Microwave Based Monitoring System for Corrosion Under Insulation

David S Herd

Submitted for the degree of Doctor of Philosophy

Heriot-Watt University

School of Engineering and Physical Sciences

Institute of Signals, Sensors and Systems

February 2016

ABSTRACT

This thesis presents the work undertaken by the author within the institute of Signals, Sensors and Systems in the school of Engineering and Physical Sciences at Heriot-Watt University, Edinburgh.

The main aim of the research was to design and develop a non-destructive sensor capable of monitoring the onset of corrosion under insulation. The development of the sensor has involved the design of a complete system to stabilise and control the sensor, the development of a COMSOL model to understand the progression of corrosion to determine the remaining useful life of the asset, and an investigation into horn antenna design to inform the design of the optimal sensor head.

The designed sensor system was tested with a variety of samples to benchmark the effectiveness of the sensor and prove the concept viability as a product. Experiments proved the concept of sensing defects in metallic surface with or without insulation layers. Samples simulating real life corrosion were tested to prove the resilience of the sensor when defects were less guaranteed.

Remaining useful life estimations were conducted on simulated defects to show the sensor ability to become a smart sensor using prognostic health management techniques.

Finally the environmental tests were conducted to ensure the sensor was indeed non-destructive, confirming that all aspects of the research had been successfully completed.

ACKNOWLEDGEMENTS

I would like to express the greatest of thanks to my academic supervisor Dr David Flynn for his help, support and encouragement throughout the past four years. I will be forever grateful for the opportunity.

Special thanks go to Mark Leonard for his overwhelming support and assistance with any mechanical task required. His machining knowhow saved me countless hours in the workshop.

I'd also like to thank my colleagues within the Institute of Signals, Sensor and Systems for the advice and assistance they have provided throughout my study. The continued support, whether it technical or personal has made the journey a little easier.

Finally, I would like to thank my friends and family for the encouragement and support throughout my research. The words "It will all be worth it in the end" have finally come through.

DECLARATION STATEMENT

TABLE OF CONTENTS

LISTS OF FIGURES

LIST OF EQUATIONS

LISTS OF TABLES

LISTS OF LIST OF PUBLICATIONS BY THE CANDIDATE

Peer reviewed journal articles

- D. Herd, D. Flynn, M.P.Y. Desmulliez , R. Lyon, P. Record, I. Andonovic "Smart Microsystems for Asset Management," Smart Systems Integration for Micro- and Nanotechnologies, Honorary volume on the occasion of Thomas Gessner's 60th birthday, pp. 307-314, August 2014

Conference Papers

- D. Herd, D. Flynn "Prognostic Health Management Techniques for Intelligent Condition Monitoring of Offshore Renewable Generation Assets", Accepted to IET Renewable Power Generation Conference (RPG), 9th – 11th September 2013, Beijing, China

- D. Flynn, D. Herd, D. Lofting, N. Skinner, P. Record, "Health and usage monitoring systems; Enabling the future prediction of remaining useful life of submarines," Accepted to the 12th International Naval and Engineering Conference (INEC), 20th – 22nd May 2014, Amsterdam, The Netherlands

- D. Flynn, D. Herd, G. Fagogenis, D. Lofting, N. Skinner, A. Campos, D. Lane, P. Record, "Intelligent asset management of submarine and ships through embedded intelligence," Accepted to the American Society of Naval Engineers (ASNE) Fleet Maintenance and Modernisation Symposium (FMMS), 9th – 10th September 2014, Virginia Beach, USA

Media Articles

- D. Herd, D. Flynn "A Review of Condition Monitoring and the Progression to Prognostic Health Management," Wind Energy Network Magazine, Operations and Maintenance Edition, pp. 66-67, October / November 2012

Patents

- M. P. Y. Desmulliez, S. K. Pavuluri, D. Flynn, D. Herd, "Microwave Cavity Sensor," Publication Number WO2013164627, November 2013

- M. P. Y. Desmulliez, S. K. Pavuluri, D. Flynn, D. Herd, "Corrosion Under Insulation Sensor," Proposed New United Kingdom Application, August 2015

