</td>
<td>16 channels 240 kbps</td>
<td>Worldwide</td>
</tr>
</tbody>
</table>

Table 3.2 Evolution of 802.15.4 specifications

The 802.15.4 standard defines two different types of nodes that can exist within the network:

1. Full Function Device (FFD)
   (i) PAN Coordinator
   (ii) PAN Router
2. Reduced Function Device (RFD)

The main differentiation between a FFD and an RFD is that an RFD cannot be a PAN Coordinator and whilst an FFD can establish multiple connections to other nodes a RFD can only establish a single connection.

Each wireless network has at least one FFD node that acts as the coordinator of the PAN. Additional FFDs can exist within the wireless network but they function only as routers. RFD nodes are devices that establish connections with FFDs and perform basic tasks. As the RFDs are simple devices they can be designed with a minimum number of components and as such their power consumption is very low, maximising battery life.
Using the Network and Application layers allows FFD and RFD devices to form Star, Tree or MESH network topologies as shown in figure 3.16. The MESH network capability makes an 802.15.4 based wireless network an ideal platform for an USAR situation.

![Network Topologies](image)

(a) Star  (b) Tree  (c) MESH

Figure 3.16 802.15.4 Network Topologies

The MAC layer implemented in 802.15.4 uses an optional Superframe Structure to control access to the wireless media, the superframe structure is shown in figure 3.17. The superframe is managed by the PAN Coordinator and is bound by beacons which are issued by the coordinator at regular programmable time intervals. The beacon itself contains information that is used to help network devices synchronise to the network. This information includes the network identifier, specifics of the superframes structure as well as the period between beacons. The period between beacons is known as the Contention Access Period (CAP) and is broken into sixteen time slots. Network devices that wish to communicate with the PAN Coordinator use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) during the CAP to reduce network traffic collisions.

The PAN Coordinator can reserve some of the time slots within the superstructure for specific network devices. These time slots are known as Guaranteed Time Slots (GTS) and are grouped at the end of the superframe. The time period occupied by GTS within the superframe is known as the Contention Free Period (CFP). The 802.15.4 specification does restrict the number of GTS available within a superframe to allow
devices using the CAP and CSMA/CA reasonable access to the wireless media. Thus, devices wishing to maintain a guaranteed level of QoS can negotiate with the PAN Coordinator to be allocated sufficient time within the superframes CFP. For a wireless robotic guidance system having sufficient GTS allocation is desirable as this would improve the level of control exerted over the USAR robot by the operator.

![Superframe structure with GTS](image)

Figure 3.17 Superframe structure with GTS

The 802.15.4 MAC provides a fully acknowledged CSMA/CA protocol. Each data packet which is transmitted must be immediately acknowledged by the destination device. There are three possible scenarios that can occur when a data packet has been transmitted. The first is the data packet is received successfully, an acknowledge is issued and it too was received successfully. This is the ideal scenario where no packet collisions were experience. The second and third scenarios involve some form of packet collision. The second scenario occurs when the data packet fails to reach the destination resulting in no acknowledgement. Whilst the third scenario occurs when a data packet is transmitted, received successfully and an acknowledge is issued but the acknowledge is corrupted and never reaches the original source. Both second and third scenarios require a retransmission of the original data packet. A random backoff timer is used to delay the retransmission to avoid the possibility of an further collisions in the wireless media.

ZigBee specifies a suite of communication protocols that are based on the IEEE 802.15.4 standard. The ZigBee specifications are controlled by the ZigBee Alliance, a non-profit organisation whose members consist of businesses, universities and government agencies. In 2002 the ZigBee alliance was formed with 25 members and by 2010 had grown to more than 400 members. The initial ZigBee specification was
released by the ZigBee Alliance in 2004. Revisions to the standard were released in 2006 and 2007. The 2007 revision is the current revision and defines two stack profiles:

1. Profile 1 – ZigBee: targeted at home and light industrial applications.
2. Profile 2 – ZigBee PRO: offered advanced networking capabilities such as multicasting, many-to-one routing and improved security.

ZigBee adds to 802.15.4’s PHY and MAC layers by specifying the Network Layer (NWK), the Application layer, ZigBee Device Objects (ZDOs) and manufacturer defined application objects (APOs). Figure 3.18 shows ZigBee’s protocol stack.

ZigBee’s PHY is based on 802.15.4’s PHY layer, thus the radio transmission characteristics for ZigBee are shown in table 3.3. There are three frequencies bands that may be used for ZigBee, all of which use Direct Sequence Spread Spectrum (DSSS) techniques. The transmissions in the 868MHz and 915Mz band use Binary Phase-Shift Keying (BPSK) for modulation whilst the transmissions in the 2.45GHz band use Offset Quadrature Phase-Shift Keying (OQPSK) modulation. Within each spectral band
specific channels are allocated, e.g. sixteen channels with 5MHz bandwidth are allocated within the 2.4GHz band.

<table>
<thead>
<tr>
<th>Frequency Range</th>
<th>Number of Channels</th>
<th>Throughput</th>
<th>Modulation</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>868.0 – 868.6 MHz</td>
<td>1 channel</td>
<td>20 kbps</td>
<td>BPSK DSSS</td>
<td>Europe</td>
</tr>
<tr>
<td></td>
<td>20 kbps</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>902.0 – 928 MHz</td>
<td>10 channels</td>
<td>40 kbps</td>
<td>BPSK DSSS</td>
<td>Australia and North America</td>
</tr>
<tr>
<td></td>
<td>40 kbps</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2400 – 2483.5 GHz</td>
<td>16 channels</td>
<td>240 kbps</td>
<td>OQPSK DSSS</td>
<td>Worldwide</td>
</tr>
<tr>
<td></td>
<td>240 kbps</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3 ZigBee specifications

The 802.15.4 MAC is also used within the ZigBee specification. Thus ZigBee also uses the superframe structure to allow both a Contention Access Period (CAP) and a Contention Free Period (CFP). The ZigBee Coordinator is responsible for determining how many Guaranteed Time Slots (GTS) are assigned within the CFP. GTS are requested by ZigBee devices wishing to negotiate a level Quality of Service (QoS) with the Coordinator. ZigBee devices operating within the CAP use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) to minimise packet collisions in the wireless medium.

Routing within a ZigBee network is provided by the Network Layer (NWK) which may implement a number of different routing schemes, one of which being the Ad-hoc On-Demand Distance Vector (AODV) routing scheme. This routing scheme issues a broadcast to determine the lowest cost path between a source node and the destination node. The initial broadcast is sent to all of the sources child nodes. If these nodes have children of their own they will retransmit the broadcast. When these broadcasts arrive at the destination node the destination node will determine which path had the lowest cost and will reply to the source node via this path. The source node will then update its routing table with the path specified by the destination node and use this for all further inter-node communication.

In addition to providing the ZigBee standard the ZigBee Alliance also provides a number of application profiles so that products produced by different manufacturers which are intended for the same market are inter-operable. This is achieved by defining APOs within the application layer. It is possible for a ZigBee device to be performing
multiple applications and each application that is running requires the ability to create and receive messages. To achieve this endpoints are created and associated with APOs. Each ZigBee device can have two hundred and forty endpoints which are given unique identifiers number from 1 to 240. When sending messages between ZigBee devices the allocation of endpoints must be consistent so that the message is received by the correct application. This consistency is achieved by the definition of endpoints within an application profile.

As an example, an application profile for Home Controls-Lighting (HCL) defines the endpoints that a ZigBee device that reads a switch can use to send a message to a ZigBee device that controls a light. The information that passes between ZigBee devices in this way is referred to as an attribute, and in the case of the light switch and light example this attribute would simply be OnOff, a variable that defined the state of the switch. Additional application Profiles cover a range of applications from Building Automation and Healthcare to Remote Control devices supporting RF4CE.

A special APO is the ZDO. The ZDO is available on all devices and is allocated an endpoint 0. The ZDO is interface to the ZigBee Device Profile (ZDP) which is responsible for network discovery, network configuration and network maintenance. The ZDO also defines the role of a specific device on the network, these roles being:

1. ZigBee Coordinator (ZC),
2. ZigBee Router (ZR), and
3. ZigBee End Device (ZED).

The network topologies available for ZigBee networks are shown in figure 3.17. Each ZigBee network can only contain one ZC and can be configured as either a Star, Tree or MESH network. The ZC is the node tasked with starting the network and does so by selecting the operational channel and inviting other nodes to join the network. In the Star topology the ZC can be used to transfer data between nodes. In the Tree and MESH topologies the ZC controls the network routing and is responsible for maintaining the system routing table.
As defined in the MAC layer, communications within a ZigBee network can occur within either the CAP or the CFP segment of the superframe structure. ZigBee devices that have negotiated a GTS with the ZC are able to sleep for the majority of the time and wake on prearranged time slots when the device expects to receive a message. Sleeping in this manner allows a ZigBee device to turn off its radio frequency (RF) circuitry when it knows there are no scheduled transmissions, and thus enter a low-power sleep mode. When a ZigBee device is utilising the CAP to communicate upon the wireless medium the ZC and the ZR must remain powered at all times, as messages may be initiated randomly by any of their child ZEDs. ZEDs however are able to utilise the sleep mode and enter into a low-power mode if they are configured to wake when an event occurs, such as a light switch is activated. In this manner the ZED would wake, wait for a beacon to signify the start of the CAP and then initiate a message sequence.

Standard ZigBee devices have a transmission range of approximately 50m. The transmission range is however dependent upon the type of antenna fitted to the ZigBee device as well as the environment in which the device is operating. Some commercially available ZigBee devices have Low-Noise Amplifiers (LNA) included in the ZigBee radio module and these can extend the transmission range up to 4kms. Whilst the transmission range of an individual ZigBee node is quite small compared to a cellular network, a MESH network consisting of multiple ZigBee devices can grow to cover a large distance. The MESH style network is highly suited for a USAR robot guidance
system. If the USAR robot deploys ZigBee nodes as it travels through the disaster zone then the network coverage will continue to expand as needed. This will generate a highly flexible and efficient wireless network as the network coverage perfectly matches the USAR robots requirements.
3.3 Wireless System for USAR Robots

In the previous section the possible systems that could be used as a wireless robotic guidance system were discussed. As the intended search and rescue scenario for this research would be similar to an indoor environment it was concluded that an ad-hoc wireless network based on 802.15.4 would provide an ideal system. This sub-section will now discuss how a wireless guidance system could be deployed within a disaster zone and in doing so demonstrate the merits of using a wireless MESH network as a robotic guidance system. Section 4 of this thesis will then expand on the concepts discussed in this section, detailing the hardware design of the wireless nodes as well as the algorithms required to add nodes dynamically to the network and recover, or self-heal, when the structure of the network is compromised.

The EUROPicom [61] model describes how rescuers create an ad-hoc wireless network as they move through a disaster zone. The rescuers monitor the signal strength between their Mobile Unit (MU) and when this becomes weak they deploy Drop Units (DU). This deployment then triggers a reconfiguration of the wireless network with the DU connecting to the existing ad-hoc wireless network and the MU creating a connection with the DU. The use of a semi-autonomous USAR robot instead of a rescuer would automate the deployment of DUs and the creation of the wireless network. The system which will be used as the basis of this discussion is based on a system similar in concept to the EUROPicom model.

Figure 3.20 shows the initial wireless network for a USAR robot to be deployed in a disaster zone. This is made up of four nodes, a FFD acting as a PAN Coordinator, a RFD and two FFDs that are acting as routers. In an USAR scenario the PAN Coordinator would be connected to the Control Station (CS) while the RFD would be connected to the USAR robot. The two FFD acting as routers are deployed to provide additional communication paths between the CS and the USAR robot. The only node that is mobile in this USAR scenario is the RFD which is mounted on an USAR robot. Figure 3.20 also displays the transmission range of the FFDs, the transmission range is assumed to constant for all FFDs and have an omni-directional transmission pattern.
One of the reasons that an ad-hoc wireless system was chosen as the basis of the USAR robot guidance system was the existence of multiple communication paths between the PAN Coordinator and the RFD. This gives the USAR robot guidance system a high level of redundancy. Should a node fail then the wireless network can be reconfigured to reestablish a path between the PAN Coordinator and the RFD. Initially there are three possible connections between the CS and the USAR robot, these are shown as paths A, B and C in figure 3.20. Path C has one-hop whilst network paths A and B have two hops. AODV is used to select the least cost path, thus path C would initially be used as the primary communication over which the USAR robot guidance commands would be transmitted.

One of the operational parameters of a wireless guidance system is how many possible communication paths are available between the PAN Coordinator and the RFD. The larger the number of possible paths the greater the wireless system reliability will be. To provide the minimum level of redundancy two paths should be established and maintained.

As the RFD moves through the disaster zone it monitors the Received Signal Strength Indicator (RSSI) from its parent nodes. As discussed in section 3.1, the RSSI reading between network nodes gives an indication as to the integrity of the wireless
communication link between those two nodes, with figure 3.1 illustrating the link between the RSSI reading and the received signal strength in dBm. A series of decreasing RSSI readings is an indication that the wireless communication link is deteriorating. As different hardware solutions for 802.15.4 systems have different RSSI characteristics, before the wireless USAR robotic guidance system is deployed a series of tests would need to be conducted to determine the relationship between the RSSI and the quality of the received signal. The result of these tests would be an RSSI reading that would become a threshold for determining the quality of the received signal, i.e. if the current RSSI reading was below this predetermined threshold then the signal quality could not be guaranteed. Figure 3.21 shows the situation where the RSSI of the connection B1 has fallen below this predefined threshold.

![Diagram](image)

Figure 3.21 Wireless guidance system – path B1 failing

The deterioration in the quality of signal along connection B1 could be attributed to a number of causes and specific to the environment in which the wireless connection was operating. If there was a direct line of sight between the FFD and the RFD in figure 3.21 then the deteriorating wireless signal could be attributed to Free Space Path Loss (FSPL). In a disaster zone though there could be many obstacles between the two nodes. If this were the case then multi-path fading could be the cause of the deterioration. The obstacles between the two nodes could be caused by the path the USAR robot has taken, or instability within the disaster zone could cause debris to fall between the two nodes. Regardless of the cause of the deterioration in signal quality, it is important to be able to
determine a possible communication failure well in advance of the failure occurring to ensure the wireless robotic guidance system remain intact.

Having concluded that connection B1 has a deteriorating signal quality the course of action that needs to be taken must be determined. If the minimum level of redundancy is being provided, which would be two possible communication paths, then connection B1 would simply be allowed to fail, resulting in path B being removed from the system. If path B is allowed to fail then the resulting wireless network is shown in figure 3.22. Path A and C remain operational with path C being the primary path upon which guidance commands will be transmitted.

As the RFD continues to move away from the PAN Coordinator the RSSI of connection A1 or C will eventually fall below the predefined RSSI threshold. When this occurs a new FFD will need be activated and inserted into the path between the PAN Coordinator and the RFD. This new FFD is equivalent to a Drop Unit (DU) described in the EUROPCOM model. In this example let’s consider connection C as the path that has had a reduction in its RSSI, this is illustrated in figure 3.23. Therefore the FFD will be inserted in path C and will create connection C1 between the new FFD and the RFD.
The insertion of an FFD into the network between the PAN Coordinator and the RFD, or any subsequent insertion of a new FFD, requires the PAN Coordinator to control the sequence of events. This requires the PAN Coordinator to:

1. Instruct the node with the existing connection to the RFD to allow new nodes to join the network through them,
2. Detect that the new FFD has connected to the network,
3. Instruct the RFD to terminate its network connection, and
4. Instruct the new FFD to allow the RFD to rejoin the network through them.

The PAN Coordinator is responsible for all message routing through an 802.15.4 based wireless network. Therefore the PAN Coordinator will need to keep a Routing Table. This Routing Table will detail all the connected network nodes as well as routing details for each node. A new FFD that has joined the network will be acting as a router so the PAN Coordinator will still communicate directly with the RFD, i.e. the PAN Coordinator will specify the unique IEEE 64-bit MAC address of the RFD as the destination and the packet will be routed through the network as per the implemented routing algorithm.
After the successful insertion of the new FFD into the wireless network figure 3.23 shows the transmission coverage of the two FFDs that have direct connections to the RFD. The result of inserting the FFD into the wireless network is that there is a new network connection, C1, and there are now two possible paths, A and C, both having 2-hops. Based on the AODV routing the least cost path for the primary USAR robot guidance commands could be either path A or path C.

As the USAR robot continues to explore the disaster zone the RFD will continue to move away from the PAN Coordinator and deeper into the disaster area. As it does so the RSSI of connection A1 or C1 will eventually fall below the predefined threshold. Figure 3.24 shows the situation when it is the A1 connection that has started to reduce in signal strength. Again, a new FFD will be activated and be inserted into the path A creating the new network connection A2. Figure 3.24 again shows the deployment of the FFD and the transmission coverage of the two FFDs that have direct connections to the USAR robot/RFD. The result of the insertion of the new FFD into the network is a new network connection, A2. This means that path A becomes a 3-hop path whilst path C remains a 2-hop path. Based on the AODV routing the least cost path for the primary USAR robot guidance will now be path C.

As before, the RFD will continue to move deeper into the disaster area. As it does so the RSSI of connection A2 or C1 will eventually fall below the predefined threshold. Figure 3.24 shows the deployment of the FFD and the transmission coverage of the two FFDs that have direct connections to the USAR robot/RFD. The result of the insertion of the new FFD into the network is a new network connection, A2. This means that path A becomes a 3-hop path whilst path C remains a 2-hop path. Based on the AODV routing the least cost path for the primary USAR robot guidance will now be path C.
3.25 shows the situation when it is the C1 connection that has recorded a reduction in its RSSI reading. A new FFD will be activated and be inserted into the path C creating the new network connection C2. Figure 3.25 shows the deployment of the FFD and the transmission coverage of the two FFDs that have direct connections to the USAR RFD. The result is the insertion of the new FFD into the network and the creation of a new network connection, C2 which means that both path A and path C are 3-hop paths. As paths A and C have the same number of hops the AODV routing can select either path A or Path C as the primary path for USAR robot guidance commands.

![Figure 3.25 Wireless guidance system – path C1 failing, new FFD deployed](image)

Figure 3.26 shows the resulting network after three FFDs have been deployed as well as coverage provided by the two FFDs with direct connections to the USAR robot/RFD. The wireless network topology that has been created is shown if figure 3.27. The first 3 layers of the network is a traditional Tree topology with the Trees root node being the PAN Coordinator.

However, the fourth layer of the wireless network doesn’t meet the criteria to be classified as a pure Tree topology as the single fourth layer node creates a connection to two nodes in the third layer. Thus the entire wireless network structure cannot be
defined as a pure Tree topology, but does display all the Tree topology characteristics except at the lowest level.

![Diagram](image1)

**Figure 3.26 Wireless guidance system – result of 3 deployed FFDs**

Most importantly though is the wireless network keeps one of the advantages of the Tree topology, that being scalability. The USAR wireless guidance system grows as FFDs are deployed and therefore the scale of the network perfectly matches the required functionality.

![Diagram](image2)

**Figure 3.27 Wireless guidance system – Network Topology**
Figure 3.28 shows possible connections D1, D2 and D3 that could be made within the wireless network. If these connections were to be made then the resulting wireless network would be a partially connected MESH topology. This feature provides additional redundancy to the wireless network.

As mentioned previously, it is possible that in an USAR situation that an established connection could fail. This could be caused by a structure falling between two nodes creating multi-path fading or the failure of a node. If path C, defined as C – C1 – C2, was the current primary path being used for the USAR robot guidance commands and the connection C1 were to fail then the wireless network could self-heal by reconfiguring such that path C was redefined as C – D2 – D3 – C2. This would introduce an additional hop in the C path but would allow the connections C and C2 to be utilised. Section 4.3.2 details the algorithm that was developed to allow the wireless robotic guidance system to self-heal.

The transmission range of commercially available 802.15.4 based modules is in the range of 30m to 50m. As FFDs are deployed the wireless network expands as too does the geographical area which is covered by the wireless network. Figure 3.29 shows two...
wireless networks, the diagram on the left shows the initial area covered by the wireless network whilst the diagram on the right shows the area covered by the wireless network after three FFDs have been deployed. An ellipse has been fitted to each diagram to provide an estimate of wireless networks coverage. In the example presented within this section the deployment of three FFDs has approximately doubled the wireless networks coverage. Even though the transmission range of individual 802.15.4 nodes is relatively small, it is possible to obtain coverage of a large geographical area due to the manner in which the wireless network is created by the deployment of FFDs.