Chapter 1 – INTRODUCTION

1.1 Motivation

Multiple industrial sectors have invested tens of millions in the past few decades to investigate solutions that can address the prevention, detection and repair of damage incurred by corrosion under insulation (CUI). Service companies such as AkzoNobel have invested in research to tailor and identify coating to reduce the onset of corrosion on pipe outer surfaces. [1] Exova opened a corrosion centre in 2013 at a cost of £3 million in the UK and is the central hub for further corrosion centres across the globe. [2] Despite this significant global investment CUI still represents a significant risk to a variety of high value assets due to the challenges associated with its prevention and detection. Projects

It is thought that many installations in the 1960's and 1970's were designed without thought that CUI would be a problem in future years, this coupled with assets being extended to surpass their initial design lifetime has now made CUI an increasingly prominent issue. This is most evident in the oil and gas sector where assets, such as oil rigs, were installed in the North Sea with an initial design life of 25 years; however, many of these assets are still producing oil 15-20 years beyond this intended design life.

In 2001, the United States (US) congress commissioned a research study into the annual cost associated with metallic corrosion to the US economy [3]. The study "Corrosion Costs and Preventative Strategies in the United States" conducted by the National Association of Corrosion Engineers (NACE International) found that the direct cost of corrosion was $267 billion per year. When taking into consideration the indirect cost this number doubles to a staggering 3.6% GDP. The report looked at 5 individual industry sectors, breaking each sector down further to a total of 28. The five main sectors and their share of annual cost is show in Figure 1-1.

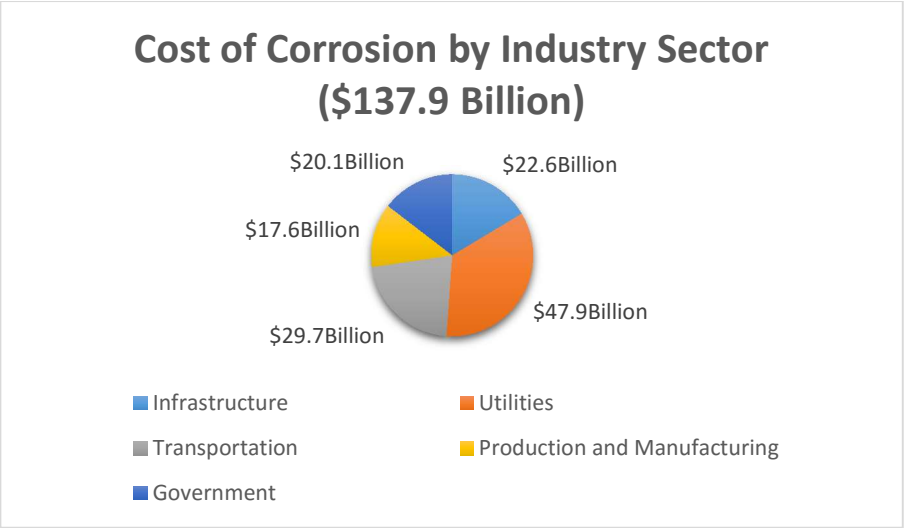


Figure 1-1 – Cost of Corrosion by Industry Sector [4]

The report provided a general overview of corrosion across all sectors, concentrating on assets and industry's critical to the US economy. For the purpose of this thesis, the main sector focus will be that of Production and Manufacturing, a detailed breakdown of this sector can be seen in Figure 1-2. Sub sectors such as Petroleum Refining, Oil and Gas Production, as well as Chemical, Petrochemical and Pharmaceutical will be of particular interest due to the nature of the assets they contain. These 3 subsectors cover 38.6% of the total cost to this main sector in the report and are the most likely to have a high percentage of costs associated with CUI [5].