![Diagram showing network coverage comparison](image)

Figure 3.29 Wireless guidance system – comparison of network coverage

The diagram on the left of figure 3.30 shows an initial wireless network for an USAR robotic guidance system. The diagram on the right shows the final wireless network resulting from the discussion within this section. The final wireless network was created by ensuring that there were a minimum of two connections between the RFD and other network nodes. If the resulting network had been created to ensure that each node could create a connection to a least three other nodes then RSSI and TOF information could then be used in conjunction with trilateration calculations to accurately position the
nodes within the disaster zone. There would be significant benefits in creating a wireless guidance system that could accurately position nodes within a disaster area as this could reduce the time it could take rescuers to reach any victims that had been discovered. Section 5 of this thesis will discuss trilateration algorithms used for wireless networks before presenting simulations and localisation results obtained from tests conducted using the wireless robotic guidance system.

Figure 3.30 Wireless guidance system – comparison of initial to final
3.4 Summary

The initial sections of this chapter reviewed wireless communication systems that could possibly be used as a wireless robotic guidance system. It was concluded that an ad-hoc wireless communication system that was capable of creating MESH networks would be an ideal candidate. Wireless networks based on 802.11, Bluetooth and 802.15.4 were then evaluated and it was concluded that wireless networks based on 802.15.4 would provide an ideal platform for a wireless robotic guidance system.

The last section of this chapter then illustrated how an 802.15.4 based network could be deployed within a disaster zone. Figures 3.20 through to 3.26 demonstrated how a wireless network could be expanded by deploying FFDs strategically within a disaster zone. Once deployed the wireless network would then undergo a reconfiguration that would see the newly deployed FFD inserted into the wireless network such that the wireless networks coverage would be expanded. This increased network coverage would then allow the USAR robot to progress further into the disaster zone.

An 802.15.4 based network created in this manner would give the following advantages:

- Redundancy; the creation of a multi-path system gives the wireless network the ability to reconfigure should nodes fail or connections between nodes become inoperable.
- Scalability; the scheme used for the wireless network creation is such that additional nodes are only deployed when the wireless networks expansion requires. Thus the wireless network is matched to the operational conditions and can easily be scaled to cope with expanding areas.
- Network Coverage; the deployment of FFDs as required ensures that the area in which the USAR robot is operating has wireless network coverage.

The 802.15.4 specification ensures that hardware compliant modules are low-cost, energy efficient and have a sufficiently high data rate to send guidance commands to a USAR robot. Therefore a wireless network based on 802.15.4 is an ideal choice for a USAR robot guidance system which would be operating in a hostile environment. The
following chapter will present a conceptual design for a robotic guidance system that is capable of deploying FFDs to increase network coverage. The design of the wireless nodes will then be presented.
4 Wireless USAR Robot Guidance System

4.1 Introduction

The previous chapter discussed wireless communication systems that could possibly be used as wireless robotic guidance systems within an USAR environment. This discussion concluded that an 802.15.4 based wireless network would be an ideal solution. The previous chapter also described how the coverage of an 802.15.4 based network could be extended by deploying Full Function Devices (FFD) as an USAR robot searched the disaster zone. A wireless guidance system that is capable of achieving this functionality will be presented in the initial sections of this chapter. This will then be followed by a detailed presentation of the design of the wireless nodes that will be used within my robotic guidance system. This chapter will then conclude by presenting results from a series End-to-End delay tests to determine how the wireless guidance system might perform in an actual disaster zone.

The previous chapter described how the 802.15.4 network could be expanded in a generic manner using the terminology associated with the 802.15.4 standard. Figure 4.1 shows how the 802.15.4 model used in the previous chapter will map across to the USAR robotic guidance system that I have developed. The USAR robotic guidance system will consist of five node types, these being:

1. Control Node (CN),
2. Position Node (PN),
3. Drop Node (DN),
4. Mobile Node 1 (MN1), and
5. Mobile Node 2 (MN2).

Within this system the CN is the FFD which acts as the PAN Coordinator. All other node types are configured as FFDs and act as routers within the system. Having now
introduced the terminology for the USAR robotic guidance system the following section will present an overview of my USAR robotic guidance system.

![Diagram of USAR robotic guidance system](image)

**Figure 4.1 802.15.4 model to USAR robotic guidance system model**

### 4.2 Wireless USAR Robot Guidance System Overview

Two USAR robots were designed as part of this research, the designs for these have been presented previously in chapter 2. The first of these was a quadruped robot whilst the second was a tank-based robot. These two USAR robots were designed to work as a team within a disaster zone as their differing physical characteristics complemented each other when working together. The quadruped robot was designed to perform the initial reconnaissance within the disaster area, whilst the tank based robot was designed to follow when it was deemed safe to do so. Both robots though will use the same wireless robotic guidance system and be teleoperated by the same operator. A single Control Station (CS) will be used and from this CS both robots will be controlled and the status of their sensors, including image sensors, will be available to the operator.
Figure 4.2 shows an overview of the wireless robotic guidance system as it could be deployed at a disaster zone.

This figure shows the two wireless networks that are available to the USAR robots within the disaster zone, these networks are:

1. 802.11 – Video Network; this is used primarily for streaming video to the control station, and
2. 802.15.4 – Control Network; this is a lower speed network and is used for:
   (i) remote guidance of the USAR robot from the operator,
(ii) data acquisition, i.e. relaying readings from the USAR robot sensors to the operator,

(iii) providing a backup for the 802.11b/g video stream, i.e. should the 802.11b/g fail then images may be transferred to the control node via this network, and

(iv) localisation of the USAR robot.

The 802.11 video network was included in this wireless system because the processing systems within the USAR robots and the laptop used as part of the CS had this wireless network available. With this high bandwidth wireless network available it was logical to use this as the wireless network to provide a video stream from within the disaster zone. However, the 802.11 wireless network is just a point-to-many topology and will become inoperable at some stage as the USAR robots move deeper into the disaster zone. At this point the operator can command the USAR robots to send video data across the wireless network that is being used for the robotic guidance. The video stream will be reduced to image captures though as the maximum bandwidth of the wireless guidance system is 250kbps. This will provide the operator with some form of visual feedback from within the disaster zone.

The wireless guidance system consists of the following components:

1. Control Node (CN) which is incorporated within the Control Station (CS),
2. Position Nodes (PN),
3. Drop Nodes (DN), and
4. Two Mobile Nodes (MN) which are incorporated within the USAR robots,
   (i) Mobile Node 1 (MN1) is designated as the tank based USAR robot, and
   (ii) Mobile Node 2 (MN2) is designated as the quadruped USAR robot

The CS is positioned in a secure location that is a safe distance from the disaster zone. The CS consists of a laptop that has an inbuilt 802.11 interface. A CN is connected to the CS and acts as the access point for the CS to the wireless robotic guidance system.
Before the USAR robots enter the disaster area Position Nodes (PN) are deployed around the disaster zone. When deployed the PNs must be positioned such that they can communicate with the CN as they form an integral part of the initial wireless network. The PNs are designed to include a GPS module and can therefore accurately determine their physical location, this information will then be used when localising the USAR robots within the disaster zone. The algorithm used to determine the USAR robots position within the disaster zone is presented in the following chapter.

When the CN is initialised it will establish an association with the PNs and at this time the PNs will transfer their physical coordinates to the CN. The USAR must then be prepared to enter the disaster zone. When initialised the tank based USAR robots MN, MN1, will create an association with the CN. The quadruped USAR robots MN, MN2, will then create an association with tank based USAR robots MN. The initial configuration of the USAR wireless guidance system is shown in figure 4.3.

![Figure 4.3 Initial wireless USAR robot guidance system configuration](image)

Figure 4.3 shows the wireless connection between MN1 and MN2. It is essential that this connection remain operational as its failure would result in the quadruped robot being unable to receive guidance commands from the CS. Such a scenario could result in the loss of the quadruped robot. The Received Signal Strength Indicator (RSSI) measurement of this connection will be used prevent this situation from occurring.

The RSSI measurement will be transferred from MN2 to the CS where it will be monitored by the operator. It will be the operators responsibility to ensure that this communication link remains intact. The operator will achieve this by ensuring that both
robots remain physically close enough within the disaster zone to ensure a reliable communication link. Future research could be performed to implement an algorithm within the tank based robot so that it automatically follows the quadruped robot through the disaster zone. This would represent many challenges as the tank based robot would need to operate autonomously using onboard sensors to perceive its surroundings and ultimately determine the optimal path to follow the quadruped robot.

As discussed in the later section of the previous chapter as MN2, the tank based robot, moves deeper into the disaster zone its wireless communication connections to its parent nodes will deteriorate. This can be caused by either Free Space Path Loss (FSPL) or multi-path fading. The RSSI will again be used to determine when a connection between MN2 and one of its parents may fail. When a series of decreasing RSSI measurements are received that are below the predefined operating threshold a Drop Node (DN) will be deployed from the tank based robot. The DN will be inserted into the wireless robotic guidance system between MN1 and its parent node. This will ensure that the wireless guidance system remains intact and will allow the tank based robot to travel further into the disaster zone. The mechanism designed to deploy DNs was presented in section 2.3 of this thesis whilst the algorithms used reconfigure the wireless network after a DN has been deployed are presented in section 4.3 of this thesis.

As discussed previously it is possible to use Received Signal Strength Indication (RSSI) and/or Time of Flight (TOF) measurements to determine the transmission distance between two wireless nodes. Therefore, if a wireless node can communicate with a minimum of two other wireless nodes then it is possible to use trilateration based on transmission distances to determine the nodes location within the disaster zone. This means that MN1 can then use the RSSI and/or TOF measurements information from its parents nodes to accurately calculate its position.

As the PNs supply accurate physical locations from their GPS modules it is possible to use these as reference positions to create accurate positioning for MN1 with latitude and longitude coordinates. As the physical position of MN1 can be determined it is possible to record the location where DNs are deployed, thus creating additional physical reference positions. Figure 4.4 illustrates how accurate positioning is achieved by my
wireless guidance system. The network shown in *green* demonstrates how it is possible to obtain an accurate position for a DN when trilateration of distances derived from RSSI and/or TOF measurements from PNs are used. The network shown in *red* demonstrates how deployed DNs can then be used for localising MN1.

![Figure 4.4 Localisation via trilateration using RSSI and TOF](image)

An algorithm to determine locations within the wireless robotic guidance system based on RSSI and/or TOF measurements is presented in chapter 5 of this thesis along with simulation and test results.

The remainder of the chapter will present the design of the wireless node modules. The algorithms implemented within the wireless network to reconfigure when DNs are deployed and self-heal when node or communication failures occur are also discussed as are the algorithms implemented within each node type.
4.3  Wireless Node Hardware Design

4.3.1 Wireless Node Hardware

There are three possible options that can be considered when designing the hardware for an 802.15.4 based wireless system, these are to build a solution around:

1. an 802.15.4 compliant Transceiver,
2. an 802.15.4 System on Chip (SoC) integrated circuit that incorporates an 802.15.4 compliant transceiver and a microcontroller, or
3. a dedicated 802.15.4 Module that incorporates the transceiver, microcontroller, RAM, FLASH, antenna and built in operating system.

Each option has advantages and disadvantages. Building a system around an 802.15.4 transceiver [65] requires a large number of additional components as well as adherence to strict printed circuit board (PCB) design guidelines for RF circuitry. 802.15.4 modules are significantly more expensive than transceivers, in some cases by a factor of ten, but can significantly save design time and PCB development costs.

Whilst a number of options were considered the Jennic JN5148-001-Myy module was selected as the basis for the 802.15.4 wireless network nodes for two reasons. Firstly the module allowed a simplified PCB design to be implemented. Secondly the JN5148 has a ranging engine designed within its wireless baseband circuitry which allows Time Of Flight (TOF) measurements to be recorded. As previously discussed in section 2.1 the TOF measurement capability, in conjunction with Received Signal Strength Indication (RSSI) measurements, allows the JN5148 to determine its distance from the transmitting device.

These transmission distances can then be used in localisation algorithms which will allow the nodes to be accurately positioned within the disaster area. Trilateration of RSSI and TOF measurements is discussed in detail in Chapter 5 where localisation
algorithms are discussed and simulated before results of the trilateration tests using the JN5148 are presented.

Figure 4.5 shows a block diagram of the JN5148 microcontroller. This microcontroller not only contains a 32-bit RISC CPU but also includes an 802.15.4 compliant 2.4GHz transceiver which includes a “Medium Access Control (MAC) accelerator with packet formatting, Cyclic Redundancy Check’s (CRC), address checking, automatic acknowledgements and timers” [66]. The JN5148’s transceiver has a transmission power of +2.5dBm and a receiver sensitivity of -95dBm which gives this a transmission range of up to 1km, but these distances are typically 30m~50m in an indoor environment.

![JN5148 Wireless microcontroller block diagram](Ref:[66])

The JN5148 is classified as an ultra low power device as the microcontroller draws a maximum of 10.6mA and the transceiver draws 15mA when transmitting data and 17.5mA when receiving data. The JN5148 has a number of low power modes that can be utilised when beacon-based communication is implemented. This allows the JN5148 to conserve battery consumption, with the current drawn being as low as 0.12µA when in deep sleep mode.

The JN5148-001-Myy module can be supplied in one of the following configurations:

1. JN5148-001-M01, antenna is integrated in the modules PCB,
2. JN5148-001-M03, a uFI connector is provided to allow the connection of an external antenna, and
3. JN5148-001-M04, provides a Low Noise Amplifier (LNA) and a Power Amplifier (PA) as well as a uFI connector to allow connection of an external antenna.

Figure 4.6 shows the block diagram of the JN5148-001-Myy based node, a complete schematic of the 802.15.4 wireless node can be found in Appendix F. Figure 4.7 shows the prototype wireless node module which can configured as either a CN, PN, DN or MN. The PN uses an additional Data Acquisition (DAQ) board, accessible from the I/O expansion connectors, to interface with the GPS module. A 4-way DIP switch is provided within the design, this is read on power up and is used by the node during initialisation to determine what type of node it will be required to operate as. The feature allows for a single board design to be used for all node types and as well as a single software created binary file.

The 2 UARTs from the JN5148 are connected to a FT2232D dual USB UART/FIFO which allows the node to be easily connected to a Windows based PC. Windows drivers are available to create a Virtual COM Port (VCP) for each of the JN5148-001-Myy’s
UARTs. The ability to have 2 VCPs on the same USB cable allows a programming channel on UART0 and a debugging channel on UART1 to co-exist within a signal USB connection and simplifies the development process. To comply with the programming techniques on Jennic's DR1048 development boards a 6-pin header strip is provided so that the JN5148 can be programmed in a similar manner as is done in the development system.

![Image of JN5148-001-M03 Wireless microcontroller module block diagram](image)

**Figure 4.7 JN5148-001-M03 Wireless microcontroller module block diagram**

The node may be powered by the three AAA 1.2V batteries, a USB cable or a power supply via a standard 2.5mm connector. All power supplies are passed to a LP2985-3.3 Ultra Low-Dropout regulator to ensure that the 3.3V supply required by the JN5148 is guaranteed. A 1 Kbit i²c EEPROM is also provided for storing any required non-volatile data.

The wireless node was designed to be a general purpose board and as such all the I/O signals from the JN5148-001-Myy are brought out to two header strips which allow for an additional DAQ board to be connected. The DAQ board acts as the interface to the Position Nodes (PN) GPS module. The GPS module used is a Fastrax IT321 which is mounted on an application board that has the ability to accept a 1575R-A GPS Internal Active Antenna. In addition to the GPS interface all digital and analog I/O signals on the DAQ board are buffered and made available via screw terminals should additional sensors be needed during future research.
4.3.2 Wireless Node Software

Jennic provide three API’s to program their microcontrollers, these being 802.15.4, JenNet and ZigBee PRO. The wireless USAR robot guidance system is programmed with the JenNet API. The 802.15.4 API was the preferred choice as this would have provided greater control of route creation and determination, but the Jennic 802.15.4 API only allows point-to-point and star network topologies which were unsuitable for this application. Figure 4.8 shows the JenNet protocol stack.

The JenNet layer adds to the 802.15.4 MAC and PHY layers by providing functionality at the Network (NWK) Layer, assisting with the network formation and routing services. The Application interface to the JenNet layer is bi-directional, user
applications communicate with the JenNet layer using predefined ‘application to stack’ functions whilst ‘stack to application’ or callback functions are used to exchange information from the JenNet layer to the application layer. Figure 4.9 shows an overview of how the user application code interacts with the Jenie stack [67].

Function `vJenie_CbConfigureNetwork()` is defined within the user application, this function is called by the Jenie Stack at startup and is used to define a number of network parameters essential to the formation of the network, the key parameters being:
1. PAN ID,
2. Network Application ID,
3. Networks radio frequency channel, and
4. Allocation of the Routing Table for the Coordinator.

Once the network parameters have been specified the function \textit{vJenie\_CbInit()} is then called by the Jenie Stack. This function allows the user application to configure the devices peripheral interfaces and initialise any user defined variables and data structures. \textit{eJenie\_Start(NODE\_TYPE)} is called as the last instruction within \textit{vJenie\_CbInit()} and starts the device as the required node type, i.e. either Coordinator, Router or End Device. Once the node is operational a cyclic main loop, \textit{vJenie\_CbMain()}, is executed as in normal microcontroller applications.

The Jenie stack provides three event driven callback functions that provide indication to the main loop when an interrupt has occurred. Each callback function returns an enumerated data type to indicate which event has triggered the callback function. The callback functions are defined as:

1. \textit{vJenie\_CbStackMgmtEvent()},
2. \textit{vJenie\_CbStackDataEvent()}, and
3. \textit{vJenie\_CbHwEvent()}.

Network events, such as \textit{NETWORK\_UP}, \textit{STACK\_RESET}, \textit{CHILD\_JOINED} and \textit{CHILD\_LEAVE} are handled by \textit{vJenie\_CbStackMgmtEvent()}. Events that are associated with data transfer between nodes, such as \textit{E\_JENIE\_DATA} and \textit{E\_JENIE\_DATA\_ACK}, are handled by \textit{vJenie\_CbStackDataEvent()}. While \textit{vJenie\_CbHwEvent()} looks after any interrupts that may occur from the devices peripheral inputs and outputs.

Routing within the JenNet network is handled by the coordinator node and as such the Routing Table is defined and resides within this node. Each entry within the Routing
Table defines the network nodes unique 64-bit MAC address as well as the 64-bit MAC address of the node which is the first hop from the coordinator, i.e. the node to which packets for the destination node must initially be sent.

Both coordinator and routers within the JenNet network require a Neighbourhood Table. The Neighbourhood Table is used to define nodes that are directly connected to that node, i.e. its child nodes. Entries in the Neighbourhood Table specify the child nodes unique 64-bit MAC address as well as statistics relating to the number of packets sent and the number of missed packets by that node.

Thus the Routing Table within the coordinator and the Neighbourhood Tables within the coordinator and the routers define the configuration of the JenNet network. The Neighbourhood Table keeps information identifying the coordinators and routers child nodes whilst the Routing Table within the coordinator contains an entry for all nodes except those nodes which are direct children of the coordinator itself.

The JenNet layer provides all the functionality required for servicing changes to the networks structure. The main events that are detected by vJenie_CbStackMgmtEvent() are:

1. **NETWORK_UP**: network matching the defined network parameters is available,
2. **STACK_RESET**: network is no longer available,
3. **CHILD_JOINED**: a child node has joined the current node, and
4. **CHILD_LEAVE**: a child node has left the current node.

These four network events enable the wireless USAR robot guidance system to grow as Drop Nodes (DN) are deployed and self-heal as network nodes fail or operational conditions within the disaster zone change in such a way as to effect the ability for nodes to wirelessly communicate. The algorithms used to allow nodes to be added to and leave the wireless USAR robot guidance system are presented in the following section.
4.3.3 Wireless Node Algorithms

Figure 4.10 shows an example of the wireless network structure that the wireless guidance system is capable of creating. Within this wireless network the nodes I have designed are designated the following roles:

- Control Node (CN): Coordinator,
- Position Node (PN): Router,
- Drop Node (DN), Router,
- Mobile Node(s) (MN), Router.

![Figure 4.10 Wireless USAR robot guidance system structure](image)

Whilst the topology of the network suggests it may be possible to configure Mobile Node 2 (MN2) as an End Device this may limit the ability of the wireless network to recover should there be a node failure or a communication channel between nodes become inoperable. Should MN1 and MN2 become orphaned from the wireless network it may be possible for MN2 to rejoin the network when MN1 cannot. In this scenario if MN2 were a router then MN1 could connect to MN2 and thus reconnect to the entire wireless network.
4.3.3.1 Wireless Network Expansion

As described in an earlier section of this chapter, as the USAR robot moves through the disaster zone, MN1 will monitor the Received Signal Strength Indicator (RSSI) between itself and its parent node. When this RSSI value fails below a predefined threshold it is deemed that the wireless signal strength has reduced to a level that may see the communication link, and thus the wireless USAR robot guidance system, fail. When this occurs MN1 will send a request to the CN requesting permission to perform a DN deployment sequence.

Therefore, the wireless USAR robot guidance system expands by deploying Drop Nodes (DN) as the USAR robot moves through a disaster area. Section 2.3.1.1 presented the mechanical design of the Drop Node Deployment System that was developed as part of this research, this section will discuss the DN deployment sequence from a wireless network perspective. Figure 4.11 illustrates the sequence of events that must occur when a new DN is deployed.

Figure 4.11 (a) shows the situation where MN1 has been given permission by the CN to perform a DN deployment. Here a DN has been deployed but is yet to establish a connection to any other network nodes. Having given permission for a new DN to be deployed the CN will change its permissions and allow additional nodes to request to join its network. The DN will then attempt to connect to the network via the coordinator as shown in Figure 4.11 (b). Once the DN establishes a connection to the CN the CN will send a command to MN1 requesting that it terminates its connection and reestablish this with the new DN as shown in figure 4.11 (c). Finally, the new DN will have its permission set to accept requests for nodes to join the network. Figure 4.11 (d) shows the final result when MN1 has rejoined the network via the new DN.
Figure 4.11 illustrates the sequence of events required to facilitate a DN being inserted between a CN and MN1, a flowchart detailing the algorithm is presented in figure 4.12. DNs can also be inserted into the wireless network between existing DNs and MN1, figure 4.13 displays the algorithm for this scenario. Both figure 4.12 and 4.13 indicate when nodes are added to and removed from the Routing Table and the Neighbourhood Tables.

The CN, or the wireless network coordinator, is not informed via the callback function VJenie_ChStackMgmtEvent() when a child joins a DN, i.e. a router. To determine when a child has joined or left a DN the CN must constantly scan the Routing Table to look for changes in the network configuration. This process is complicated by the fact that the Routing Table isn’t maintained by the JenNet API functions, i.e. old or redundant node entries are not automatically removed from the Routing Table, only the flag indicating the entry is VALID is cleared. This means that the whole Routing Table must be interrogated each scan to determine the status of each entries VALID flag.
Ad-Hoc Wireless MESH Network Implementation for Rescue Robots

Figure 4.12 Drop Node Deployment Sequence – Control Node
Once the DN has successfully been deployed the CN continues to send movement commands to the MNs and the MNs continue to send sensor data to the CN as an acknowledgement that the command has been received. The CN will continue to send the command packets to the 64-bit addresses of the MNs, the DNs act as routers and pass these packets through without any additional processing. Therefore, as all routing is performed via the JenNet stack it is done as quickly as possible as there is no need for the user application to perform any routing based operations.
4.3.3.2 Wireless Network Recovery

As the USAR robot will be operating in a hostile environment it is important that the wireless USAR robot guidance system be capable of handling situations were network nodes fail or become unavailable. There are many reasons that could cause a network node to fail, these could range from a simple component failure on the nodes PCB to the node being crushed by disturbed debris. The wireless communication channel can also be affected by moving debris, causing highly variable radio propagation conditions making wireless communications unreliable.

Figure 4.14 illustrates the scenario where a DN fails. Figure 4.14 (a) shows the intact wireless network before the DN 2 fails. Figure 4.14 (b) shows the network just after the DN failure which results in three network connections, C1, C2 and D3, becoming inoperable. In this example the primary wireless communication link that was being used for the USAR robot guidance system was C1 and C2, therefore it is important that a connection be reestablished between MN2 and the DN 1. Figure 4.14 (c) shows the resulting network connections once the communication link C has been established. Figure 4.15 details the algorithms used by the nodes affected by the failure of the DN.

![Figure 4.14 Drop Node Failure – Network Perspective](image)
In the situation shown in Figure 4.14 DN2 has failed, this failure will be detected differently by the two effected nodes, DN1 and MN1. DN1 will receive an interrupt via the callback function `vJenie_CbStackMgmtEvent()` informing it that there has been a `CHILD_LEAVE` event. DN1 will then reassert its permission flag to allow the orphaned node to connect. MN1 will determine that DN2 has failed when a preset number of packet failures have been detected, this preset number can be defined in the `vJenie_CbConfigureNetwork()` callback function which is executed on node startup. Once MN1 has determined that its parent is no longer available it will reset and go back to looking for a network to which it can connect. As the parent of the failed DN is now accepting new children, MN2 will establish a connection with DN1.
4.3.3.3 Control Node

The Control Node (CN) is the coordinator within the 802.15.4 based network, the algorithm implemented within the CN is shown in figure 4.16.

![Figure 4.16 Control Node State Machine](image)

Figure 4.16 Control Node State Machine

When the CN is powered up it performs a series of initialisations that define the Network ID, the PAN ID and the preferred RF channels before creating the network.
Once the network has been created the CN will then enter the System Initialisation mode. In this mode it will wait until the required network nodes have been detected and added to the network. The initial system configuration for the wireless USAR robot guidance system is shown in figure 4.17 and consists of at least two Position nodes (PN) and two Mobile Nodes (MN).

![Initial wireless USAR robot guidance system configuration](image)

Figure 4.17 Initial wireless USAR robot guidance system configuration

When all required nodes have been detected and connected to the network the CN enters its Standard Operational mode. In this mode it receives commands from the Control Station and sends these commands to the required nodes within the wireless network. These commands will consist mainly of positioning commands to the USAR robots. The USAR robots will acknowledge these commands and within the acknowledge packet will return the readings from the onboard sensors. The structure of both the command and acknowledge packets were presented in section 2.3.1.2 and 2.3.2.2 when the designs of the USAR robots were discussed.

When MN2 determines the RSSI between itself and its parent node has dropped below the predefined threshold it will send a request to the CN requesting that the CN initiate a DN deployment sequence. When the CN receives this request it will enter the Drop Node Deployment mode. If this is the first DN to be deployed then the sequence of events that need to be performed is detailed in figure 4.12. If this is not the first DN deployment then the CN must monitor the size of the Routing Table to determine when the new DN has been added to the network successfully. At the completion of the Drop Node Deployment mode the CN returns to its Standard Operational mode until the next DN deployment is required.
4.3.3.4 Position Node

The Position Node (PN) is configured as a router within the 802.15.4 based network, the algorithm implemented within the PN is shown in figure 4.18.

The algorithm implemented within a PN is quite simple. On power-up the PN will read the coordinates from the connected GPS device. Then when the PN associates with the CN the GPS coordinates will be transferred to the CN, which will then in turn transfer these coordinates to the CS to be used in the localisation algorithm. At regular time periods the CN will request that the PN send a PING packet to MB2 so that RSSI and TOF measurements can be made, these measurements will again be transferred to the CN so they can be relayed to the Control Station. The PN can be used as a router within the JenNet network, but as the relaying of packets is handled within the JenNet layer of the protocol stack there is no impact on the PNs algorithm.
4.3.3.5 Drop Node

The Drop Node (DN) is configured as a router within the 802.15.4 based network, the algorithm implemented within the DN is shown in figure 4.19.

Figure 4.19 Drop Node State Machine
After initialisation the DN will wait to join the network. The DN can join the network via the CN, if it is the first drop node deployed, or via another DN. After joining the network the DN will then wait for MN1 to create an association at which point the DN will then enter its Operational Mode.

Whilst there is an association between the DN and MN1 the DN will monitor the RSSI measurement between the two network nodes. When the DN detects that the RSSI has dropped below a predefined threshold it will send a message to the CN informing the CN that the connection between itself and the MN is deteriorating and may become inoperable. The CN will then send an acknowledge packet to inform the DN that a DN deployment sequence is to occur.

At the start of a DN deployment, the DN will modify its permission setting to allow additional nodes to join the network. The DN will then wait for the new DN to join the network. The addition of the new DN will be detected by the CN by an increase in size of its Routing Table. Once the CN has added the new DN to the routing table the CN will send a message to the DN permitting it to terminate the association between itself and MN1, MN1 will then create an association with the new DN and thus rejoin the network.

Once the DN deployment sequence has been completed the DN returns to its Operational Mode. After the DN deployment the DNs purpose within the network is to route messages between the CN and MN1. As the routing is performed by the JenNet layer the DN algorithm at this point simply remains in an infinite loop whilst the network is healthy.

4.3.3.6 Mobile Node 1 (Tank-based USAR robot)

Mobile Node 1 (MN1) is mounted within the tank-based USAR robot and is configured as a router within the 802.15.4 based network. The algorithm implemented within MN1 is shown in figure 4.20.
After initialisation MN1 will wait to join the network. If MN2 has not already created an association with MN1 then MN1 will wait for it to do so. When associated with MN2, MN1 will enter its Operational Mode where it will:

1. Receive movement commands from the CN,
2. Send sensor readings to the CN, and
3. Monitor the RSSI between itself and MN2, these reading will be sent to the CN so they can be monitored by the operator at the Control Station.

When a DN deployment sequence has been initiated MN1 will be requested to leave and then rejoin the network. The leaving and rejoining of the network is coordinated by the
CN in such a way that when MN1 rejoins the network it will be to the newly deployed DN.

4.3.3.7 Mobile Node 2 (Quadruped USAR robot)

Mobile Node 2 (MN2) is mounted within the quadruped USAR robot and is configured as a router within the 802.15.4 based network. The algorithm implemented within MN2 is shown in figure 4.21.

The algorithm implemented within MN2 is quite simple. After initialisation MN2 will create an association with MN1 which will remain intact for the duration of the USAR robot deployment. When associated with MN1, and as such part of the network, MN2 will enter its Operational Mode where it will receive movement commands from the CN. These CN commands will be acted upon and acknowledged, the acknowledgement packet will also contain all the sensor readings from the quadruped USAR robot.
4.4 Wireless USAR Robot Guidance System Test Results

To test the performance of the wireless robotic guidance system an End-To-End (ETE) delay test was performed. The ETE delay is defined as the time that is taken for a packet to be transmitted through a network from the source node to the destination node. As the number of routers within a wireless network increases so should the ETE delay. This is due to the time it takes a router to:

- (i) receive and process a packet,
- (ii) determine which child the packet must be forwarded to, and
- (iii) retransmit the packet towards its destination node.

Figure 4.22 shows an overview of the configuration that was used to perform the ETE delay test. Software within the Control Node (CN) was modified to transmit a data packet to the last node in the wireless guidance system when a pushbutton was pressed. For this test the CN is able to determine the last node in the wireless network by using the last entry within the Routing Table, this is only valid because of the way the wireless network is formed. When a new Drop Node (DN) joins the network its parent node will stop allowing association with new nodes, thus the next new node must create an association with the last DN to have joined the wireless network. This creates a tree topology where each parent only has one child.
Our ETE delay test was conducted in the hallway on the 1st floor of the Electrical Engineering (EE) laboratory, this is the same hallway used for the RSSI and TOF distance based measurements which are presented in the next chapter. The hallway used is approximately 40m, so whilst performing this test there was a direct line of sight between transmitters and receivers. Because this test was done in near ideal conditions no packets were lost during our tests, so the time taken by the router to retransmit a packet was able to be determined.

An 802.15.4 Packet Sniffer was used to determine the ETE delay, this was achieved using laptop, Texas Instruments Packet Sniffer (v 2.12.3) software and a ChipCon CC2400 Evaluation board [68]. The CN was connected serially to the same laptop so diagnostic information could be displayed during the test.

The wireless nodes were initially turned off and aligned in a straight line down the middle of the hallway, separated by a distance of 5m. The initial test was conducted between the CN and the first DN. Ten transmissions from the CN were initiated and recorded by the Packet Sniffer. The next DN was then turned on, after this DN had associated to the previous DN ten transmission were initiated and recorded. This procedure was repeated until the wireless network consisted of eight wireless nodes, a CN and seven DNs. The results obtained are shown in table 4.1 and graphically displayed in figure

<table>
<thead>
<tr>
<th>Network Depth</th>
<th>End-to-end Delay (ms)</th>
<th>Standard Deviation (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.484</td>
<td>0.614</td>
</tr>
<tr>
<td>2</td>
<td>7.071</td>
<td>2.044</td>
</tr>
<tr>
<td>3</td>
<td>11.317</td>
<td>2.747</td>
</tr>
<tr>
<td>4</td>
<td>14.898</td>
<td>1.862</td>
</tr>
<tr>
<td>5</td>
<td>18.735</td>
<td>3.153</td>
</tr>
<tr>
<td>6</td>
<td>23.730</td>
<td>3.896</td>
</tr>
<tr>
<td>7</td>
<td>27.325</td>
<td>2.101</td>
</tr>
</tbody>
</table>

Table 4.1 End-to-End Delay test results

The results show there is a linear relationship between the depth of our wireless robotic guidance system and the recorded ETE delay. The ETE delay increased by an average
of 4.12ms for each additional wireless node added to the wireless guidance system. The reason the ETE delay within this wireless robotic guidance system is small is due to the fact that routing within a JenNet based network is performed at the Network Layer. Based on these results, each time a DN is deployed by the wireless robotic guidance system data will be delayed to the USAR robots MN by an additional 4.12ms.

![Wireless Robot Guidance System End to End Delay](image)

Figure 4.23 End-to-End Delay test results

Figure 4.24 shows a screen capture of the Texas Instruments Packet Sniffer application. As can be seen from this figure all the data within the 802.15.4 packets is available to be displayed. The lower pane of this window shows the traffic flow through the wireless robotic guidance system. The saw-tooth waveform is indicative of data being sent from the CN to the last DN cyclically.

Figure 4.25 has been included to highlight the use of the Routing Table and the Neighbourhood table within the JenNet based network. The screen capture shown in this figure provides a snapshot of the CNs Routing Table and Neighbourhood table when seven DNs are connected to the wireless robotic guidance system. Here it shows there are 6 entries within the Routing Table and one entry in the Neighbourhood Table, this being the CNs child.
Ad-Hoc Wireless MESH Network Implementation for Rescue Robots

Figure 4.24 Packet Sniffer screen capture

Figure 4.25 Control Node diagnostics Screen Capture
4.5 Summary

Establishing wireless communications to USAR robots searching through collapsed building and rubble presents a number of challenges. Generally there are obstacles between the USAR robot and the Control Station which can attenuate, and in some cases block, wireless signals between the two. To prove successful in these types of environments a wireless guidance system must be capable of adapting to the operational conditions in which the USAR robot is to operate.

The 802.15.4 based wireless guidance system design presented at the beginning of this chapter was able to adapt to the operating conditions by deploying Drop Nodes when the USAR robot detected that the wireless network was deteriorating. Once a DN had been deployed the wireless network would reconfigure with the DN being inserted into the wireless network. This ensured the wireless network remained intact as well as ensuring that the network coverage was extended to allow the USAR robot to continue deeper into the disaster zone.

The next section within this chapter then presented the design of the 802.15.4 wireless nodes and discussed the algorithms used at a network level and in the actual wireless nodes. The wireless nodes were based on a JN5148 module from Jennic which incorporated both a 32-bit microcontroller and an 802.15.4 compliant transceiver. The hardware design of the wireless nodes was presented which was then followed by a description of JenNet and the Application Programming Interface (API) used to create the wireless network. This description detailed how the API interfaced with the protocol stack to determine changes in the networks configuration. Network level algorithms for detecting and adding Drop Nodes as well as network recovery from the loss of a Drop Node were then presented and discussed. To conclude the chapter algorithms, in the form of state machines, for each of the five wireless node types were then presented.

This chapters final section then presented the results from an End-to-End delay test. These test results indicated that each additional Drop Node (DN) added to the wireless guidance system would add an average delay of 4.12ms to end-to-end delay between the Control Node (CN) and the Mobile Node (MN). This average delay of 4.12ms was
achieved in an ideal environment, i.e. direct line of sight over a 5m distance. The use of the packet sniffer also verified there were no packet retransmissions, so the result achieved here is a best case scenario and this delay could increase when the wireless guidance system was deployed within an actual disaster area.

Having developed an 802.15.4 wireless robotic guidance system the following chapter will now discuss how the wireless guidance system can be used to determine an accurate position for the USAR robot within a disaster zone.
5 Localisation

5.1 Introduction

The wireless USAR robotic guidance system presented in the previous section of this thesis details how an ad-hoc wireless network can be created within a disaster zone by deploying Drop Nodes (DN) to extend the coverage of the wireless network. My wireless USAR robotic guidance system also utilises Position Nodes (PN). PNs not only provide additional network paths within the wireless network, they also acquire GPS coordinates when they are initially deployed. Having nodes with fixed physical references, such as PNs, allows the USAR robots Mobile Node (MN) to determine their physical locations via trilateration within the disaster zone. As the MN location is known when DNs are deployed they too can be assigned physical locations. This allows DN locations to be used as reference positions in future trilateration calculations, allowing the position of the USAR robot to be determined as it moves deeper into the disaster area. Therefore, as well as providing a wireless USAR robot guidance system my system is capable of localising the USAR robots accurately within a disaster zone.

This chapter will begin by discussing some of the localisation techniques that are currently used in wireless networks. The mathematical concepts behind trilateration are then presented. The algorithm upon which the localisation technique is based, ‘Trilateration using Clustering and Centroids’ [69], is then discussed. Simulation results based on this localisation algorithm are then presented before localisation results from tests conducted within the wireless robotic guidance system are presented and discussed.

The ability for an USAR robot to accurately determine its position within a disaster area could significantly reduce the time required for rescuers to extract survivors. But localisation within a disaster zone presents a number of unique challenges. Disasters such as earthquakes and cyclones can cause significant damage to buildings and infrastructure. Thus the environment in which an USAR robot operates is generally
unstructured, i.e. it cannot be guaranteed that landmarks that could have been used as position references will remain in their normal positions.

An ideal candidate for localisation of an USAR robot is the Global Positioning System (GPS). However many USAR robots are designed to search within damaged and collapsed buildings, situations that resemble an indoor environment. Thus the Global Positioning System (GPS) cannot be used for localisation for all USAR deployments as reception of the satellites positioning signals, L1 1.57542GHz and L2 1.2276 GHz, cannot be guaranteed in an indoor environment.

One method of localisation that may be used for USAR robots is via a wireless communication network. A significant volume of research has been conducted in the field of localisation within Wireless Sensor Networks (WSN) and an analogy can be drawn between this research and the issues associated with localisation via a wireless robotic guidance system.

Localisation based on wireless networks can be categorised as either range-free or range-based. Range-free localisation use techniques that estimate nodes positions via network traffic and routing information. Paper [70] presents a range-free localisation technique for WSNs known as DV-HOP. Using this technique a wireless node estimates its distance from an anchor node, i.e. a node that knows it fixed location, by multiplying the hop-count and the hop-distance. Once a series of distances are estimated then trilateration is used to estimate the nodes position. Other examples of range-free localisation techniques are Monte Carlo Localisation (MCL) [71], APIT [72], and Distributed Range-free Localisation DRL [73].

Range-based localisation techniques use physically measured parameters to determine the distance between nodes within a wireless network. Trilateration techniques then use these distances to determine the nodes location. The physically measured parameters available within a wireless network that can be measured consist of Received Signal Strength Indicator (RSSI), Link Quality Index (LQI), Time of Flight (TOF) and Angle of Arrival (AoA). In [74] and [75] localisation schemes within WSNs using RSSI and LQI are presented. Both schemes collect RSSI and LQI data from their network nodes
then use different algorithms to determine the locations within the WSN, [74] uses a Support Vector Regression (SVR) algorithm whilst [75] uses a Weighted Centroid Localisation (WCL) algorithm.

Localisation within wireless robotic guidance systems has also been heavily researched. In [76] and [77] RSSI readings from an 802.15.4 based network are used in conjunction with other sensors to determine an USAR robots location within a disaster zone. The system described in [76] is conceptually similar to the one that I have developed for this thesis, in that an 802.11g network is provided for video and audio transfer between the USAR robot and a control station whilst a ZigBee network has been provided for the robot guidance and localisation. A digital compass is also used in this project so that the USAR robots position can be defined in terms of \((x_o, y_o, \theta_o)\) where \(\theta_o\) defines the angle the USAR is traveling. The system described in [77] also uses the RSSI readings from a ZigBee network for USAR robot localisation. A digital compass and encoders mounted on the motors provide additional location specific information. These three parameters are then feed into a software system, referred to as a Suppression Modulator, which gives a weight to all location monitoring sensors based on perceived accuracy before determining the USAR robots position.

Both these systems only use RSSI information as an input parameter from the wireless network to the localisation algorithm, whilst the system that I have developed may use both RSSI and TOF. Thus the system is a range-based localisation system that can use RSSI and/or TOF to determine distances before using trilateration techniques to determine the USAR robots location.