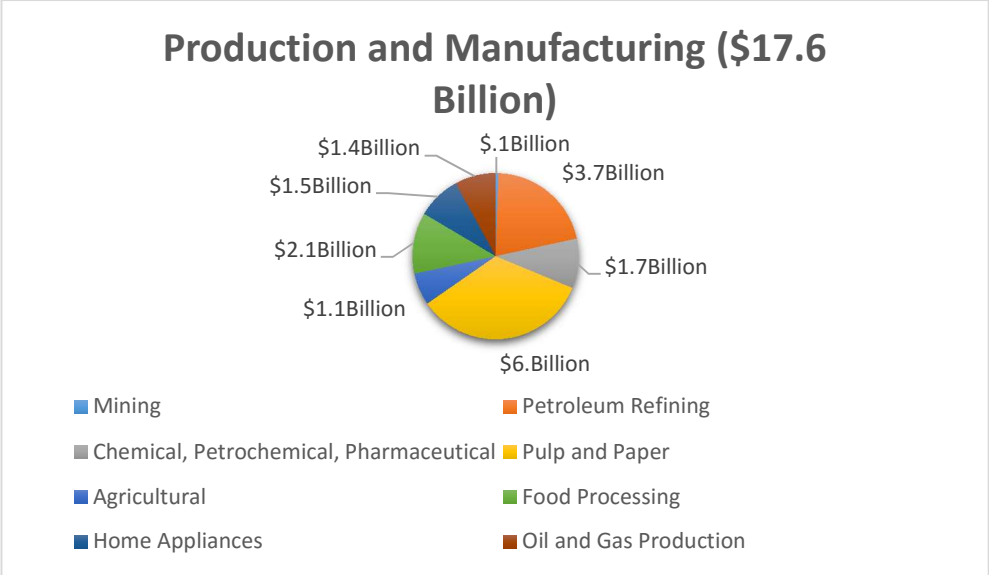


Figure 1-2 – Annual Share of Production and Manufacturing Sector [4]

A study prepared by Exxon Mobil Chemical was presented to the European Federation of Corrosion in 2003. The report determined that CUI was responsible for more leaks in the refining and chemical industries than general process corrosion. General process corrosion is considered to be corrosion caused by the process of production, i.e. fluid flow within pipes, or corrosion due to type of production. It was also found that between 40% and 60% of overall piping maintenance in this industry was related to CUI. [6]

In 2009 an updated version of the inspection code API570 was published. This inspection code from the American Petroleum Institute (API), covers the inspection, rating, repair and alteration procedures for metallic and Fibre-Reinforced Plastic (FRP) piping systems [7]. The aim of the code is to 'specify the in-service inspection and condition-monitoring program that is needed to determine the integrity of piping'. [8] The code outlines several areas susceptible to CUI with the focus primarily on the subsectors outlined previously.

Key areas of interest are:

- Areas exposed to mist overspray from cooling towers
- Areas subject to process spills, ingress of moisture or acid vapours
- Carbon steel piping systems
- Dead legs and attachments that protrude from insulated piping and operate at a temperature different than the active line
- Piping systems with deteriorated coatings and/or wrappings

This thesis outlines an innovative non-destructive evaluation (NDE) technology based on microwave sensing. The sensor represents a smart front end instrument for the application of asset health monitoring, specifically focussing on the aforementioned key areas outlined in the API code. The research within this thesis demonstrates a novel NDE method capable of monitoring, for the first time, the various dielectric and metallic layers as well as precursors to CUI onset and CUI damage itself e.g. metallic defect and metal loss. Furthermore the data from the sensor is incorporated into a Physics of Failure Model (PoF) to demonstrate a fusion health management system where data driven and model driven techniques are combined to give a comprehensive system overview. There are a number of advantageous features to the designed system, with the (NDE) sensor providing an insight to the asset health without the need to remove insulation layers, no couplant or stimulus to be applied to the asset for signature

detection, and 3D imaging capability of the asset via a quick and reliable sensing mechanism.

The aim of this thesis is to research and develop a novel sensor system to detect the presence of corrosion under insulation. To enable this achievement the chapters of this thesis will reflect the primary research objectives with respect to the design, manufacture, assembly, programming, data classification, modelling and experimental characterisation of this innovative sensor system as follows:

1- Identify the motivation for developing a new corrosion detector;
2- Perform a literature search into the state of the art corrosion identification techniques;
3- Based on literature findings design a new sensing method
4- Develop Anomaly Detection Method, "Microwave Based Monitoring System for Corrosion Under Insulation"
5- Manufacture and assemble sensor system
6- Test sensor functionality
7- Implement Remaining Useful Life estimation

1.2 Outline of Thesis

This section outlines each chapter giving a brief overview of the contents. A graphical representation of the thesis structure follows this in Figure 1-3.

Chapter 1 outlines the motivation for the thesis detailing the current industrial challenges with corrosion under insulation. The main objectives of the thesis, including the aims, are outlined before leading into subsequent chapters covering the design and experimental work in the design of the sensor system.

Chapter 2 outlines the process of corrosion, specifically how corrosion under insulation forms. An overview of the commercially available inspection techniques is outlined with an assessment of their various capabilities and limitations. An introduction to microwave sensing will conclude this chapter.