Trilateration is a process used to determine the location of a point in space \((x_o, y_o, z_o)\) when the distances from fixed points are known. Figure 5.1 shows the situation where the location of the point \(N\) is trying to be determined from the points \(A, B\) and \(C\). In terms of an USAR robot scenario, \(N\) is the USAR robot that is mobile through the disaster zone whilst \(A, B\) and \(C\) are nodes whose coordinates are known. The radii of the circles \(A, B\) and \(C\) are equivalent to distances derived from the RSSI and/or TOF readings between nodes \(A, B, C\) and \(N\). Thus, if the locations of \(A, B, C\) and the distances \(A_r, B_r, C_r\) are known then it is possible to calculate the location of \(N\).
Using the example presented in figure 5.1 and considering only two dimensions \((x, y)\) the distances from \(N\) to \(A\), \(B\) and \(C\) can be represented by the following series of equations:

\[
\begin{align*}
A_r &= \sqrt{(x_n - x_a)^2 + (y_n - y_a)^2} \\
B_r &= \sqrt{(x_n - x_b)^2 + (y_n - y_b)^2} \\
C_r &= \sqrt{(x_n - x_c)^2 + (y_n - y_c)^2}
\end{align*}
\]

If these equations are rearranged such that they all equal zero:

\[
\begin{align*}
(x_n - x_a)^2 + (y_n - y_a)^2 + A_r^2 &= 0 \\
(x_n - x_b)^2 + (y_n - y_b)^2 + B_r^2 &= 0 \\
(x_n - x_c)^2 + (y_n - y_c)^2 + C_r^2 &= 0
\end{align*}
\]

Thus there are 3 equations and 2 unknowns \((x_n, y_n)\). Since there are more equations than unknowns this is known as an overdetermined system to which there is no unique solution. This can be further complicated if a fourth point \(D\) with distance \(D_r\) is included in the calculations, this results in having 4 equations and 2 unknowns. There are a number of mathematical techniques that can be applied to overdetermined systems to
determine an approximation, the most popular are the Linear Least Squares method and the Non-Linear Least Squares method presented in detail in [78].

Mathematical techniques that solve range-based trilateration problems have been well researched over the past decade. The majority of this research has been targeted at cellular systems, specifically investigating methods that increased the accuracy of localising mobile stations (MS) with multiple wireless connections to base stations (BS). The algorithms discovered within this research [79-81] can be directly applied to our USAR localisation scenario. However, the computational complexity of these algorithms means they could not be implemented in real-time on the hardware used within this project.

In [69] Kaminsky presents a geometric trilateration solution “Circle Intersections with Clustering”. This algorithm was chosen as the basis for my localisation algorithm as it was computationally simple, demonstrated resilience to noisy distance readings from the 802.15.4 based hardware and generated localisation results with an acceptable accuracy, i.e. less than 1.5m.

The implementation of the “Circle Intersections with Clustering” algorithm is presented in detail in the following section along with results from my simulations. The final section of this chapter will present the localisation results obtained using the 802.15.4 network based around the Jennic JN5148-M03 module.
5.2 Localisation Algorithm

The localisation algorithm is based around an algorithm presented by Kaminsky [69], Trilateration using Clustering/Centroids and is illustrated in the flowchart shown in figure 5.2.

![Flowchart of Localisation Algorithm](image-url)

Figure 5.2 Trilateration using Clustering/Centroids Algorithm
Figure 5.3 shows three circles $C_1(x_1, y_1)$, $C_2(x_2, y_2)$ and $C_3(x_3, y_3)$ with radii $R_1$, $R_2$ and $R_3$ respectively. This algorithm breaks the system of circles down into circle-pairs, thus in the three circle example shown in figure 5.2 there are 3 circle-pairs, these are $C_1.C_2$, $C_2.C_3$ and $C_3.C_1$.

The distance between all the intersecting points is calculated. The two intersecting points that have the shortest distance between them are deemed to be the initial two points of the centroid. The initial centre of the centroid is determined by averaging these initial centroid points.
The remaining intersecting points are then tested to determine which is the closest to the centre of centroid and that intersecting point is then deemed to be the next point of the centroid. The average of all centroid positions is again calculated. This continues until an intersection point from each circle pair has been placed in the centroid. Thus, this algorithm determines the closest intersection points of all circle pairs and creates a centroid – the centre of this centroid is then deemed to be the approximate location of the intersection of all circles within the system.

Figure 5.4 shows the centroid resulting from figure 5.3. Here the centroid centre \( T(x_t, y_t) \) is defined by the three coordinates \( P_{12a} (x_{12a}, y_{12a}) \), \( P_{23a} (x_{23a}, y_{23a}) \) and \( P_{31a} (x_{31a}, y_{31a}) \) and is given by the equation:

\[
T(x_t, y_t) = \left( \frac{x_{12a} + x_{23a} + x_{31a}}{3}, \frac{y_{12a} + y_{23a} + y_{31a}}{3} \right) \quad [5.3]
\]

Figure 5.4 Centroid for 3 Intersecting Circles

A geometric method is also used to determine the points of intersections for the circle-pairs. Figure 5.5 shows the scenario where there is a circle-pair whose centres are at \( C_1 \) \((x_1, y_1)\) and \( C_2 \) \((x_2, y_2)\), with radii \( R_1 \) and \( R_2 \) and intersect at points \( P_{12a} \) \((x_{12a}, y_{12a})\) and \( P_{12b} \) \((x_{12b}, y_{12b})\).
To determine the intersections points $P_{12a}$ and $P_{12b}$ the centroid centre created by this circle pair must be first determined, this centroid centre being $i(x_i, y_i)$. Once this is achieved Pythagoras's theorem can then be used to determine the distance from the centroid centre $i$ to the points of intersection $P_{12a}$ and $P_{12b}$. Once this distance has been determined the coordinates of the points of intersection are given by the following formulas:

$$
\begin{align*}
    x_{12a} &= x_i + \left( \frac{x_2 - x_1}{d_{12}} \right) \left( \frac{d_{12}^2 + R_1^2 - R_2^2}{2} \right) - \left( \frac{y_2 - y_1}{d_{12}} \right) \left( \sqrt{R_1^2 - d_{12}^2} \right) \\
    y_{12a} &= y_i + \left( \frac{y_2 - y_1}{d_{12}} \right) \left( \frac{d_{12}^2 + R_1^2 - R_2^2}{2} \right) + \left( \frac{x_2 - x_1}{d_{12}} \right) \left( \sqrt{R_1^2 - d_{12}^2} \right) \\
    x_{12b} &= x_i + \left( \frac{x_2 - x_1}{d_{12}} \right) \left( \frac{d_{12}^2 + R_1^2 - R_2^2}{2} \right) + \left( \frac{y_2 - y_1}{d_{12}} \right) \left( \sqrt{R_1^2 - d_{12}^2} \right) \\
    y_{12b} &= y_i + \left( \frac{y_2 - y_1}{d_{12}} \right) \left( \frac{d_{12}^2 + R_1^2 - R_2^2}{2} \right) - \left( \frac{x_2 - x_1}{d_{12}} \right) \left( \sqrt{R_1^2 - d_{12}^2} \right)
\end{align*}
$$

[5.4] [5.5]
5.3 Localisation Simulation

A simulator based on the algorithm presented in the previous section was written in C to determine its suitability for an USAR robot localisation system. Figure 5.6 shows the flowchart for the software implementation of the Trilateration using Clustering/Centroids algorithm. To resemble a real-world scenario this simulator was developed so that the radii of the circles could be varied randomly within a predefined range, i.e. \( R_{\text{used}} = R \pm \varepsilon \) where \( \varepsilon \) is the randomly induced error. These varying radii would simulate fluctuating distance measurements expected from the wireless nodes.

Two improvements were made to the algorithm presented in figure 5.2, these were:

1. A seed target position was added as the initial centroid centre, and
2. A test within the software was inserted to determine if the circles did in fact intersect.

During testing of this simulator it was found that if the bounds of the random error \( \varepsilon \) were large it was possible for the simulator to select an initial centroid centre that was a long way from the resulting centroid. This initial error could then lead to a result which was not within the centroid. As this simulator is tracking the movement of an USAR robot and the previous position of the USAR is known it was deemed acceptable to use the previous location as a ‘seed’ target position, thus ensuring that the resultant centroid was as accurate as possible.

If the random error \( \varepsilon \) was large then in some instance where this error was subtracted from the radii, i.e. \( R_1 - \varepsilon \) and \( R_2 - \varepsilon \), then it was possible for the circle-pair to not intersect. Thus, a test in the simulator was required to ensure that if a circle-pair didn’t intercept then this circle-pair was removed from the determination of the centroid.
Due to the method that the wireless nodes are deployed, as detailed in section 4, it is only feasible that the USAR robot will be able to localise within the disaster zone from
3 or 4 wireless nodes with known positions. Thus the simulations concentrate on the scenarios where there are 3 or 4 wireless nodes from which localisation can be achieved.

Figures 5.7 and 5.8 show the scenarios where 3 and 4 wireless nodes have been used for localisation. The details of the coordinates and distance measurements, i.e. radii of the circles, used during the simulation are shown in tables 5.1 and 5.2.

Table 5.1 Coordinates and Radii of circles used to simulate 3 node localisation

<table>
<thead>
<tr>
<th>Circle</th>
<th>x (m)</th>
<th>y (m)</th>
<th>R (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle1</td>
<td>0.00</td>
<td>0.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Circle2</td>
<td>28.00</td>
<td>0.00</td>
<td>37.47</td>
</tr>
<tr>
<td>Circle3</td>
<td>8.00</td>
<td>12.00</td>
<td>17.83</td>
</tr>
</tbody>
</table>
Figure 5.8 Simulation scenario using 4 nodes for localisation

<table>
<thead>
<tr>
<th></th>
<th>x (m)</th>
<th>y (m)</th>
<th>R (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circle1</td>
<td>0.00</td>
<td>0.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Circle2</td>
<td>28.00</td>
<td>0.00</td>
<td>37.47</td>
</tr>
<tr>
<td>Circle3</td>
<td>8.00</td>
<td>12.00</td>
<td>17.83</td>
</tr>
<tr>
<td>Circle4</td>
<td>15.00</td>
<td>20.00</td>
<td>13.85</td>
</tr>
</tbody>
</table>

Table 5.2 Coordinates and Radii of circles used to simulate 4 node localisation

The specifications for the TOF capabilities of Jennic’s JN5148 [82] specify that the accuracy of distances calculated via TOF measurements can vary by ±3m, this is over a range of 10 to 300m outdoors and 10 to 24m indoors. Therefore the simulations were performed using distances with random TOF measurement errors to ±5m.

Table 5.3 shows the detailed results for the simulations. 500 tests were done for each TOF distance error. As the TOF distance error was increased so too was the possibility that the circles representing the transmission distance would not intersect. This is shown in table 5.3 as the number of successful simulations and also displayed graphically in figure 5.9. A simulation was deemed successful if all the circles representing transmission distances intersected, i.e. in the 3 wireless node test there had to be 3 successful intersections whilst in the 4 wireless node test there had to be 6 successful intersections. The values of x, y and the distance are averaged values from all successful
simulations whilst the standard deviation for the distances and maximum recorded
distance are also displayed in table 5.3. Figure 5.10 shows the distance that the
simulators results were from the expected results, this is displayed as Distance error vs
TOF error.

<table>
<thead>
<tr>
<th>TOF Error</th>
<th>Samples</th>
<th>Success</th>
<th>x (m)</th>
<th>y (m)</th>
<th>Distance (m)</th>
<th>Stnd Dev (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>500</td>
<td>500</td>
<td>4.989383</td>
<td>29.57234</td>
<td>0.239166</td>
<td>0.273503</td>
<td>0.656826</td>
</tr>
<tr>
<td>0.50</td>
<td>500</td>
<td>500</td>
<td>4.983570</td>
<td>29.55483</td>
<td>0.480437</td>
<td>0.549667</td>
<td>1.355213</td>
</tr>
<tr>
<td>0.75</td>
<td>500</td>
<td>500</td>
<td>4.984702</td>
<td>29.52705</td>
<td>0.726362</td>
<td>0.831713</td>
<td>2.113661</td>
</tr>
<tr>
<td>1.00</td>
<td>500</td>
<td>500</td>
<td>4.993974</td>
<td>29.48730</td>
<td>0.980479</td>
<td>1.124967</td>
<td>2.970422</td>
</tr>
<tr>
<td>1.25</td>
<td>500</td>
<td>500</td>
<td>5.017514</td>
<td>29.43079</td>
<td>1.251356</td>
<td>1.445324</td>
<td>4.471313</td>
</tr>
<tr>
<td>1.50</td>
<td>500</td>
<td>488</td>
<td>4.967194</td>
<td>29.38179</td>
<td>1.482437</td>
<td>1.697919</td>
<td>4.148804</td>
</tr>
<tr>
<td>1.75</td>
<td>500</td>
<td>470</td>
<td>4.886715</td>
<td>29.34835</td>
<td>1.708783</td>
<td>1.957661</td>
<td>5.548287</td>
</tr>
<tr>
<td>2.00</td>
<td>500</td>
<td>449</td>
<td>4.727087</td>
<td>29.30197</td>
<td>1.908527</td>
<td>2.182337</td>
<td>5.978441</td>
</tr>
<tr>
<td>2.25</td>
<td>500</td>
<td>433</td>
<td>4.662369</td>
<td>29.28429</td>
<td>2.109370</td>
<td>2.400706</td>
<td>6.629090</td>
</tr>
<tr>
<td>2.50</td>
<td>500</td>
<td>421</td>
<td>4.648847</td>
<td>29.57234</td>
<td>2.326936</td>
<td>2.647210</td>
<td>7.793591</td>
</tr>
<tr>
<td>2.75</td>
<td>500</td>
<td>402</td>
<td>4.575015</td>
<td>29.27508</td>
<td>2.499766</td>
<td>2.842038</td>
<td>7.93591</td>
</tr>
<tr>
<td>3.00</td>
<td>500</td>
<td>386</td>
<td>4.506860</td>
<td>29.27117</td>
<td>2.666054</td>
<td>3.027737</td>
<td>9.494301</td>
</tr>
<tr>
<td>3.50</td>
<td>500</td>
<td>362</td>
<td>4.432516</td>
<td>29.25149</td>
<td>3.045362</td>
<td>3.457978</td>
<td>10.88026</td>
</tr>
<tr>
<td>4.00</td>
<td>500</td>
<td>343</td>
<td>4.325122</td>
<td>29.21916</td>
<td>3.391260</td>
<td>3.851012</td>
<td>11.85495</td>
</tr>
<tr>
<td>4.50</td>
<td>500</td>
<td>320</td>
<td>4.257745</td>
<td>29.22345</td>
<td>3.687754</td>
<td>4.172698</td>
<td>12.17856</td>
</tr>
<tr>
<td>5.00</td>
<td>500</td>
<td>308</td>
<td>4.126564</td>
<td>29.15822</td>
<td>4.024256</td>
<td>4.551492</td>
<td>12.85495</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TOF Error</th>
<th>Samples</th>
<th>Success</th>
<th>x (m)</th>
<th>y (m)</th>
<th>Distance (m)</th>
<th>Stnd Dev (m)</th>
<th>Max Error (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>500</td>
<td>435</td>
<td>4.990023</td>
<td>29.56407</td>
<td>0.237862</td>
<td>0.272091</td>
<td>0.721219</td>
</tr>
<tr>
<td>0.50</td>
<td>500</td>
<td>344</td>
<td>4.899422</td>
<td>29.45364</td>
<td>0.423187</td>
<td>0.479013</td>
<td>0.999707</td>
</tr>
<tr>
<td>0.75</td>
<td>500</td>
<td>311</td>
<td>4.854924</td>
<td>29.35246</td>
<td>0.631103</td>
<td>0.711215</td>
<td>1.458734</td>
</tr>
<tr>
<td>1.00</td>
<td>500</td>
<td>296</td>
<td>4.816238</td>
<td>29.24874</td>
<td>0.839891</td>
<td>0.944906</td>
<td>1.97167</td>
</tr>
<tr>
<td>1.25</td>
<td>500</td>
<td>287</td>
<td>4.785340</td>
<td>29.14207</td>
<td>1.045936</td>
<td>1.174327</td>
<td>2.897107</td>
</tr>
<tr>
<td>1.50</td>
<td>500</td>
<td>272</td>
<td>4.711083</td>
<td>29.04522</td>
<td>1.237657</td>
<td>1.383880</td>
<td>3.108541</td>
</tr>
<tr>
<td>1.75</td>
<td>500</td>
<td>260</td>
<td>4.657317</td>
<td>28.94824</td>
<td>1.437971</td>
<td>1.607150</td>
<td>3.781446</td>
</tr>
<tr>
<td>2.00</td>
<td>500</td>
<td>243</td>
<td>4.536516</td>
<td>28.85984</td>
<td>1.611453</td>
<td>1.800668</td>
<td>3.629843</td>
</tr>
<tr>
<td>2.25</td>
<td>500</td>
<td>232</td>
<td>4.448932</td>
<td>28.77394</td>
<td>1.788172</td>
<td>1.999164</td>
<td>4.069198</td>
</tr>
<tr>
<td>2.50</td>
<td>500</td>
<td>225</td>
<td>4.366471</td>
<td>28.64073</td>
<td>1.990235</td>
<td>2.222327</td>
<td>4.576611</td>
</tr>
<tr>
<td>2.75</td>
<td>500</td>
<td>214</td>
<td>4.343960</td>
<td>28.56912</td>
<td>2.160311</td>
<td>2.414333</td>
<td>4.972505</td>
</tr>
<tr>
<td>3.00</td>
<td>500</td>
<td>204</td>
<td>4.287648</td>
<td>28.48229</td>
<td>2.380681</td>
<td>2.658962</td>
<td>5.530769</td>
</tr>
<tr>
<td>3.50</td>
<td>500</td>
<td>188</td>
<td>4.247175</td>
<td>28.32632</td>
<td>2.739069</td>
<td>3.051460</td>
<td>6.332737</td>
</tr>
<tr>
<td>4.00</td>
<td>500</td>
<td>176</td>
<td>4.160655</td>
<td>28.18131</td>
<td>3.059930</td>
<td>3.409554</td>
<td>7.357057</td>
</tr>
<tr>
<td>4.50</td>
<td>500</td>
<td>163</td>
<td>4.036371</td>
<td>27.98636</td>
<td>3.369845</td>
<td>3.738449</td>
<td>7.708476</td>
</tr>
<tr>
<td>5.00</td>
<td>500</td>
<td>159</td>
<td>3.939742</td>
<td>27.80587</td>
<td>3.708809</td>
<td>4.107203</td>
<td>8.566875</td>
</tr>
</tbody>
</table>

Table 5.3 Simulator results
Figure 5.9 Successful simulations completed

Figure 5.10 Distance error vs TOF error
It can be seen from figure 5.10 that the results for trilateration when 4 nodes are used is slightly better than when 3 nodes are used. As the size of the TOF error increases so does the difference between the 4 node and 3 node trilateration results. If the TOF errors of 3m are considered the results for the 3 node simulation was 2.66m whilst the 4 node simulation was 2.38m, which is 0.28m closer to the expected result. For all TOF error distances simulated the 3 node standard deviations and maximum distance results were all greater than that of the 4 node results, highlighting that trilateration with 4 nodes gives better accuracy than using 3 nodes. If the TOF error can be kept to 1.5m or less then the accuracy of the localisation results should be less than 1.5m which would be an acceptable distance for positioning of an USAR robot within a disaster zone.

Figure 5.11 shows the distribution of results for a TOF error of 3m for both the 3 node and 4 node simulations. Whilst the majority of results are close to the target position, there are a number of outlying results with a maximum distance from the target of 7.5m and 5.5m for 3 node and 4 node results respectively.

Figure 5.11 Plot of calculated (x, y) results for TOF Error of 3m
5.4 Localisation Results

The hardware in the 802.15.4 based wireless guidance system is based around Jennic’s JN5148. Jennic offer three options to program their devices, these being 802.15.4, ZigBee PRO and JenNet. To program the Jennic JN5148 for TOF measurements the 802.15.4 programming environment [83] was selected. Figure 5.12 shows a block diagram of the software structures required to program the JN5148 for 802.15.4 functionality.

![Figure 5.12 Jennic 802.15.4 Software Architecture](image)

The Application Queue API is optional but was used for the TOF application. This API handles all the interrupts generated by the JN5148 hardware and provides a queue-based interface consisting of pointers to data structures containing information relating the type of event as well as any additional data relating to the interrupt type. Three queues are provided by this API, these are:

1. MAC Data Services (MCPS),
2. MAC Management Services (MLME), and
3. Application Hardware Interface (AHI).
The MCPS and MLME services are provided by the 802.15.4 API whilst any AHI services are provided by the Integrated Peripherals API. Figure 5.13 shows the sequence of events that must occur via the 802.15.4 Stack API to allow a Coordinator and an End Device to become associated and start TOF measurements.