Chapter 3 presents the fundamental equations of microwave based electromagnetism, such as Maxwell's equations, followed by a summary of the theory surrounding

fundamental components, i.e. waveguide and horn antennas, of microwave sensing. Finally a description of Frequency Modulated Continuous Wave (FMCW) Radar devices is given with the basics of signal processing for such FMCW signals.

Chapter 4 describes the system design process of all fundamental components in the microwave sensor. The chapter is divided into subsections of the main components, horn and waveguide and the software used to control both the user interface and the data acquisition device.

Chapter 5 explains the experimental setup and protocol used in order to carry out the experimental work. This chapter then details each experiment and discuss results obtained. The chapter concludes with an investigation into the effects of environmental conditions as well as verification that this is indeed a Non-Destructive Evaluation monitoring system.

Chapter 6 outlines Prognostic Health Management techniques, briefly describing the concept and the benefits to any system. This is then developed into the specific benefits of implementing such a practice in this system. The chapter goes on to explain the implementation of a corrosion model in COMSOL Multiphysics, describing its use as a remaining useful life (RUL) estimation tool using experimental data.

Chapter 7 collates the primary findings of the thesis to provide an overview of the systems overall performance. The conclusions are presented to cover overall system performance against current state-of-the-art systems, before ending with suggested future work for further system improvements.

Figure 1-3 is a chapter board which outlines the chapters in more details. The figure provides a graphical representation of each chapter in terms of the main sections within it. Each row on the figure represents a single chapter, with each item on the row being a main heading of said chapter.