On power-up the Coordinator will perform an ‘Energy Scan’, this involves searching for an 802.15.4 channel that is available to be used. When a suitable channel has been detected then an entry will be made into the MLME Queue, this is read by the Coordinator which will then issue a command to the 802.15.4 API to start the device as a Coordinator. On power-up the End Device will start an ‘Active Channel Search’, this search will be unsuccessful until a Coordinator with the same PAN ID becomes active. When a Coordinator is active an entry will be made in the MLME Queue informing the End Device that a response has been received to the ‘Active Channel Search’.
The End Device will then attempt to establish an association with the Coordinator. Again the Coordinators MLME Queue is used to inform the Coordinator that an End Device is attempting to create an association. The Coordinator will then respond to the End Device informing the End Device as to whether it’s association attempt has been successful or not. In this TOF measurement application the only restriction placed on association attempts is that the number of End Devices associated with the Coordinator cannot exceed ten, this was done purely to control memory usage within the device itself. When the End Device receives the association request response an entry will be made in the End Devices MLME Queue which will determine whether the association attempt has been successful. Once an association has been performed between a Coordinator and an End Device it is then possible to start gathering TOF and RSSI measurements using the TOF API.

Figure 5.14 shows the complete algorithms implemented in the Coordinator and the End Devices for the TOF measurement tests. After the Coordinator and the End Device have associated both enter an infinite control loop. Within this loop the Coordinator will request a TOF measurement from the TOF API, this is initiated by a function call to \texttt{bAppApiGetTof()}. When the TOF measurement has concluded the TOF data will be returned to the control loop within a data structure, data is then extracted from this structure and sent to UART1 where it will be logged by a data logging device.

Figure 5.15 shows the data structure returned from a TOF measurement. Data available within this data structure is the TOF measurement in picoseconds, the RSSI and the Signal Quality Index (SQI) readings at both the Coordinator and End Device, a time stamp as well as a status bite which when equal to zero indicates a successful test has been completed.
Figure 5.14 Events required to associate a Coordinator and an End Device
typedef struct
{
    int32  s32Tof;
    int8   s8LocalRSSI;
    uint8  u8LocalSQI;
    int8   s8RemoteRSSI;
    uint8  u8RemoteSQI;
    uint32 u32Timestamp;
    uint8  u8Status;
} tsAppApiToF_Data;

Figure 5.15 TOF Measurement results data structure

The SQI is defined in [53] as “a value in the range 0 to 255 that represent the correlation quality of the received signal. An SQI value above 200 generally indicates a good signal. The SQI of each TOF value can be used to evaluate the reliability of the measurement”. Thus the SQI will be used when processing the returned data to remove outlying data.

In this particular application the End Device performs no functions within its main control loop, the End Device will simply respond to the TOF requests via the IEEE 802.15.4 Stack Layers provided within Jennic’s 802.15.4 software architecture.

A LabView application was created to perform that data logging for the TOF measurement results. This application simply reads the serial data sent by the Jennic module, writes this data to the PC’s hard drive in a csv file format and then displays the recorded information graphically. The LabView application was configured to record 100 TOF measurement results, this resulted in 100 TOF measurements and 200 RSSI measurements, 100 in the forward direction and 100 in the reverse direction, for each test location. Figure 5.16 shows the LabView GUI where TOF, RSSI and SQI are all displayed graphically.

Finally the csv data files are then processed by Matlab scripts to remove bad data, i.e. when SQI<240 and any remaining outlying TOF results. The processing of the csv files by Matlab will be discussed in further detail as the TOF results are presented in the remainder of this section.
Figure 5.16 LabView Data Logging Graphical User Interface

Figure 5.17 shows the TOF readings recorded for a JN5148-M03 in an indoor environment before being processed by Matlab. The raw data is shown within this figure along with the calculated running average. This test was performed over 10m, as shown previously in equation 3.8:

\[
d = 0.0003TOF
\]

\[
\therefore TOF = \frac{d}{0.0003} = \frac{10}{0.0003} = 3333
\]

Thus the value that would be expected for a TOF reading over 10m would be 3333, this is shown as the theoretical value within figure 5.17. After 100 data samples were taken the averaged distance was 10.178m, an error of just 1.8%. The data did fluctuate significantly though over the 100 sample period. Even though the maximum and
minimum recorded readings had errors of 27.7% and 22.2% the resulting calculated running average was very close to the expected value.

![TOF data from JN5148-001-M03 indoors](image)

Figure 5.17 TOF Data from JN5148-001-M03 indoors – distance 10m

The JN5148-001-Myy module that is being used as the basis of the 802.15.4 wireless robotic guidance system come in two configurations:

1. M00 – module has an integrated printed circuit board antenna, and
2. M03 – module has an SMA connector to allow for an external antenna.

To determine if one of these modules had superior performance it was decided to test RSSI and TOF capabilities of both the M00 and the M03 in an outdoor and an indoor environment.

The outdoor tests were performed on an asphalt sports court. Two JN5148 devices were used, one configured as a Coordinator and the other configured as an End Device. Tests were conducted at 1m intervals until wireless connection between the two devices became inoperable, the results of these tests are shown in table 5.4.
<table>
<thead>
<tr>
<th>Dist</th>
<th>M00 module</th>
<th>TOF</th>
<th>Error (%)</th>
<th>RSSI</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.17</td>
<td>17</td>
<td>69.68</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4.26</td>
<td>17</td>
<td>63.02</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4.76</td>
<td>15</td>
<td>64.08</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5.40</td>
<td>17</td>
<td>64.94</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6.18</td>
<td>13</td>
<td>64.44</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>8.81</td>
<td>17</td>
<td>61.65</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7.48</td>
<td>10</td>
<td>59.47</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5.41</td>
<td>11</td>
<td>55.84</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6.02</td>
<td>17</td>
<td>53.68</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>9.27</td>
<td>13</td>
<td>52.57</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>9.34</td>
<td>27</td>
<td>50.28</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10.25</td>
<td>7</td>
<td>49.16</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>12.79</td>
<td>27</td>
<td>48.05</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>15.21</td>
<td>37</td>
<td>47.27</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>16.47</td>
<td>43</td>
<td>45.54</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>16.96</td>
<td>33</td>
<td>44.62</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>14.43</td>
<td>21</td>
<td>43.34</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>15.79</td>
<td>38</td>
<td>41.71</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>20.95</td>
<td>26</td>
<td>43.64</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>21.47</td>
<td>11</td>
<td>42.19</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>21.23</td>
<td>9</td>
<td>40.75</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>20.04</td>
<td>58</td>
<td>39.80</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>25.00</td>
<td>36</td>
<td>39.16</td>
<td>38</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>28.95</td>
<td>98</td>
<td>39.53</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>28.00</td>
<td>99</td>
<td>39.12</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dist</th>
<th>M003 module</th>
<th>TOF</th>
<th>Error (%)</th>
<th>RSSI</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.88</td>
<td>98</td>
<td>73.05</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.51</td>
<td>91</td>
<td>68.16</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.85</td>
<td>41</td>
<td>65.76</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.19</td>
<td>0</td>
<td>62.05</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.20</td>
<td>27</td>
<td>57.53</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.47</td>
<td>1</td>
<td>54.83</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4.71</td>
<td>3</td>
<td>54.59</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>3.90</td>
<td>45</td>
<td>52.37</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>5.18</td>
<td>76</td>
<td>50.38</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10.04</td>
<td>1</td>
<td>50.11</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>7.11</td>
<td>73</td>
<td>49.18</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>12.71</td>
<td>14</td>
<td>48.51</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>11.10</td>
<td>4</td>
<td>49.43</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>9.35</td>
<td>99</td>
<td>47.90</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>11.46</td>
<td>38</td>
<td>45.28</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>14.55</td>
<td>6</td>
<td>44.93</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>19.62</td>
<td>25</td>
<td>45.39</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>17.06</td>
<td>10</td>
<td>44.52</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>19.30</td>
<td>11</td>
<td>43.40</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>17.40</td>
<td>57</td>
<td>42.67</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>19.50</td>
<td>99</td>
<td>39.95</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>19.00</td>
<td>30</td>
<td>41.78</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>24.06</td>
<td>4</td>
<td>41.36</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>21.00</td>
<td>98</td>
<td>36.58</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>22.00</td>
<td>99</td>
<td>38.17</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>21.51</td>
<td>79</td>
<td>36.06</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>24.00</td>
<td>97</td>
<td>36.40</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>28.99</td>
<td>94</td>
<td>38.60</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>27.35</td>
<td>74</td>
<td>35.97</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>28.51</td>
<td>67</td>
<td>36.51</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4 RSSI and TOF results for M00 and M03 modules Outdoors

Matlab scripts were used to process the data recorded by the LabView data logging software. Two variables were used to remove outlying TOF data, these variable were the SQI and a TOF limiting constant, $k_{TOF}$. The data was initially processed with SQI values ranging from 200 to 250 and it was found that varying the SQI reading had minimal impact on the resulting RSSI and TOF data, so an SQI level of 240 was selected as the minimum allowable for a reading to be deemed good.
A TOF limiting constant, $k_{TOF}$, was then applied to the TOF results to remove outlying TOF measurement, this ensured that only TOF measurements with a $\pm k_{TOF}$ range of the expected result were included in the measurement calculations. In an USAR robot scenario, the previous position of the USAR robot will be known, so if $k_{TOF}$ is set such that it allows for the maximum expected TOF error and the expected maximum movement of the USAR robot within the one second localisation polling period then this methodology is justified.

Values of 10m, 5m and 3m for $k_{TOF}$ were tested, this resulted in an error rate of 27%, 45%, 57%. As the specified TOF error for the JN5148 is $\pm 3m$ and the USAR robot will never travel at more than $2m/s$ within the disaster zone, it was decided that $k_{TOF}$ be set to 5m. This would ensure that TOF measurements within the $\pm 3m$ TOF error specification to be included within the results and leave us with a relatively large dataset of results.

Figures 5.18 and 5.19 show the results of RSSI and TOF plotted against the measured distances. RSSI is generally only reliable for distance measurement applications over 6m [82], so only the first 10m of the RSSI readings are shown. The results show that RSSI readings for both the M00 and M03 modules are reliable over 10m. The TOF results show how both the M00 and M03 modules performed. The M00 module was only able to operate over a 24m range whilst the M03 was able to operate over 30m. But as shown in table 5.4 once the transmission distance was higher than 24m a significantly large number of measurements, greater than 90%, were discarded. Generally though, the TOF results stayed within the specified $\pm 3m$ error range over the full range of distances tested.

Similar tests were then performed in an indoor environment with both the M00 and M03 modules. A hallway with the Electrical Engineering (EE) building was used for these tests as this site would be used for the localisations tests. As a hallway was used for these tests there was a direct line of sight between the two nodes whilst these tests were being conducted. The RSSI and TOF distance measurement tests were carried out on the first floor of this two storey building. The floor is made from reinforced concrete, the walls are hollow brick and a tin roof was located 3.2m above the floor. Results were
recorded via the LabView data logging software and then processed by Matlab scripts to remove any outlying TOF measurement results. It was again found that values of SQI of 240 and a of $k_{TOF}$ 5m resulted in a set of results where 25% of recorded TOF measurements were deemed to be outlying data.

Figure 5.18 RSSI vs Distance for JN5148-Myy in an Outdoor Environment

Figure 5.19 TOF vs Distance for JN5148-Myy in an Outdoor Environment
Figures 5.20 and 5.21 show the results of RSSI and TOF plotted against the measured distances. The RSSI results were good to 4m but after this point the results for the M03 fluctuated away from the expected theoretical values. The TOF results show that a transmission distance of 40m was able to be achieved in the indoor hallway, for the M00 module this was 16m longer than in the outdoor environment. Table 5.5 shows the TOF errors, TOF measurements that were deemed as outlying data. These TOF errors are consistent over the full 40m testing distance. The TOF results over the full range of tested distances remained close to the specified ±3m TOF error.

The recorded data sets were both analysed to determine the RSSI level that could indicate when the wireless communication link was deteriorating. This would then be the threshold that would determine when a Drop Node should be deployed to ensure the wireless robotic guidance system remained intact. Matlab scripts processed the data with the SQI being used as the variable to filter out good results, so the number of bad results versus RSSI could be determined. It was concluded that for the outdoor environment an RSSI reading of 40 should be used whilst for the indoor environment an RSSI reading of 44 should be used.

After testing both the M00 and M03 module in an outdoor and indoor environment it was decided to use the M03 module for further localisation testing. The primary reason for this was that the number of RSSI and TOF errors received by the M00 module was far greater than the M03 modules. In the indoor environment, the M00 module had an average of 25% bad TOF measurements and 18% bad RSSI measurements whilst the M03 module had an average of 20% for the TOF measurements and 1% for the RSSI measurements.

Therefore the localisations tests would use the JN5148 M03 module with the following parameters:

- $\text{SQI} \geq 240$, and
- $k_{\text{TOF}} = 5\text{m}$. 
Figure 5.20 RSSI vs Distance for JN5148-Myy in an Indoor Environment

Figure 5.21 TOF vs Distance for JN5148-Myy in an Indoor Environment
Table 5.5 RSSI and TOF results for M00 and M03 modules Indoors

<table>
<thead>
<tr>
<th>Dist</th>
<th>M00 Module</th>
<th></th>
<th>M03 Module</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOF</td>
<td>RSSI</td>
<td>TOF</td>
<td>RSSI</td>
</tr>
<tr>
<td></td>
<td>(m)</td>
<td>Error (%)</td>
<td>(m)</td>
<td>Error (%)</td>
</tr>
<tr>
<td>1</td>
<td>2.57</td>
<td>10</td>
<td>75.35</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>3.35</td>
<td>10</td>
<td>66.95</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>6.88</td>
<td>46</td>
<td>64.04</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>1.37</td>
<td>47</td>
<td>51.73</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>2.86</td>
<td>12</td>
<td>54.17</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>10.06</td>
<td>56</td>
<td>56.57</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>9.20</td>
<td>22</td>
<td>56.21</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>10.12</td>
<td>24</td>
<td>55.39</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>10.87</td>
<td>12</td>
<td>51.43</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11.50</td>
<td>99</td>
<td>61.01</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>14.15</td>
<td>19</td>
<td>53.37</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>13.08</td>
<td>17</td>
<td>52.83</td>
<td>17</td>
</tr>
<tr>
<td>13</td>
<td>16.16</td>
<td>22</td>
<td>60.19</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>13.32</td>
<td>17</td>
<td>52.32</td>
<td>17</td>
</tr>
<tr>
<td>15</td>
<td>14.70</td>
<td>21</td>
<td>57.47</td>
<td>21</td>
</tr>
<tr>
<td>16</td>
<td>16.20</td>
<td>13</td>
<td>57.81</td>
<td>13</td>
</tr>
<tr>
<td>17</td>
<td>19.39</td>
<td>26</td>
<td>59.99</td>
<td>23</td>
</tr>
<tr>
<td>18</td>
<td>20.23</td>
<td>16</td>
<td>60.64</td>
<td>13</td>
</tr>
<tr>
<td>19</td>
<td>20.04</td>
<td>19</td>
<td>55.47</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>20.74</td>
<td>16</td>
<td>50.19</td>
<td>16</td>
</tr>
<tr>
<td>21</td>
<td>21.68</td>
<td>19</td>
<td>51.49</td>
<td>19</td>
</tr>
<tr>
<td>22</td>
<td>25.90</td>
<td>37</td>
<td>51.44</td>
<td>9</td>
</tr>
<tr>
<td>23</td>
<td>25.09</td>
<td>16</td>
<td>49.57</td>
<td>17</td>
</tr>
<tr>
<td>24</td>
<td>24.03</td>
<td>10</td>
<td>53.43</td>
<td>10</td>
</tr>
<tr>
<td>25</td>
<td>27.64</td>
<td>28</td>
<td>49.26</td>
<td>25</td>
</tr>
<tr>
<td>26</td>
<td>26.08</td>
<td>31</td>
<td>47.65</td>
<td>31</td>
</tr>
<tr>
<td>27</td>
<td>26.57</td>
<td>20</td>
<td>50.03</td>
<td>20</td>
</tr>
<tr>
<td>28</td>
<td>26.40</td>
<td>35</td>
<td>45.82</td>
<td>30</td>
</tr>
<tr>
<td>29</td>
<td>29.62</td>
<td>18</td>
<td>53.97</td>
<td>18</td>
</tr>
<tr>
<td>30</td>
<td>30.43</td>
<td>15</td>
<td>54.05</td>
<td>15</td>
</tr>
<tr>
<td>31</td>
<td>31.53</td>
<td>31</td>
<td>51.13</td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>29.99</td>
<td>23</td>
<td>46.78</td>
<td>23</td>
</tr>
<tr>
<td>33</td>
<td>32.68</td>
<td>23</td>
<td>47.29</td>
<td>23</td>
</tr>
<tr>
<td>34</td>
<td>34.78</td>
<td>39</td>
<td>44.90</td>
<td>37</td>
</tr>
<tr>
<td>35</td>
<td>36.03</td>
<td>32</td>
<td>51.15</td>
<td>22</td>
</tr>
<tr>
<td>36</td>
<td>32.38</td>
<td>33</td>
<td>47.50</td>
<td>17</td>
</tr>
<tr>
<td>37</td>
<td>35.12</td>
<td>30</td>
<td>48.59</td>
<td>29</td>
</tr>
<tr>
<td>38</td>
<td>38.62</td>
<td>24</td>
<td>50.09</td>
<td>24</td>
</tr>
<tr>
<td>39</td>
<td>40.13</td>
<td>13</td>
<td>47.84</td>
<td>13</td>
</tr>
<tr>
<td>40</td>
<td>40.71</td>
<td>25</td>
<td>47.05</td>
<td>25</td>
</tr>
</tbody>
</table>

To test the localisation algorithm the first floor of the Electrical Engineer (EE) building was chosen as the location, an overview of this building is shown in figure 5.22. This
building and it’s internal fittings could resemble a disaster zone environment as there are many obstacles that can cause attenuation of wireless signals. These will be detailed later within this section. Figure 5.22 shows four locations where fixed nodes will be placed, labelled A, B, C, and D, and twenty test points, labelled 1 to 20.

![Figure 5.22 Indoor localisation overview](image)

The four fixed nodes were tested over a 10m distance to ensure their transceivers all had similar characteristics. The results of these 10m distance tests were:

- Node A: 9.253m 1% TOF error,
- Node B: 10.503m 7% TOF error,
- Node C: 10.192m 3% TOF error, and
- Node D: 10.254m 2% TOF error.

An initial test of the localisation algorithm was conducted in an outdoor environment, again on an asphalt sports court. Figure 5.23 shows the overview of this test scenario. It was decided to place the fixed nodes in exactly the same locations as that proposed for
the EE building tests so comparisons could be made between the outdoor and indoor localisation tests. Five locations from the EE building scenario were also used.

In the results obtained for RSSI and TOF measurements, as shown previously in figures 5.19 and 5.20, the RSSI measurements were accurate over the 10m they were tested. To determine the distance used in the localisation algorithm the following rule was used:

\[
\text{if the expected distance measurement is less than 6m then use RSSI result} \\
\text{else use the TOF result}
\]

Tables 5.6 and 5.7 present the results of the outdoor localisation test. Table 5.6 shows distances measured by the RSSI and TOF readings, e.g. for Position A to Location 1 the actual distance is 12.5m, the measured distance was 9.06m which was an error of 3.44m and 7% of the RSSI/TOF readings were errors. Only three of the distances were greater than the ±3m TOF error specified by Jennic, whilst the average number of erroneous readings was 11.4%. The distances recorded were then passed to the localisation
algorithm and the resulting data can be seen in table 5.7, detailed results can be found in Appendix G-1.