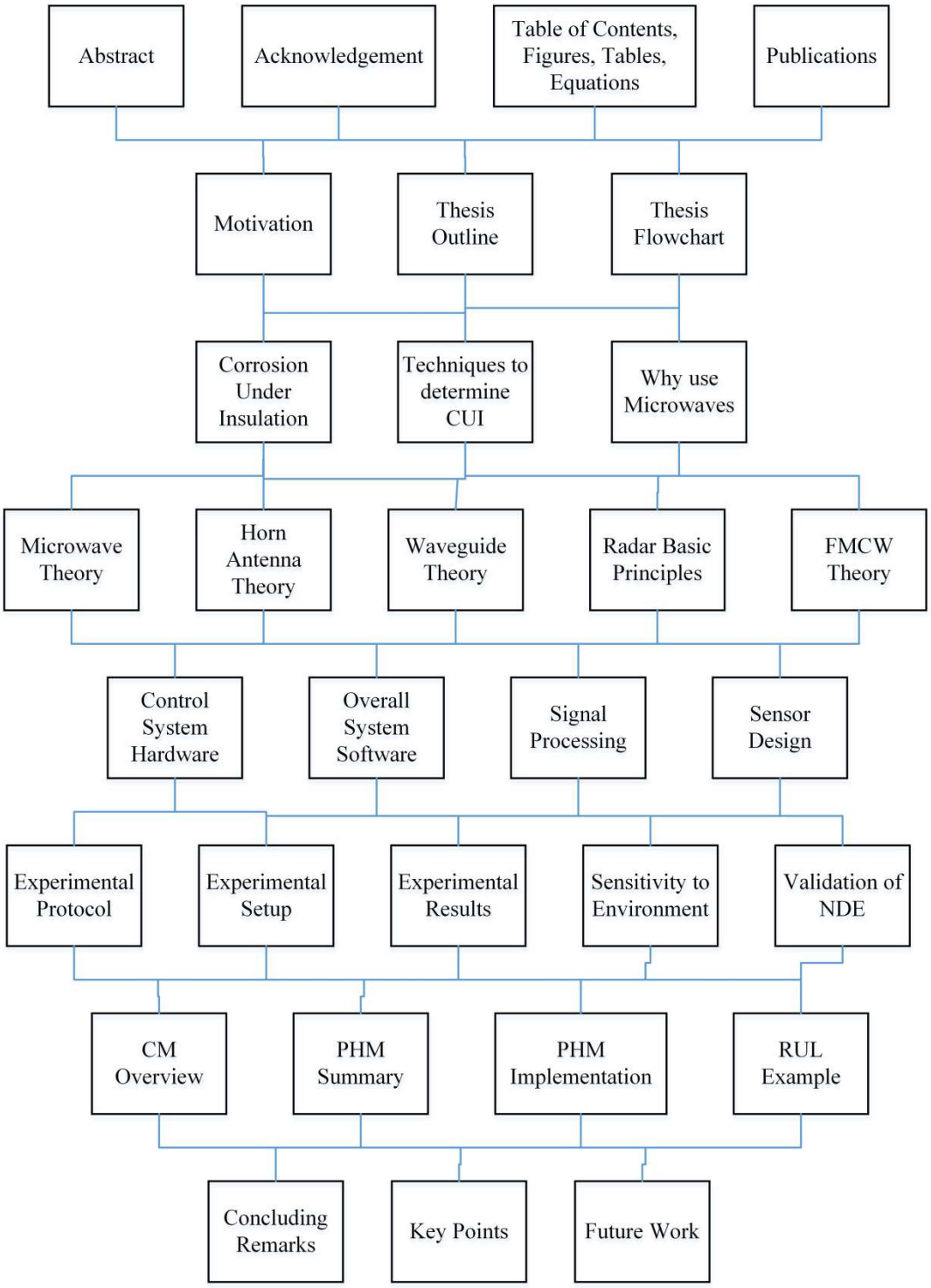


Figure 1-3 – Thesis Layout

Chapter 2– Literature Review

2.1 Corrosion Under Insulation

An explanation of the process of CUI is enabled through a summary of an oil pipeline application. Oil pipelines normally consist of three main layers, the first, is the outer protective layer which is usually a high density polyethylene (HDPE) coating or spiral steel structure to keep out environmental substances. The second is an insulation layer. There are two reasons for insulating pipelines. The first is to thermally insulate the pipes to reduce the heat loss when hot production processes are critical. The same can be said for cold flow processes, insulation prevents environmental ambient heat from penetrating the process line and causing unintended heating. The second reason for insulating pipes is from a health and safety point of view for personnel. When hot processes are used, temperatures in pipelines can exceed 150 ^{0}C which can create extremely hot surfaces close to personnel. By insulating the pipes the risk of injury is reduced from contact with the surfaces. The third layer is typically the pipe itself which may or may not have been treated with an anti-corrosion agent such as a fusion bonded epoxy (FBE) film.

Corrosion under insulation primarily occurs due to the structure of the pipelines concerned detailed in Figure 2-1.

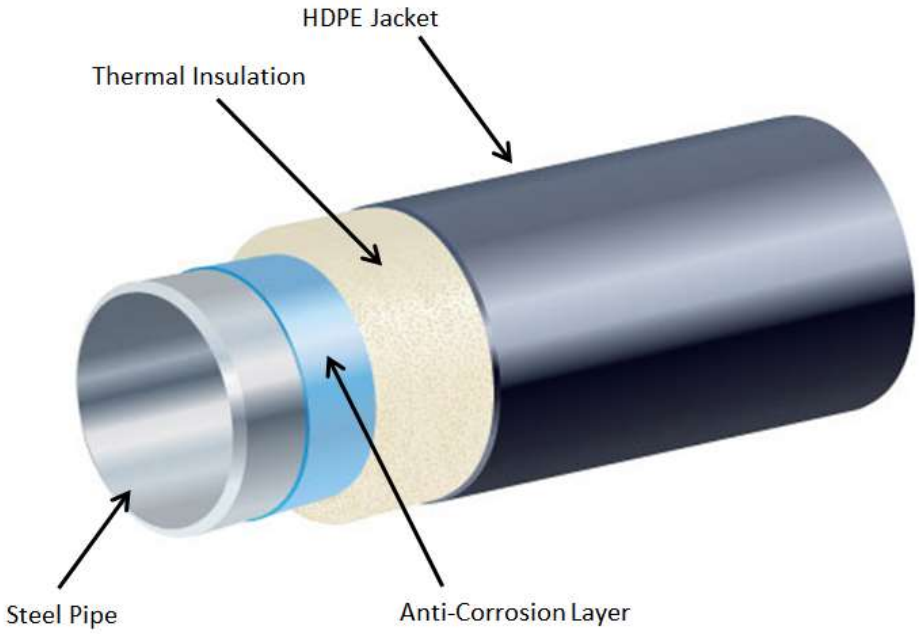


Figure 2-1 - Oil Pipe Layer Structure [9]

Water which penetrates through the outer layer settles in the micro gaps between the thermal insulation and the steel pipe surface. Here the water is in direct contact with the surface of the pipe and the process of corrosion can begin.

Steel pipes are the most susceptible to corrosion due to their high Iron content; a damaged specimen can be seen in Figure 2-2.

Figure 2-2 - Corrosion Under Insulation Example [10]

Iron is very susceptible to corrosion due to its relatively high energy level when processed. Corrosion is a way for the chemical chain to tend more towards its natural state, i.e. to lose energy. During the process of corrosion, the excess energy gained during the mining and processing phase, is lost by transferring the energy into forming either rust layers or localised pitting. The presence of moisture and oxygen automatically begins the process of the metal returning to its natural lowest state of energy.

In the process of corrosion, both the cathodic and anodic reactions occur at the same time. Therefore the process of corrosion can be slowed or reduced by changing either reaction, this can be achieved by adding protective layers to the outer metal surface to prevent the rate of the initial corrosion by limiting the contact of moisture and oxygen with the metal surface. In doing this the corrosive reaction is reduced.

The corrosion reaction of iron in oxygenated water can be represented by the following chemical equations [11]:

$$\mathbf{2}\,Fe \rightarrow \mathbf{2}\,Fe^{++} + \mathbf{4}\,e^{-} \quad anodic\ reaction \qquad \textbf{2-1}$$

$$O_2 + 2\,H_2O + 4\,e^- \rightarrow 4\,OH^-\ cathodic\ reaction$$

2-2

The overall reaction is displayed below in 2-3.

$$2\,Fe + O_2 + 2\,H_2O \rightarrow 2\,Fe^{++} + 4\,OH^- \rightarrow 2\,Fe\,(OH)_2$$

2-3

The corrosion reaction can further be explained by the corrosion mechanism shown in Figure 2-3.

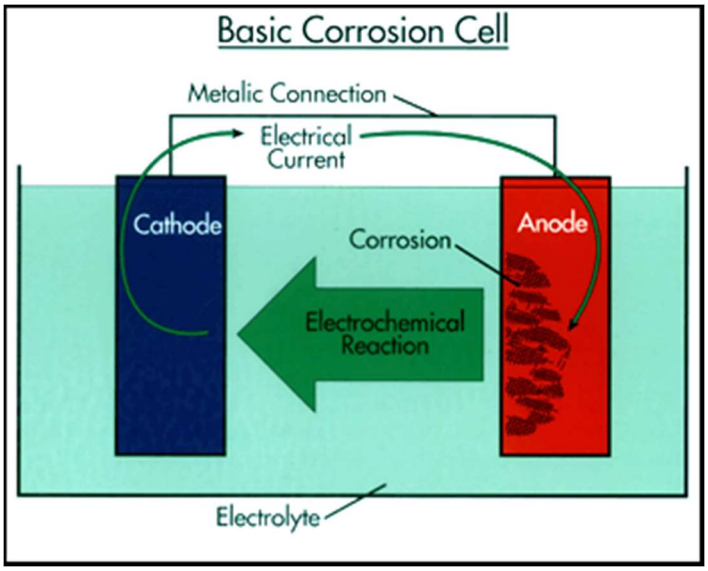


Figure 2-3 – Basic Corrosion Cell [12]

The corrosion cell consists of four main parts, an Anode, Cathode, Electrolyte and an Electrical Path. The electrolyte throughout this thesis will be considered to be water or moisture and the electrical path will be the metallic pipe connecting both the anode and cathode. [13]

The anode is the section of pipe which deteriorates from corrosion and produces electrons. The cathode is the section that does not corrode and consumes the electrons from the cathode. This establishes an electrical current flow from the cathode to the anode through the steel pipe and the electrolyte solution. The corrosion cell defines what can be described as galvanic corrosion.

Galvanic corrosion occurs when two different metals join or meet, i.e. welds. A potential difference is created between the two metals due to their different energy levels. The most active metal becomes the anode, with the least active acting as the cathode. When moisture is present, this creates a conduction path allowing electrons to flow which creates the ideal corrosion mechanism as outlined in Figure 2-3. [14, 15]

CUI can be further aggravated with the addition of chlorides. When the external protective layer becomes breached, it allows for external contaminants to enter the insulation layer. The contaminants can be from a number of sources, i.e. Acid rain and plant operations. In introducing these contaminants along with the moisture at the corrosion site, an imbalance in PH can occur further increasing the corrosion rate. The insulation itself when wet can leach chlorides into the moisture further contributing to the corrosion rate. [16]

When a metal is under tensile stress and suffering from chloride based corrosion it is defined as Chloride Stress Cracking Corrosion (CSCC). Corrosion in this form is extremely dangerous as it can lead to sudden failures due to its localised nature. [17]

Figure 2-4 – Chloride Stress Corrosion Cracking [18]

External factors such as increased loading can cause the asset to fail unexpectedly in this locally affected area.