<table>
<thead>
<tr>
<th>Actual (m)</th>
<th>Reading (m)</th>
<th>Diff (m)</th>
<th>BAD (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.50</td>
<td>9.06</td>
<td>3.44</td>
</tr>
<tr>
<td>4</td>
<td>13.89</td>
<td>10.84</td>
<td>3.05</td>
</tr>
<tr>
<td>12</td>
<td>4.00</td>
<td>3.52</td>
<td>0.48</td>
</tr>
<tr>
<td>15</td>
<td>10.37</td>
<td>8.96</td>
<td>1.41</td>
</tr>
<tr>
<td>18</td>
<td>14.15</td>
<td>12.34</td>
<td>1.81</td>
</tr>
<tr>
<td>Actual (m)</td>
<td>Reading (m)</td>
<td>Diff (m)</td>
<td>BAD (%)</td>
</tr>
<tr>
<td>1</td>
<td>9.36</td>
<td>5.78</td>
<td>3.58</td>
</tr>
<tr>
<td>4</td>
<td>15.93</td>
<td>16.26</td>
<td>0.33</td>
</tr>
<tr>
<td>12</td>
<td>15.56</td>
<td>12.82</td>
<td>2.74</td>
</tr>
<tr>
<td>15</td>
<td>10.75</td>
<td>8.85</td>
<td>1.90</td>
</tr>
<tr>
<td>18</td>
<td>5.37</td>
<td>4.66</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 5.6 RSSI and TOF results for M03 modules Outdoors

| Actual (m) | Reading (m) | Diff (m) | BAD (%) |
| 1          | 9.36        | 10.27    | 0.92    | 1       |
| 4          | 5.44        | 4.78     | 0.66    | 0       |
| 12         | 15.56       | 18.35    | 2.79    | 12      |
| 15         | 19.28       | 18.16    | 1.11    | 2       |
| 18         | 16.40       | 14.57    | 1.83    | 50      |

| Actual (m) | Reading (m) | Diff (m) | BAD (%) |
| 1          | 11.93       | 9.53     | 2.40    | 1       |
| 4          | 6.71        | 4.36     | 2.35    | 2       |
| 12         | 10.20       | 7.82     | 2.38    | 3       |
| 15         | 18.01       | 16.78    | 1.23    | 27      |
| 18         | 18.50       | 17.33    | 1.17    | 40      |

Table 5.7 Localisation results for M03 modules Outdoors

The localisation results presented in table 5.7 are an average distance of 1.32m from the actual location, with the closest being 1.07m and the farthest away being 1.52m. This is a good result for an USAR robot scenario where the localisation is being used to position survivors within a disaster zone. If a rescuer can make it to the position determined by this algorithm they should be close enough to actually see the victim and remove them from the disaster zone.
Figure 5.24 shows the results presented in table 5.7 graphically, with the red points being the actual positions and the blue points the positions determined by the RSSI and TOF measurements and the localisation algorithm.

![Graphical representation of Outdoor Localisation Tests](image)

To illustrate how the cluster and centroid concept has been applied in this algorithm the cluster created for positions 15 and 18, as shown on figure 5.23, will be examined. Table 5.8 shows the five points that were determined by the algorithm and used created the cluster for positions 15 and 18 whilst figure 5.25 displays the cluster surrounding the centroid position.

<table>
<thead>
<tr>
<th>Position 15</th>
<th>Position 18</th>
</tr>
</thead>
<tbody>
<tr>
<td>x (m)</td>
<td>y (m)</td>
</tr>
<tr>
<td>1</td>
<td>5.24</td>
</tr>
<tr>
<td>2</td>
<td>5.22</td>
</tr>
<tr>
<td>3</td>
<td>4.14</td>
</tr>
<tr>
<td>4</td>
<td>5.43</td>
</tr>
<tr>
<td>5</td>
<td>5.60</td>
</tr>
<tr>
<td>AVG</td>
<td>4.94</td>
</tr>
<tr>
<td></td>
<td>7.10</td>
</tr>
</tbody>
</table>

Table 5.8 Position 15 and 18 Centroid data
It can be seen in figure 5.25 that the areas within the clusters are quite large but the resulting centroid of these clusters is quite close to the expected coordinates, 1.07m and 1.33m for position 15 and 18 respectively.

Figure 5.25 Position 15 and 18 Clusters and Centroids

Figure 5.26 shows the detailed overview of the location for the indoor localisation tests. The first floor of the Electrical Engineering (EE) building was selected as a suitable location, this building has a concrete reinforced floor, hollow brick walls and a tin roof located 3.2m from the floor. The first floor of the EE building contains a number of items that can cause attenuation for wireless signals, these are details in the legend of figure 5.26. There are a number of metallic items that will block wireless signals whilst others that will severely attenuate wireless signals. There are also three 802.11 Wireless Access Points (WAP) distributed on the roof of this building which could possibly cause signal corruption. This environment was deliberately chosen for these characteristics as it allows us to determine what impact this may have on this localisation system within an actual disaster area.
In the results obtained for RSSI and TOF measurements, as shown previously in figures 5.20 and 5.21, the RSSI measurements for the M03 were not good after 4m, therefore to determine the distance used in the localisation algorithm the following rule was used:

\[
\text{if the expected distance measurement is less than 4m then use RSSI result} \\
\text{else use the TOF result}
\]

Tables 5.9 and 5.10 present the results of the indoor localisation test, detailed results can be found in Appendix G-2. Table 5.9 shows distances measured by the RSSI and TOF readings, e.g. for Position B to Location 1 the actual distance is 9.4m, the measured distance was 12.08m which was an error of 3.40m and 76% of the RSSI/TOF readings were errors. The average number of erroneous readings was 30.9%, a significant increase from the 11.4% experienced in the outdoor localisation tests.
Table 5.9 (a) RSSI and TOF results for M03 modules Indoors

<table>
<thead>
<tr>
<th>Position A</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Reading</td>
<td>Diff</td>
<td>BAD</td>
</tr>
<tr>
<td></td>
<td>(m)</td>
<td>(m)</td>
<td>(m)</td>
<td>(%)</td>
</tr>
<tr>
<td>1</td>
<td>12.53</td>
<td>NaN</td>
<td>--</td>
<td>51</td>
</tr>
<tr>
<td>2</td>
<td>14.35</td>
<td>NaN</td>
<td>--</td>
<td>96</td>
</tr>
<tr>
<td>3</td>
<td>15.68</td>
<td>15.0900</td>
<td>0.5883</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>13.92</td>
<td>12.2154</td>
<td>1.7062</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>10.33</td>
<td>13.8336</td>
<td>3.4987</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>11.44</td>
<td>11.8324</td>
<td>0.3952</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>10.03</td>
<td>11.4500</td>
<td>1.4190</td>
<td>58</td>
</tr>
<tr>
<td>8</td>
<td>6.77</td>
<td>7.1500</td>
<td>0.3817</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>6.47</td>
<td>9.9200</td>
<td>3.4539</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>8.33</td>
<td>9.6900</td>
<td>1.3611</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>6.95</td>
<td>2.8100</td>
<td>4.1449</td>
<td>83</td>
</tr>
<tr>
<td>12</td>
<td>4.10</td>
<td>4.1800</td>
<td>0.0800</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>6.95</td>
<td>9.8500</td>
<td>2.8951</td>
<td>5</td>
</tr>
<tr>
<td>14</td>
<td>8.96</td>
<td>11.1800</td>
<td>2.2151</td>
<td>59</td>
</tr>
<tr>
<td>15</td>
<td>10.41</td>
<td>14.8900</td>
<td>4.4799</td>
<td>75</td>
</tr>
<tr>
<td>16</td>
<td>8.64</td>
<td>11.0100</td>
<td>2.3746</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>12.13</td>
<td>NaN</td>
<td>--</td>
<td>78</td>
</tr>
<tr>
<td>18</td>
<td>14.18</td>
<td>NaN</td>
<td>--</td>
<td>84</td>
</tr>
<tr>
<td>19</td>
<td>14.35</td>
<td>16.9000</td>
<td>2.5539</td>
<td>98</td>
</tr>
<tr>
<td>20</td>
<td>13.63</td>
<td>NaN</td>
<td>--</td>
<td>86</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Position B</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Reading</td>
<td>Diff</td>
<td>BAD</td>
</tr>
<tr>
<td></td>
<td>(m)</td>
<td>(m)</td>
<td>(m)</td>
<td>(%)</td>
</tr>
<tr>
<td>1</td>
<td>9.40</td>
<td>12.8005</td>
<td>3.4020</td>
<td>76</td>
</tr>
<tr>
<td>2</td>
<td>11.53</td>
<td>11.3815</td>
<td>0.1525</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>15.40</td>
<td>15.8318</td>
<td>0.4359</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>15.95</td>
<td>17.7269</td>
<td>1.7760</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>15.48</td>
<td>14.5719</td>
<td>0.9049</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>17.19</td>
<td>15.0297</td>
<td>2.1614</td>
<td>37</td>
</tr>
<tr>
<td>7</td>
<td>18.88</td>
<td>19.6843</td>
<td>0.7998</td>
<td>69</td>
</tr>
<tr>
<td>8</td>
<td>20.07</td>
<td>NaN</td>
<td>--</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>18.66</td>
<td>NaN</td>
<td>--</td>
<td>84</td>
</tr>
<tr>
<td>10</td>
<td>16.89</td>
<td>NaN</td>
<td>--</td>
<td>98</td>
</tr>
<tr>
<td>11</td>
<td>14.71</td>
<td>18.1055</td>
<td>3.3959</td>
<td>56</td>
</tr>
<tr>
<td>12</td>
<td>15.59</td>
<td>19.8702</td>
<td>4.2807</td>
<td>87</td>
</tr>
<tr>
<td>13</td>
<td>12.34</td>
<td>16.0859</td>
<td>3.7420</td>
<td>79</td>
</tr>
<tr>
<td>14</td>
<td>10.97</td>
<td>12.7909</td>
<td>1.8195</td>
<td>25</td>
</tr>
<tr>
<td>15</td>
<td>10.79</td>
<td>14.2667</td>
<td>3.4791</td>
<td>30</td>
</tr>
<tr>
<td>16</td>
<td>10.58</td>
<td>11.2483</td>
<td>0.6713</td>
<td>5</td>
</tr>
<tr>
<td>17</td>
<td>7.19</td>
<td>6.9075</td>
<td>0.2850</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>5.45</td>
<td>5.1718</td>
<td>0.2764</td>
<td>2</td>
</tr>
<tr>
<td>19</td>
<td>4.87</td>
<td>7.0374</td>
<td>2.1658</td>
<td>8</td>
</tr>
<tr>
<td>20</td>
<td>5.98</td>
<td>8.1986</td>
<td>2.2209</td>
<td>4</td>
</tr>
</tbody>
</table>

There were a number of distance measurements that failed when performing the tests, these are designated as Not a Number (NaN) within table 5.9 and table 5.10. These NaN results were caused by the metallic obstacles within the EE Building. The measurements from Position A to Locations 1 and 2 both returned NaN results. Figure 5.27 illustrates why this is the case, the green lines from fixed node A shows the shortest direct path for wireless signals to travel to the nodes at various locations. The wireless signal from fixed node A to 1 and 2 are both obstructed but metallic cupboards, shelving and filing cabinets thus no communications is ever established between these nodes. Direct signal path from fixed node A to 15 travels through a number of hollow brick wall which contain plumbing for the toilets, even though a communication link was able to be established between fixed node A and location 15 there was a TOF measurement error of 4.48m which is well outside of the ±3m TOF error specification.
<table>
<thead>
<tr>
<th></th>
<th>Position C</th>
<th></th>
<th>Position D</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual (m)</td>
<td>Reading (m)</td>
<td>Diff (m)</td>
<td>BAD (%)</td>
<td>Actual (m)</td>
</tr>
<tr>
<td>1</td>
<td>9.40</td>
<td>13.4884</td>
<td>4.0899</td>
<td>57</td>
</tr>
<tr>
<td>2</td>
<td>6.09</td>
<td>6.9141</td>
<td>0.8287</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3.61</td>
<td>1.7300</td>
<td>1.8801</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>5.52</td>
<td>6.9773</td>
<td>1.4607</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>8.92</td>
<td>12.3741</td>
<td>3.4560</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>8.46</td>
<td>8.9451</td>
<td>0.4874</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>10.80</td>
<td>11.2616</td>
<td>0.4624</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>14.52</td>
<td>14.5825</td>
<td>0.0659</td>
<td>53</td>
</tr>
<tr>
<td>9</td>
<td>13.71</td>
<td>13.5613</td>
<td>0.1512</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>11.20</td>
<td>13.4131</td>
<td>2.2161</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>12.34</td>
<td>10.8978</td>
<td>1.4461</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>15.59</td>
<td>NaN</td>
<td>--</td>
<td>94</td>
</tr>
<tr>
<td>13</td>
<td>14.71</td>
<td>16.4011</td>
<td>1.6915</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>17.67</td>
<td>19.7453</td>
<td>2.0712</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>19.30</td>
<td>23.0728</td>
<td>3.7758</td>
<td>32</td>
</tr>
<tr>
<td>16</td>
<td>15.49</td>
<td>NaN</td>
<td>--</td>
<td>93</td>
</tr>
<tr>
<td>17</td>
<td>15.61</td>
<td>26.0728</td>
<td>10.4609</td>
<td>92</td>
</tr>
<tr>
<td>18</td>
<td>16.42</td>
<td>19.0492</td>
<td>2.6272</td>
<td>33</td>
</tr>
<tr>
<td>19</td>
<td>14.69</td>
<td>11.8512</td>
<td>2.8366</td>
<td>15</td>
</tr>
<tr>
<td>20</td>
<td>12.80</td>
<td>1.5000</td>
<td>11.2958</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 5.10 RSSI and TOF results for M03 modules Indoors

In situations where there were NaN results the localisation algorithm would use the remaining signals to determine a location. Table 5.11 shows the results of the indoor localisation tests. The average distance that these results were from the actual locations was 1.11m, this is extremely good considering the number of attenuating obstacles that are present in the test environment. There were three results that were significantly larger than the average, these were at locations 12, 14 and 15. These bad results were due to the locations of the nodes when testing. Location 12 was prevented from getting a clear signal from fixed nodes B and C, thus it had to localise from 2 readings that were in excess of 4m off the expected distance. Locations 14 and 15 had a similar issue to that suffered by node located at 12. Nodes at location 14 and 15 couldn’t get clear signals from fixed nodes A and C, thus having to localise from only two reliable sources.
Figure 5.27 Indoor localisation wireless signal path Fixed Node A

Table 5.11 Localisation results for M03 modules Indoors

<table>
<thead>
<tr>
<th>Intercepts</th>
<th>Actual</th>
<th>Reading</th>
<th>DIST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x$ (m)</td>
<td>$y$ (m)</td>
<td>$x$ (m)</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>12.00</td>
<td>12.32</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>15.00</td>
<td>14.61</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>19.00</td>
<td>19.72</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>19.00</td>
<td>19.34</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>17.00</td>
<td>17.00</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>19.00</td>
<td>18.19</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>19.50</td>
<td>20.23</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>18.00</td>
<td>18.21</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>17.00</td>
<td>17.56</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>17.00</td>
<td>16.85</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>14.00</td>
<td>14.08</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>12.00</td>
<td>12.58</td>
</tr>
<tr>
<td>13</td>
<td>6</td>
<td>10.00</td>
<td>9.33</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>6.00</td>
<td>3.74</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>4.00</td>
<td>3.69</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>8.00</td>
<td>7.63</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>6.00</td>
<td>5.45</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>4.50</td>
<td>2.97</td>
</tr>
<tr>
<td>19</td>
<td>6</td>
<td>6.00</td>
<td>6.36</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>8.00</td>
<td>7.21</td>
</tr>
</tbody>
</table>

Average distance: 1.07
Figure 5.28 shows the graphical representation of results generated by the localisation algorithm. As mentioned previously, the only locations not close to the USAR robots path are locations 12, 14 and 15.

![Graphical representation of Indoor Localisation Tests](image.png)
5.5 Summary

After discussing various methods by which localisation may be achieved within wireless networks simulations of the ‘Trilateration using Clustering and Centroids’ were performed. The simulations were performed with random TOF errors to ±5m. The specifications for Jennic’s JN5148 indicate that distance measurements using TOF can be within ±3m. The value of ±5m was chosen as this would allow an USAR robot to move 2m between localisation calculations. As localisation calculations are performed by this system every 1s, this restricts the maximum speed of an USAR robot travel to 2m/s. As the maximum speed of the tank-based USAR is 0.76m/s this is not an unrealistic restriction to impose on an USAR robot operating within a disaster zone.

Results from these simulations indicated that when a wireless node was able to communicate with three other nodes then a position could be determined to within 2.66m with a standard deviation of 3.03m. When a wireless node could communicate with four other nodes the simulation results indicated that a position to within 2.38m with a standard deviation of 2.66m could be achieved. The results of these simulations indicate that localisation with the wireless robotic guidance system should be no worse than 2.66m, given the JN5148 modules can have a maximum ±3m error in their TOF based distance measurements.

Before testing localisation within the wireless guidance system a series of tests were performed using the JN5148 modules to determine the relationship between RSSI and TOF measurement based distances and the actual distance between communicating wireless nodes. Tests were performed on two types of JN5148 modules, the first being the JN5148-M00 which has its antenna integrated into the modules PCB and the second being the JN5148-M03 which uses a standard 2.5GHz antenna attached to an SMA connector.

The initial tests were performed outdoors in an open area on an asphalt surface. The JN5148-M00 was able to communicate over a 24m range whilst the JN5148 was able to communicate over a 30m range. RSSI and TOF measurements were conducted every metre, these were then processed via a Matlab script to remove outlying data. The
results obtained for RSSI based distance for first 10m were as expected for both module types. Results for the TOF based measurements were in the majority of cases with the ±3m distance specified by the manufacturer. The data acquired was also analysed and it was concluded that for an outdoor environment that an RSSI reading of 40 would indicate that a communication link was deteriorating and a Drop Node Sequence should be initiated.

Similar tests were then performed in an indoor environment. The indoor environment selected was the first floor of a building of a two storey building. The floor was constructed from reinforced concrete, the walls were constructed from hollow bricks and a tin roof was situated 3.2m above the first floor. In this environment the JN5148-M00 and JN5148-M03 were able to communicate 40m. Interestingly the JN5148-M03 was only able to establish a reliable communication link over 30m outdoors whilst achieving 40m in the indoor environment. This characteristics of the hallway used as the concrete reinforced floor, hollow brick walls and a tin roof, contributed to act as a ‘wave guide’. This allowed mutli-path fading to constructively interfere with the original signal and thus increase the distance over which reliable communication could be established.

The distances calculated from the RSSI readings indoor for both modules was not as good as those achieved outside, with only the RSSI based distance measurements for the first 4m being close to the expected reading. Once the Matlab scripts had removed the outlying data the majority of TOF distance measurements were within the ±3m specification.

It was decided to use the JN5148-M03 for the localisation tests as results from the indoor distance tests showed that the:

- JN5148-M00 module had an average of 25% bad TOF measurements and 18% bad RSSI measurements, whilst the
- JN5148-M03 module had an average of 20% for the TOF measurements and 1% for the RSSI measurements.
Using a slightly modified version of the ‘Trilateration using Clustering and Centroids’ algorithm and RSSI and TOF distance measurements the location of a wireless node was determined to an average accuracy of 1.32m outdoors and 1.07m indoors. During localisation testing the average TOF error obtained for both outdoor and indoor was 1.8m, well within the ±3m as specified by the manufacturer. The simulations indicated that for situations where three nodes and four node trilateration was used with an induced TOF error of ±1.75m the expected accuracy should be 1.7m and 1.437m respectively. Thus the results achieved by the localisation tests are well within the results indicated by the simulations.

The results could be refined to generate more accurate results by:

1. Weighting the calculations used to determine the centroid with scaling factors based on how good the RSSI-and/or TOF results were. This could be achieved by using the number of RSSI-TOF errors that occur whilst gathering data. A weighting could then be assigned based on this such that more weight was given to the centroid coordinates that had been generated by good distance measurements.

2. The use of additional sensors from the USAR robot, such as encoders, digital compass and accelerometer, to compliment the localisation results from the wireless robotic guidance system. Having an algorithm that had many sources of location based data could result in more accurate localisation.

The results obtained for localising via the wireless USAR robotic guidance system allow us to not only control the USAR robots within a disaster zone wirelessly but also accurately determine its position to within an accuracy of 1.1m. Within an USAR environment this distance is a good result. If a rescuer were able to be directed to within 1.1m of a survivor there is a high probability that the survivor could be found and therefore extracted from the disaster zone.
6 Conclusion

The research undertaken in this thesis has aimed to advance both the practical and theoretical applicability of radio controlled robots in USAR scenarios. The achievements and results are presented in this chapter. This is followed by suggestions of future research that could be performed to build upon the results achieved thus far.

6.1 Contributions and Major Findings

Chapter 2 presented the design of a quadruped robot which incorporated the biometric influences of a bear. This robot was unique as it is capable of both quadrupedal and bipedal motion, a feature not provided in any commercially available robots. As a quadruped two additional DOF were incorporated into the design, one to allow the shoulders to roll and another to allow lateral movement at the hips. This is a unique feature as conventional quadruped robots, such as the Sony Aibo, have a fixed torso and as such are unable to replicate realistic quadrupedal motion. The extra DOF at the hip is also unique when the quadruped robot demonstrates bipedal motion. The ability to move the upper torso when walking as a biped is another method that can be adopted to change the Centre of Gravity (COG) and increase stability.