2.1.1 Corrosion Stages

The stages of corrosion on carbon steel have been well defined into 4 stages as outlined in Table 2-1.

Stage	Corrosion Signs
1	• Paint surface has bubbles • Paint surface has broken • Possible minor metal pitting • Loose powdery deposits may be present
2	• Powder/Scale on metal surface • Condensed rust spots • Metal surface shows sign of pitting/etching
3	• Metal surface has visible defects • Metal shows uneven finish • Large rust spots and surface damage
4	• Total metal loss at point of corrosion • Stage2/3 surrounding area

Table 2-1 – Corrosion Stages

This classification of corrosion allows for industry to understand how assets are ageing across their infrastructure. [19, 20] Many older assets contain carbon steel or black steel, which are the most susceptible to corrosion, and are now exceeding their design lifetime. Newer assets which have anti-corrosion layers incorporated into the pipeline design are now also running beyond the intended operational lifetime. This means that many assets are approaching a stage where CUI could be prevalent.

Many companies installing new infrastructure are looking at new technologies and techniques to further enhance the lifetime of assets. Sacrificial layers such as zinc are added to the pipe surface to delay the degradation of the critical pipe integrity, this however only delays the inevitability of corrosion developing. Companies specialising in anti-corrosion coatings are now a global concern such as Tenaris [21] and Specialised Polymer Coatings (SPC). [22] Collaborations between large scale industrial companies is also providing new insight and technology to address corrosion, such as AkzoNobel and Siemens who are collaborating to supply anti-corrosion coating for a German power station. [23]

New steel compounds are also in development with ChromX promising anti-corrosion performance for up to 100 years.

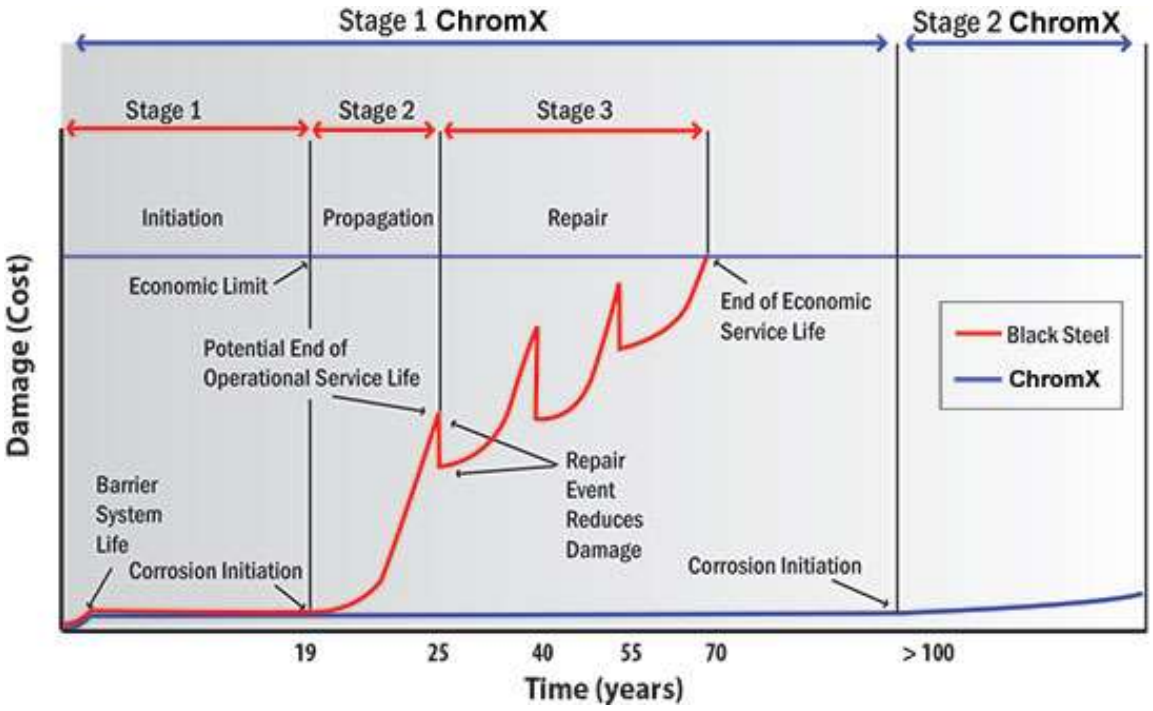


Figure 2-5 – Corrosion Vs Time [24]

The graph shown in Figure 2-5 presents a comparison between Black Steel and a new steel compound 'ChromX'. As shown, ChromX and Black Steel both have an anti-corrosion barrier system which fails at the same time, but the ChromX compound can surpass the total lifetime of Black Steel before corrosion even begins.