As discussed in Chapter 2 the quadruped robots three DOF head could either be controlled by the main processing system, HyInt, or the microcontroller mounted within the head itself, HySense Lite. Incorporated within the design of HySense Lite is a 3 axis gyroscope and a 3 axis accelerometer. The availability of these sensors allows HySense Lite to directly control the stability of the head. A stable head increases the quality of the images that can be acquired by camera by reducing the jitter. Improving acquired images in this manner has the potential to improve the performance of image processing techniques implemented within HyInt.

The modular system design of the quadruped robot incorporated custom designs for the main processing module, the two microcontroller based data acquisition modules, the
OLED eye module and the QTouch sensor module. At the time these modules were designed there were no equivalent commercially available products necessitating the need to have these custom designed and manufactured. The quadruped robots modular system design integrated with a network of Dynamixel modules has since been emulated by commercially successful robots, such as the DARwIn-OP manufactured by Robotis.

The design of the tank based robot increased personal knowledge relating to track based rescue robot systems but for the most part followed existing protocols for both manufacture and design. However, the design of the Drop Node Deployment System (DNDS) was unique. The DNDS incorporated a custom designed linear actuator to deploy Drop Nodes (DN) as well as implementing a novel solution to allow the DNs to determine when they had been released from the DNDS. The ability to detect deployment allowed DNs to determine when it was necessary to join the wireless network, a feature critical to the successful operation of the wireless robotic guidance system.

Finally the major contribution to the body of knowledge in USAR robotics has been the development of wireless robotic guidance system presented in Chapters 3, 4 and 5. Using a commercially available 802.15.4 module, which incorporates a 32-bit microcontroller and a 802.15.4 transceiver, the wireless robotic guidance system is able to dynamically adapt to the environment in which it is deployed. Modifying a pre-existing localisation algorithm ‘Trilateration using Clustering and Centroids’ [69] and utilising RSSI and TOF based distance measurements the wireless robotic guidance system was able to localise a wireless node to an average of 1.32m outdoors and 1.07m in an indoor environment similar to an anticipated disaster zone created from a collapsed building.

This research has advanced the field of knowledge in USAR robotics and provides opportunities that could be taken up by private industry, but it also raises opportunities for future research.
6.2 Future Research

All research works are conducted within the parameters of both time and funds available. This section discusses some of the potential research possibilities that were revealed in the process of undertaking this research but were not able to be pursued despite their promise.

Further research projects involving the quadruped robot could include:

- **Walking gaits**: further work could be conducted on both quadrupedal and bipedal walking gaits for the quadruped robot. Initial work was performed that created a very fundamental quadrupedal walk using inverse kinematics to create elliptical trajectories (this work was not documented within this thesis). The 2 extra DOF that allow shoulder roll and side-to-side hip movement could be incorporated into a quadrupedal walking gait which would result in a walk similar to a bear. Bipedal walking gaits are possible with this quadruped and the research could be conducted to incorporate lateral movement of the torso at the hip to change Centre of Gravity and increase stability of bipedal walking gaits.

- **Head Stability**: developing algorithms to control the stability of the 3 DOF head using a 3 axis gyroscope and a 3 axis accelerometer as the operational parameters.

- **Autonomy**: develop algorithms to allow the quadruped to display autonomous motions over rough terrain. This would require developing algorithms to implement stability as well as the addition of contact sensors within the quadruped robots feet.

Further research projects involving the tank based robot could include:

- **Autonomy**: develop algorithms to allow the tank based robot to move autonomously through a cluttered environment. This would require the tank based robot to perceive its environment and could be achieved by either image processing techniques, to detect and avoid obstacles, or the addition of a Laser Range Finder (LRF) to provide a 3D representation of the tank based robots surrounds.
Further research projects involving the both quadruped robot and tank based robot could include:

- It was proposed within the research that the tank based robot follow the quadruped robot through the disaster zone. Implementing algorithms that allowed the tank based robot to autonomously follow the quadruped could form the basis of an interesting research project. The tank based robot could either achieve this by keeping visual contact with the quadruped robot or by using localisation provided by the wireless robotic guidance system.

Further research projects involving the wireless robot guidance system could include:

- This research proposed that camera images be transferred with the ACK packets of the wireless robotic guidance system. Research could be conducted into image compression techniques across a 250kbps wireless communication.
- Research could be conducted to improve the reliability of the TOF measurements by adding digital filters to the TOF measurements.
- Localisation improvements within the wireless robotic guidance system could provide an interesting research project. Adding localisation from other sources, such as encoders and magnetic compasses, could increase the accuracy achieved by this research.

This research is significant because it is a part of a worldwide effort by a community of researches whose aim is to increase human safety through rescue robotics and has added to the body of knowledge.
Appendix A – Robotis Dynamixel Specifications

<table>
<thead>
<tr>
<th>Specification</th>
<th>RX-64</th>
<th>RX-28</th>
<th>RX-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>14.8V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standby Current</td>
<td>50mA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stall Torque</td>
<td>5.3 N.m</td>
<td>3.7 N.m</td>
<td>1.3 N.m</td>
</tr>
<tr>
<td>No Load Speed</td>
<td>64rpm</td>
<td>85rpm</td>
<td>54rpm</td>
</tr>
<tr>
<td>Reduction Ratio</td>
<td>1/200</td>
<td>1/193</td>
<td>1/193</td>
</tr>
<tr>
<td>Operating Angle</td>
<td>300°</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Angular Resolution</td>
<td>0.29°</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>40.2 mm</td>
<td>35.6 mm</td>
<td>35.6 mm</td>
</tr>
<tr>
<td>Length</td>
<td>61.1 mm</td>
<td>50.6 mm</td>
<td>50.6 mm</td>
</tr>
<tr>
<td>Depth</td>
<td>41.0 mm</td>
<td>35.5 mm</td>
<td>35.5 mm</td>
</tr>
<tr>
<td>Weight</td>
<td>125g</td>
<td>72g</td>
<td>64.5g</td>
</tr>
<tr>
<td>Motor</td>
<td>Swiss Maxon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A-1 RX series Dynamixel Specifications

Appendix B – Quadruped Robot Schematics

Appendix B contains all the schematics for PCB’s used within the Quadruped USAR robot presented in Section 2.

- B.1 HyInt Motherboard Interface (PB0803)
- B.2 HyInt Power Supply Board GPIO
- B.3 HyInt Power Supply Board Schematics (PB0803)
- B.4 HySense GPIO
- B.5 HySense (PB0806)
- B.6 HySense Lite GPIO
- B.7 HySense Lite (PB0805)
- B.8 OLED Display Board (PB0701)
- B.9 Touch Sensor Module (PB0702)
Appendix B.1 HyInt Motherboard Interface (PB0803)
## Appendix B.2 HyInt Power Supply Board GPIO

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>PD0 I/O</td>
</tr>
<tr>
<td>13</td>
<td>PB1 O</td>
</tr>
<tr>
<td>14</td>
<td>PB2 O</td>
</tr>
<tr>
<td>15</td>
<td>PB3 -</td>
</tr>
<tr>
<td>16</td>
<td>PB4 -</td>
</tr>
<tr>
<td>17</td>
<td>PB5 -</td>
</tr>
<tr>
<td>18</td>
<td>PB6 I</td>
</tr>
<tr>
<td>19</td>
<td>PB7 I</td>
</tr>
<tr>
<td>20</td>
<td>PC0 I</td>
</tr>
<tr>
<td>21</td>
<td>PC1 I</td>
</tr>
<tr>
<td>22</td>
<td>PC2 I</td>
</tr>
<tr>
<td>23</td>
<td>PC3 I</td>
</tr>
<tr>
<td>24</td>
<td>PC4 I</td>
</tr>
<tr>
<td>25</td>
<td>PC5 O</td>
</tr>
<tr>
<td>26</td>
<td>PC6 -</td>
</tr>
<tr>
<td>27</td>
<td>PC7 -</td>
</tr>
<tr>
<td>28</td>
<td>PD0 -</td>
</tr>
<tr>
<td>29</td>
<td>PD1 -</td>
</tr>
<tr>
<td>30</td>
<td>PD2 O</td>
</tr>
<tr>
<td>31</td>
<td>PD3 O</td>
</tr>
<tr>
<td>32</td>
<td>PD4 I</td>
</tr>
<tr>
<td>33</td>
<td>PD5 I/O</td>
</tr>
<tr>
<td>34</td>
<td>PD6 I/O</td>
</tr>
<tr>
<td>35</td>
<td>PD7 I/O</td>
</tr>
</tbody>
</table>

### HyInt Intelligent Power Supply (PB0803)

- Pin Description:
  - **I/O**: Input/Output
  - **O**: Output
  - **-**: Not defined
  - **I**: Input
  - **J1**: GPIO Spine
  - **J2**: GPIO Spine
  - **J3**: GPIO Spine
  - **J4**: GPIO Spine
  - **J5**: GPIO Spine
  - **J6**: GPIO Spine
  - **J7**: GPIO Spine
  - **J8**: GPIO Spine

- **GPIO Spine**:
  - **12 PD0**: 5 GPIO Spine
  - **13 PB1**: LED 0
  - **14 PB2**: LED 1
  - **15 PB3**: Switch 1
  - **16 PB4**: Switch 2
  - **17 PB5**: Switch 3
  - **18 PB6**: DIP Switch 1
  - **19 PB7**: DIP Switch 2
  - **20 PC0**: ADC0 Voltage Sense
  - **21 PC1**: ADC1 Current Sense Electronics
  - **22 PC2**: ADC2 Current Sense Motor
  - **23 PC3**: ADC3 ATI/EL-PWR
  - **24 PC4**: ADC4 GLX-SLEEP
  - **25 PC5**: ADC5 GLX-PWR
  - **26 PC6**: ADC6
  - **27 PC7**: ADC7

- **GPIO Spine**:
  - **28 PD0**: RXD CM-IGLX
  - **29 PD1**: TXD CM-GLK
  - **30 PD2**: RC-ELEC
  - **31 PD3**: RC MOTOR
  - **32 PD4**: DIP Switch 2
  - **33 PD5**: J1 4 GPIO Spine
  - **34 PD6**: J1 5 GPIO Spine
  - **35 PD7**: J1 6 GPIO Spine
Appendix B.3 HyInt Power Supply Board Schematics (PB0803)
Ad-Hoc Wireless MESH Network Implementation for Rescue Robots
Appendix B.4 HySense GPIO

<table>
<thead>
<tr>
<th>Pin</th>
<th>HySense Lite Board (PB0805)</th>
</tr>
</thead>
<tbody>
<tr>
<td>51 PA0</td>
<td>U12  5 Diff1 (U5 - GP2D2) Enb</td>
</tr>
<tr>
<td>50 PA1</td>
<td>U13  5 Diff2 (U5 - GP2D2) Enb</td>
</tr>
<tr>
<td>49 PA2</td>
<td>U9   10 A2D - ShuttDown</td>
</tr>
<tr>
<td>48 PA3</td>
<td>U8   16 A2D - Busy</td>
</tr>
<tr>
<td>47 PA4</td>
<td>U8   18 A2D - BS</td>
</tr>
<tr>
<td>46 PA5</td>
<td>J13  4 GPIO</td>
</tr>
<tr>
<td>45 PA6</td>
<td>J13  5 GPIO</td>
</tr>
<tr>
<td>44 PA7</td>
<td>J13  6 GPIO</td>
</tr>
<tr>
<td>10 PB0</td>
<td>-</td>
</tr>
<tr>
<td>11 PB1</td>
<td>J11  6 SCK</td>
</tr>
<tr>
<td>12 PB2</td>
<td>J11  5 MOSI</td>
</tr>
<tr>
<td>13 PB3</td>
<td>-</td>
</tr>
<tr>
<td>14 PB4</td>
<td>U4   6 Select TTL (Axx or Dxx/RX)</td>
</tr>
<tr>
<td>15 PB5</td>
<td>-</td>
</tr>
<tr>
<td>16 PB6</td>
<td>U1   3 PC TxDEnb</td>
</tr>
<tr>
<td>17 PB7</td>
<td>-</td>
</tr>
<tr>
<td>35 PC0</td>
<td>J12  3 Cap Sensor OUT1</td>
</tr>
<tr>
<td>36 PC1</td>
<td>J12  4 Cap Sensor OUT2</td>
</tr>
<tr>
<td>37 PC2</td>
<td>J12  5 Cap Sensor OUT3</td>
</tr>
<tr>
<td>38 PC3</td>
<td>J12  6 Cap Sensor GND</td>
</tr>
<tr>
<td>39 PC4</td>
<td>J12  7 Cap Sensor ShuttDown</td>
</tr>
<tr>
<td>40 PC5</td>
<td>J12  8 Cap Sensor LED1</td>
</tr>
<tr>
<td>41 PC6</td>
<td>J12  9 Cap Sensor LED2</td>
</tr>
<tr>
<td>42 PC7</td>
<td>J12  10 Cap Sensor LED3</td>
</tr>
<tr>
<td>25 PD0</td>
<td>BCL</td>
</tr>
<tr>
<td>26 PD1</td>
<td>SDA</td>
</tr>
<tr>
<td>27 PD2</td>
<td>RXD0</td>
</tr>
<tr>
<td>28 PD3</td>
<td>TXD0</td>
</tr>
<tr>
<td>29 PD4</td>
<td>-</td>
</tr>
<tr>
<td>30 PD5</td>
<td>-</td>
</tr>
<tr>
<td>31 PD6</td>
<td>-</td>
</tr>
<tr>
<td>32 PD7</td>
<td>-</td>
</tr>
<tr>
<td>2 PE0</td>
<td>RXD0 (AX or DX/RX)</td>
</tr>
<tr>
<td>3 PE1</td>
<td>TXD0 (AX or DX/RX)</td>
</tr>
<tr>
<td>4 PE2</td>
<td>TxDEnb (AX or DX/RX)</td>
</tr>
<tr>
<td>5 PE3</td>
<td>-</td>
</tr>
<tr>
<td>6 PE4</td>
<td>-</td>
</tr>
<tr>
<td>7 PE5</td>
<td>-</td>
</tr>
<tr>
<td>8 PE6</td>
<td>-</td>
</tr>
<tr>
<td>9 PE7</td>
<td>-</td>
</tr>
<tr>
<td>61 PF0</td>
<td>ADC0 U9  9 5-axis IMU AX</td>
</tr>
<tr>
<td>60 PF1</td>
<td>ADC1 U9  8 5-axis IMU AY</td>
</tr>
<tr>
<td>59 PF2</td>
<td>ADC2 U9  7 5-axis IMU AZ</td>
</tr>
<tr>
<td>58 PF3</td>
<td>ADC3 U9  5 5-axis IMU REF</td>
</tr>
<tr>
<td>57 PF4</td>
<td>ADC4 U9  3 5-axis IMU GX</td>
</tr>
<tr>
<td>56 PF5</td>
<td>ADC5 U9  4 5-axis IMU GY</td>
</tr>
<tr>
<td>55 PF6</td>
<td>ADC8 J1  1 GP2D2 Distance (GPIO-A0 En)</td>
</tr>
<tr>
<td>54 PF7</td>
<td>ADC7 J1  1 GP2D2 Distance (GPIO-A1 En)</td>
</tr>
<tr>
<td>33 PG0</td>
<td>HEALTHY LED</td>
</tr>
<tr>
<td>34 PG1</td>
<td>-</td>
</tr>
<tr>
<td>43 PG2</td>
<td>-</td>
</tr>
<tr>
<td>18 PG3</td>
<td>-</td>
</tr>
<tr>
<td>19 PG4</td>
<td>-</td>
</tr>
</tbody>
</table>
Appendix B.5 HySense (PB0806)
## Appendix B.6 HySense Lite GPIO

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>SCK</td>
</tr>
<tr>
<td>P1</td>
<td>MOSI</td>
</tr>
<tr>
<td>P2</td>
<td>MISO</td>
</tr>
<tr>
<td>P3</td>
<td>U6 6 Select-TTL (AX or DX/RX)</td>
</tr>
<tr>
<td>P4</td>
<td>U1 3 PC-TxEmb</td>
</tr>
<tr>
<td>P6</td>
<td>J16 5 P87 SPI Ext</td>
</tr>
<tr>
<td>P8</td>
<td>J11 3 Cap SensorOUT1 J1 OUT1</td>
</tr>
<tr>
<td>P9</td>
<td>J11 4 Cap SensorOUT2 J1 4 OUT2</td>
</tr>
<tr>
<td>P10</td>
<td>J11 5 Cap SensorOUT3 J1 5 OUT3</td>
</tr>
<tr>
<td>P11</td>
<td>J11 6 Cap SensorGND J1 6 GND</td>
</tr>
<tr>
<td>P12</td>
<td>J11 7 Cap SensorShutDown J1 7 ShutDown</td>
</tr>
<tr>
<td>P13</td>
<td>J11 8 Cap SensorLED1 J1 8 LED1</td>
</tr>
<tr>
<td>P14</td>
<td>J11 9 Cap SensorLED2 J1 9 LED2</td>
</tr>
<tr>
<td>P15</td>
<td>J11 10 Cap SensorLED3 J1 10 LED3</td>
</tr>
<tr>
<td>P0</td>
<td>SCL</td>
</tr>
<tr>
<td>P1</td>
<td>SDA</td>
</tr>
<tr>
<td>P2</td>
<td>RXD1</td>
</tr>
<tr>
<td>P3</td>
<td>TXD1</td>
</tr>
<tr>
<td>P6</td>
<td>J19 3 SCL SWI-START</td>
</tr>
<tr>
<td>P7</td>
<td>J19 4 SDA Zg-100 - ZB Status</td>
</tr>
<tr>
<td>P8</td>
<td>J4 2 RXD1 RS485, Zg100, J3 (Audio Jack)</td>
</tr>
<tr>
<td>P9</td>
<td>J4 4 TXD1 RS485, Zg100, J3 (Audio Jack)</td>
</tr>
<tr>
<td>P10</td>
<td>J2 4 Zg100 - ZB Reset</td>
</tr>
<tr>
<td>P0</td>
<td>RXD0</td>
</tr>
<tr>
<td>P1</td>
<td>TXD0</td>
</tr>
<tr>
<td>P2</td>
<td>TxEmb (AX or DX/RX)</td>
</tr>
<tr>
<td>P3</td>
<td>5 P33</td>
</tr>
<tr>
<td>P4</td>
<td>6 SW-UP</td>
</tr>
<tr>
<td>P5</td>
<td>7 SW-DOWN</td>
</tr>
<tr>
<td>P6</td>
<td>8 SW-LEFT</td>
</tr>
<tr>
<td>P7</td>
<td>9 SW-RIGHT</td>
</tr>
<tr>
<td>P0</td>
<td>ADC0 U11 9 5-axis IMU AX</td>
</tr>
<tr>
<td>P1</td>
<td>ADC1 U11 8 5-axis IMU AY</td>
</tr>
<tr>
<td>P2</td>
<td>ADC2 U11 7 5-axis IMU AZ</td>
</tr>
<tr>
<td>P3</td>
<td>ADC3 U11 5 5-axis IMU REF</td>
</tr>
<tr>
<td>P4</td>
<td>ADC4 U11 3 5-axis IMU GX</td>
</tr>
<tr>
<td>P5</td>
<td>ADC5 U11 4 5-axis IMU GY</td>
</tr>
<tr>
<td>P6</td>
<td>ADC6 U5 1 GP2D2 Distance (GPIO-A0 En)</td>
</tr>
<tr>
<td>P7</td>
<td>ADC7 U6 1 GP2D2 Distance (GPIO-A1 En)</td>
</tr>
<tr>
<td>P0</td>
<td>PG0</td>
</tr>
<tr>
<td>P1</td>
<td>PG1</td>
</tr>
<tr>
<td>P2</td>
<td>PG2</td>
</tr>
<tr>
<td>P3</td>
<td>PG3 I/O</td>
</tr>
<tr>
<td>P4</td>
<td>PG4 I/O</td>
</tr>
</tbody>
</table>

Ad-Hoc Wireless MESH Network Implementation for Rescue Robots

p 223
Appendix B.7 HySense Lite (PB0805)
Appendix B.8 OLED Display Board (PB0701)
Appendix B.9 Touch Sensor Module (PB0702)
## Appendix C – HyInt to HySense Communication Protocol

Definition of OpCodes used for HyInt to HySense and HySense Lite communications.

<table>
<thead>
<tr>
<th>OpCode</th>
<th>HySense</th>
<th>HySense Lite</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>Y</td>
<td></td>
<td>Read Distance Measurement Sensor</td>
</tr>
<tr>
<td>0x02</td>
<td>Y</td>
<td></td>
<td>Read Accelerometer</td>
</tr>
<tr>
<td>0x04</td>
<td>Y</td>
<td></td>
<td>Read specific Analog Input</td>
</tr>
<tr>
<td>0x05</td>
<td>Y</td>
<td></td>
<td>Read ALL Analog Inputs</td>
</tr>
<tr>
<td>0x09</td>
<td>Y</td>
<td></td>
<td>Read Gyro</td>
</tr>
<tr>
<td>0x0A</td>
<td>Y</td>
<td>N</td>
<td>Read Micromag Compass</td>
</tr>
<tr>
<td>0x0B</td>
<td>Y</td>
<td>N</td>
<td>Set RGB LED</td>
</tr>
<tr>
<td>0x12</td>
<td>Y</td>
<td></td>
<td>Set IO Direction</td>
</tr>
<tr>
<td>0x13</td>
<td>Y</td>
<td></td>
<td>Write IO</td>
</tr>
<tr>
<td>0x14</td>
<td>Y</td>
<td></td>
<td>Read IO</td>
</tr>
<tr>
<td>0x15</td>
<td>Y</td>
<td></td>
<td>Distance Measurement Sensor Power</td>
</tr>
<tr>
<td>0x16</td>
<td>N</td>
<td>Y</td>
<td>Set Touch Panel</td>
</tr>
<tr>
<td>0x20</td>
<td>Y</td>
<td>Y</td>
<td>Update Main Function</td>
</tr>
<tr>
<td>0x22</td>
<td>Y</td>
<td>Y</td>
<td>Read Main Version</td>
</tr>
<tr>
<td>0x24</td>
<td>Y</td>
<td>Y</td>
<td>Shutdown</td>
</tr>
<tr>
<td>0x26</td>
<td>Y</td>
<td>Y</td>
<td>Read EEPROM</td>
</tr>
<tr>
<td>0x27</td>
<td>Y</td>
<td>Y</td>
<td>Write EEPROM</td>
</tr>
<tr>
<td>0x28</td>
<td>Y</td>
<td>Y</td>
<td>Read module ID</td>
</tr>
<tr>
<td>0x29</td>
<td>Y</td>
<td>Y</td>
<td>Write module ID</td>
</tr>
<tr>
<td>0x30</td>
<td>N</td>
<td>Y</td>
<td>OLED Initialise</td>
</tr>
<tr>
<td>0x31</td>
<td>N</td>
<td>Y</td>
<td>OLED Display Mode</td>
</tr>
<tr>
<td>0x45</td>
<td>N</td>
<td>Y</td>
<td>OLED Clear</td>
</tr>
<tr>
<td>0x66</td>
<td>N</td>
<td>Y</td>
<td>OLED Move Eyes</td>
</tr>
<tr>
<td>0x67</td>
<td>N</td>
<td>Y</td>
<td>OLED Draw Eyes</td>
</tr>
</tbody>
</table>

Table C-1 HyInt to HySense communication protocol OpCodes
Appendix D – MMP-40 Tank base Specifications

<table>
<thead>
<tr>
<th>Body Style/Chassis:</th>
<th>Aluminum, black anodized. Water Resistant.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions:</td>
<td>69.85 cm x 50.80cm x 19.05 cm (L x W x H)</td>
</tr>
<tr>
<td>Weight:</td>
<td>18.14 kg</td>
</tr>
<tr>
<td>Payload:</td>
<td>13.61 kg</td>
</tr>
<tr>
<td>Power:</td>
<td>24V 7.2 Ah Rechargeable NiCd Battery</td>
</tr>
<tr>
<td>Run Time:</td>
<td>Approximately 2 hours of continuous operation.</td>
</tr>
<tr>
<td>Drive:</td>
<td>2-Motors, 100W each</td>
</tr>
<tr>
<td></td>
<td>24V Gearhead motors, all metal gears with custom secondary</td>
</tr>
<tr>
<td></td>
<td>heavy duty gearboxes and axles. Steel gears with full ball</td>
</tr>
<tr>
<td></td>
<td>bearings.</td>
</tr>
<tr>
<td></td>
<td>500 CPR, 3-channel quadrature optical encoders...</td>
</tr>
<tr>
<td>Wheel Details:</td>
<td>19.05 cm diameter UHMW spiral spoked drive sprockets for</td>
</tr>
<tr>
<td></td>
<td>shock absorption.</td>
</tr>
<tr>
<td>Track Details:</td>
<td>Custom manufactured light weight and flexible rubber tracks with all-terrain tread patterns.</td>
</tr>
<tr>
<td>Speed:</td>
<td>76.2 cm/sec.</td>
</tr>
<tr>
<td>Steering:</td>
<td>Tank steer, differential type, zero turning radius</td>
</tr>
<tr>
<td>Suspension:</td>
<td>Semi-flexible spiral spoked UHMW drive sprockets with rubber tracks</td>
</tr>
<tr>
<td>Motor Controllers:</td>
<td>Custom High Current Motor drivers with thermal and over current protection. Motor drivers are controlled with a standard Hobby-type PWM signal or with RS-232 serial control.</td>
</tr>
<tr>
<td>Specifications:</td>
<td>6-50 Volt, 30A continuous</td>
</tr>
</tbody>
</table>

Table D-1 MMP-40 Tank base Specifications
Appendix E – FitPC-2 Specifications

<table>
<thead>
<tr>
<th>CPU</th>
<th>Intel Z530 1.6GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chipset</td>
<td>Intel US15W SCH</td>
</tr>
<tr>
<td>Memory</td>
<td>1GB / 2GB DDR2-533 on-board</td>
</tr>
<tr>
<td>Storage</td>
<td>8GB SSD hard drive</td>
</tr>
<tr>
<td></td>
<td>miniSD socket</td>
</tr>
<tr>
<td>Display &amp; Graphics</td>
<td>Intel GMA500 graphics acceleration</td>
</tr>
<tr>
<td></td>
<td>Full hardware video acceleration of H.264, MPEG2, VC1, and WMV9</td>
</tr>
<tr>
<td></td>
<td>DVI Digital output up to 1920 x 1200 through HDMI connector</td>
</tr>
<tr>
<td>Audio</td>
<td>Line-out, line-in, mic</td>
</tr>
<tr>
<td></td>
<td>5.1 Channels S/PDIF</td>
</tr>
<tr>
<td>Networking</td>
<td>1000 BaseT Ethernet</td>
</tr>
<tr>
<td></td>
<td>802.11n WLAN (based on RaLink RT3090, single antenna)</td>
</tr>
<tr>
<td>USB</td>
<td>6 USB 2.0 High Speed ports</td>
</tr>
<tr>
<td>BIOS</td>
<td>Phoenix BIOS</td>
</tr>
<tr>
<td></td>
<td>Bootable from HDD, USB thumb drive, USB CDROM, USB hard disk, over network</td>
</tr>
</tbody>
</table>

Table E-1 FitPC-2 Specifications
Appendix F – Wireless Node Schematics
Ad-Hoc Wireless MESH Network Implementation for Rescue Robots
Appendix G – Localisation Results

Appendix G.1 Detailed Outdoor Localisation Results

Point No.1

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 12.000000, 12.500000, 0.000000, 12.500000, 9.355347, 9.355347, 11.926861
1, 4, 12.460745, 11.335039, 1.252765, 9.063100, 5.775400, 10.270400, 9.531300

dx: -0.460745
dy: 1.164961
dD: 1.252765

Centroid Data:
0 13.197507 9.655378
1 12.957262 9.012404
2 14.402889 8.738758
3 9.746067 16.768652
1 RESULT of Centroid: 12.460745 11.335039

Point No.4

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 19.000000, 12.000000, 0.000000, 13.892444, 15.925529, 5.442656, 6.708204
1, 3, 18.764927, 10.569664, 1.449524, 10.839200, 16.258301, 4.781500, 4.357200

dx: 0.235073
dy: 1.430336
dD: 1.449524

Centroid Data:
0 19.545925 12.590111
1 17.975933 9.043036
2 18.537848 8.645508
1 RESULT of Centroid: 18.764927 10.56966
Point No. 12

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 12.000000, 4.000000, 0.000000, 4.000000, 15.563499, 15.563499, 10.198039
1, 3, 9.325134, 4.559113, 2.732675, 3.520000, 12.823800, 18.349001, 7.821500

dx: 2.674866
dy: -0.559113
dD: 2.732675

Centroid Data:
0 9.399802 2.372629
1 6.617625 4.796201
2 18.366159 -0.926115
3 4.141062 8.498725
4 5.426152 8.613235
1 RESULT of Centroid: 9.325134 4.559113

Point No. 15

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 4.000000, 6.600000, 0.000000, 10.371114, 10.750000, 19.275957, 18.009996
1, 5, 4.938648, 7.103605, 1.065213, 8.962200, 8.852400, 18.161800, 16.778601

dx: -0.938648
dy: -0.503605
dD: 1.065213

Centroid Data:
0 5.239016 6.768680
1 5.221955 5.863373
2 5.603706 6.277615
3 4.141062 8.498725
4 5.426152 8.613235
1 RESULT of Centroid: 4.938648 7.103605
Point No. 18

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 4.500000, 12.000000, 0.000000, 14.150971, 5.373314, 16.397333, 18.500000
1, 5, 5.823618, 12.143940, 1.331421, 12.338100, 4.663100, 14.568000, 17.332399

dx: -1.323618
dy: -0.143940
dD: 1.331421

Centroid Data:
0 5.209873 10.301598
1 6.116430 12.937165
2 6.047434 13.160429
3 6.268780 13.276043
4 6.799192 11.188401
1 RESULT of Centroid: 5.823618 12.143940
Appendix G.2  Detailed Indoor Localisation Results

Point No. 1

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 12.000000, 12.500000, 0.000000, 9.355347, 9.355347, 11.926861
1, 3, 12.317841, 11.796152, 0.772285, 12.800500, 13.488400, 13.378500

dx: -0.317841
dy: 0.703848
dD: 0.772285

Centroid Data:
0 9.072150 9.443103
1 11.434870 6.930043
2 16.764341 18.311462
3 19.739157 15.639781
4 20.298758 10.379097
5 19.831501 9.374625
1 RESULT of Centroid: 12.317841 11.796152

Point No. 2

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 15.000000, 14.000000, 0.000000, 11.498804, 6.018513, 10.630146
1, 3, 14.609266, 14.547510, 0.672637, 11.381500, 6.914100, 12.231700

dx: 0.390734
dy: -0.547510
dD: 0.672637

Centroid Data:
0 14.554180 13.089973
1 13.554467 14.848021
2 15.328418 16.252047
3 19.739157 15.639781
4 20.298758 10.379097
5 19.831501 9.374625
1 RESULT of Centroid: 14.609266 14.547510
Point No. 3

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 19.000000, 14.000000, 0.000000, 15.652476, 15.369532, 3.496070, 8.544003
1, 4, 19.448742, 12.884348, 1.202518, 15.090000, 15.831800, 1.730000, 8.365300

dx: -0.448742
dy: 1.115652
dD: 1.202518

Centroid Data:
0 19.467379 13.972709
1 18.521757 13.607895
2 19.282629 13.216333
3 19.739157 15.639781
4 20.298758 10.379097
5 19.831501 9.374625
1 RESULT of Centroid: 19.448742 12.884348

Point No. 4

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 19.000000, 12.000000, 0.000000, 13.892444, 15.925529, 5.442656, 6.708204
1, 6, 19.343012, 10.604990, 1.436562, 12.215400, 17.726900, 6.977300, 5.330800

dx: -0.343012
dy: 1.395010
dD: 1.436562

Centroid Data:
0 19.046919 10.438101
1 17.899359 10.696428
2 20.605516 11.145179
3 18.719025 10.201505
4 20.298758 10.379097
5 19.831501 9.374625
1 RESULT of Centroid: 19.343012 10.604990
Point No. 5

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 17.000000, 9.000000, 0.000000, 10.295630, 15.450647, 8.872570, 5.830952
1, 5, 18.185137, 9.129215, 1.192160, 13.833600, 14.571900, 12.374100, 5.926300

dx: -1.185137
dy: -0.129215
dD: 1.192160

Centroid Data:
0 16.086477 5.611063
1 17.767050 12.574165
2 19.728226 11.473579
3 13.850685 6.612018
4 24.540834 5.839173
5 18.322687 12.794510
1 RESULT of Centroid: 18.185137 9.129215

Point No. 6

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 19.000000, 9.000000, 0.000000, 11.401754, 17.167484, 8.409667, 4.242640
1, 6, 17.575048, 10.095644, 1.797477, 15.029700, 8.945100, 7.725800

dx: 1.424952
dy: -1.095644
dD: 1.797477

Centroid Data:
0 20.322592 8.410718
1 17.393982 10.531413
2 16.558659 9.093368
3 16.172504 11.072305
4 15.254912 9.767195
5 18.322687 12.794510
1 RESULT of Centroid: 17.575048 10.095644
Point No. 7

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 19.500000, 6.600000, 0.000000, 9.990496, 18.862993, 10.761621, 2.570992
1, 5, 20.232958, 6.758353, 0.749869, 11.450000, 19.684299, 11.261600, 1.070000

dx: -0.732958
dy: -0.158353
dD: 0.749869

Centroid Data:
0 20.145250 6.089337
1 20.937620 6.127501
2 20.881502 7.226438
3 21.243721 6.756931
4 21.622351 6.205872
5 17.300262 8.302393
1 RESULT of Centroid: 20.232958 6.758353

Point No. 8

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 18.000000, 3.000000, 0.000000, 6.708204, 14.488703, 5.000000
1, 3, 18.211979, 2.526166, 0.519090, 7.150000, 14.582500, 5.607700

dx: -0.211979
dy: 0.473834
dD: 0.519090

Centroid Data:
0 18.562410 2.838534
1 17.243086 3.030478
2 19.042419 1.235651
3 18.532721 6.551430
4 6.967612 1.540211
5 17.300262 8.302393
1 RESULT of Centroid: 18.211979 2.526166
Point No. 9

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 17.000000, 4.000000, 0.000000, 6.403124, 13.682928, 5.385165
1, 3, 17.556892, 5.309640, 1.423125, 9.920000, 13.561300, 6.410100

dx: -0.556892
dy: -1.309640
dD: 1.423125

Centroid Data:
0 15.776892 4.463022
1 21.147738 3.837356
2 16.302940 8.938182
3 18.532721 6.551430
4 6.967612 1.540211
5 17.300262 8.302393
1 RESULT of Centroid: 17.556892 5.309640

Point No. 10

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 17.000000, 6.600000, 0.000000, 8.280097, 11.160757, 5.035872
1, 3, 16.845234, 6.221906, 0.408543, 9.690000, 13.413100, 7.604100

dx: 0.154766
dy: 0.378094
dD: 0.408543

Centroid Data:
0 14.444518 5.141483
1 15.094032 9.182759
2 20.842382 3.963378
3 18.532721 6.551430
4 6.967612 1.540211
5 17.300262 8.302393
1 RESULT of Centroid: 16.845234 6.221906
**Point No. 11**

Trilateration Calculations: Peter Turner 09/03/13  
For 4 circles (2D) based on Centroids  
Max TOF Error used 0.000000

0, 0, 14.000000, 6.600000, 0.000000, 6.896376, 14.682047, 12.311072, 8.022469  
1, 4, 14.075461, 5.135157, 1.466785, 2.810000, 18.105499, 10.897800, 8.622100

dx: -0.075461  
dy: 1.464843  
dD: 1.466785

Centroid Data:  
0 13.729459 8.436955  
1 14.176816 2.375305  
2 13.821357 2.139804  
3 18.532721 6.551430  
4 6.967612 1.540211  
5 17.300262 8.302393  
1 RESULT of Centroid: 14.075461 5.135157

**Point No. 12**

Trilateration Calculations: Peter Turner 09/03/13  
For 4 circles (2D) based on Centroids  
Max TOF Error used 0.000000

0, 0, 12.000000, 4.000000, 0.000000, 4.000000, 15.563499, 10.198039  
1, 3, 12.582533, 2.120236, 1.967958, 4.180000, 19.870199, 11.948500

dx: -0.582533  
dy: 1.879764  
dD: 1.967958

Centroid Data:  
0 10.256007 3.798801  
1 15.919697 1.452024  
2 12.154425 -0.769882  
3 6.436786 8.128540  
4 6.967612 1.540211  
5 17.300262 8.302393  
1 RESULT of Centroid: 12.582533 2.120236
Point No. 13

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 10.000000, 6.600000, 0.000000, 6.896376, 12.311072, 14.682047, 12.014991
1, 6, 9.330454, 6.290105, 0.737785, 9.850000, 16.085899, 16.401100, 15.680000

dx: 0.669546
dy: 0.309895
dD: 0.737785

Centroid Data:
0 11.680002 3.215867
1 6.460432 8.093858
2 6.468082 8.149871
3 6.436786 8.128540
4 6.967612 1.540211
5 17.300262 8.302393
1 RESULT of Centroid: 9.330454 6.290105

Point No. 14

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 6.000000, 6.600000, 0.000000, 8.919641, 10.934464, 17.651133, 16.011246
1, 6, 3.742246, 5.895966, 2.364977, 11.180000, 12.790900, 19.745300, 18.645500

dx: 2.257754
dy: 0.704034
dD: 2.364977

Centroid Data:
0 4.929070 4.592886
1 3.367947 6.708022
2 3.388783 7.130172
3 3.227502 6.930778
4 3.409205 4.572752
5 1.873215 4.737154
1 RESULT of Centroid: 3.742246 5.895966
Point No. 15

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 4.000000, 6.600000, 0.000000, 10.371114, 10.750000, 19.275957, 18.009996
1, 6, 3.694321, 5.269590, 1.365075, 14.890000, 14.266700, 23.072800, 16.005301

\[ \begin{align*}
dx & : 0.305679 \\
dy & : 1.330410 \\
dD & : 1.365075
\end{align*} \]

Centroid Data:
0 6.231178 3.258849
1 1.724520 3.265934
2 -0.863978 7.498679
3 -2.194311 4.498183
4 8.407107 14.450027
5 8.555730 -2.684538
1 RESULT of Centroid: 3.694321 5.269590

Point No. 16

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 8.000000, 7.600000, 0.000000, 8.588365, 10.538619, 14.091132
1, 3, 7.632349, 8.026523, 0.563107, 11.010000, 11.248300, 12.701000

\[ \begin{align*}
dx & : 0.367651 \\
dy & : -0.426523 \\
dD & : 0.563107
\end{align*} \]

Centroid Data:
0 9.384868 7.474402
1 10.271181 10.873422
2 2.873344 6.158268
3 8.182035 16.463083
4 8.172271 16.604521
5 -0.195268 11.699804
1 RESULT of Centroid: 7.632349 8.026523
Point No. 17

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

[Centroid Data coordinates]

1 RESULT of Centroid: 5.452305 11.636000

Point No. 18

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

[Centroid Data coordinates]

1 RESULT of Centroid: 2.965351 11.665912
Point No. 19

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 6.000000, 13.000000, 0.000000, 14.317822, 4.787745, 14.660235, 17.464249
1, 6, 6.356792, 13.880793, 0.950314, 16.900000, 7.037400, 11.851200, 17.425900

dx: -0.356792
dy: -0.880793
dD: 0.950314

Centroid Data:
0 5.140202 10.405582
1 9.158564 12.563122
2 8.039740 16.429436
3 8.182035 16.463083
4 8.172271 16.604521
5 -0.195268 11.699804
1 RESULT of Centroid: 6.356792 13.880793

Point No. 20

Trilateration Calculations: Peter Turner 09/03/13
For 4 circles (2D) based on Centroids
Max TOF Error used 0.000000

0, 0, 8.000000, 13.000000, 0.000000, 5.905926, 12.764110, 15.652476
1, 3, 7.214439, 12.640974, 0.863716, 8.198600, 12.397800, 18.314899

dx: 0.785561
dy: 0.359026
dD: 0.863716

Centroid Data:
0 9.297237 11.092499
1 7.602237 17.319891
2 3.958283 9.151506
3 19.739157 15.639781
4 20.298758 10.379097
5 19.831501 9.374625
1 RESULT of Centroid: 7.214439 12.640974
References


[44] K. Young-Duk, K. Jeong-Ho, S. Duk-Han, M. Jeon-II, R. Young-Sun, and A. Jinung, "Design and implementation of user-friendly remote controllers for


[52] NXP. (2010). *Data Sheet: JN5148-001*