The black steel can resist the onset of corrosion for roughly 20 years at which point the damage to the asset begins, with an overall service life in the region of 100 years. [25] The corrosion will always increase with time, but can be slowed or superficially repaired temporarily with sanding and painting techniques. Although this technique may increase the overall lifetime, it does not halt the corrosion process and inevitably after a period the asset will ready a level where the corrosion will have weakened the steel to a point which is no longer feasible to repair in terms of cost or from a safety point of view.

Although ChromX promises to be the material of the future which answers all the problems currently associated with CUI, it does come at a cost. With the current oil price [26], investment in new technology and infrastructure is at an all-time low. The cost of new materials will decrease with demand and uptake, but at present they are not economically viable. Although potentially solving the issue of CUI in future

developments, there still needs to be a solution to determine CUI in legacy assets which are not economically viable to replace.

2.2 Techniques to determine CUI

There are a wide variety of commercially accepted systems available on the market to determine the presence of corrosion under insulation. The method and technology base used in these systems varies significantly between industry approach and application. The following subsections will detail the leading commercial solutions and outline their primary advantages and limitations in corrosion under insulation monitoring.

2.2.1 Visual Inspection

Visual inspection of a pipeline is generally considered to be the first assessment technique for identification of possible CUI. This technique although very crude, can identify serious issues which can either directly cause CUI or are of a direct consequence of CUI. Issues such as sagging cladding can indicate that water is being held in the insulation and therefore could indicate the presence of corrosion on the pipe surface. External breaks, cracks or fractures in the cladding also relate to water ingress in the insulation and should be further inspected.

The next step to visual inspection would involve the removal of the external cladding and insulation to inspect the condition of the pipe. Although effective in determining the actual condition of the pipe surface, this only provides a detailed overview of condition at the specific site. This type of sampling technique can very easily misinform the overall condition of a section of pipe due to missing smaller sections within this area with more serious symptoms.

The most effective form of visual inspection would be when whole sections of pipework had insulation removed, assessed and then new insulation installed. This of course would be highly expensive in terms of cost and time resource limiting this to small areas.

In summary, visual inspection can give a good initial overview of any structural issues which may allow the ingress of water. It is extremely low cost as no equipment is required and can be carried out quickly to identify problems. The major disadvantage is in terms of assessing extent of CUI. In order to visibly check the pipe condition, the

insulation has to be removed which is both costly and time consuming. This form of CUI inspection also has an extremely low sampling rate and could therefore misinform the actual condition of the pipe.

2.2.2 Thermography (Infrared)

Thermography is a simplistic approach used to determine the presence of moisture in a pipe as a precursor to corrosion being in existence. This, although effective in determining presence of moisture, does not give the user any insight into the condition of the asset or indeed, the quantity of the moisture present. Figure 2-6 shows a typical image of an infrared measurement when moisture is present.

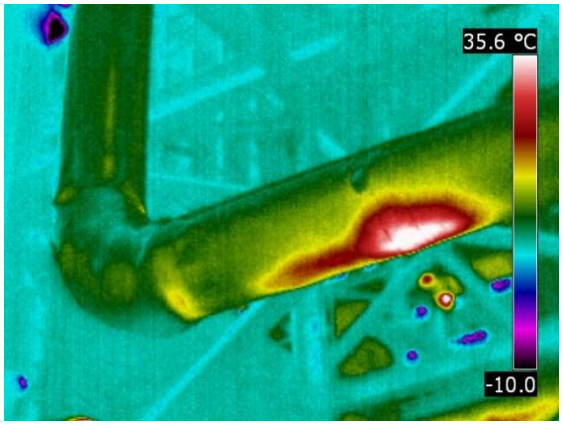


Figure 2-6 – Infrared Image of Pipeline Section [27]

This principal is only effective on hot product production lines. This approach utilises the lower temperature coefficient of water in comparison to steel. This means that water will heat quicker than the steel pipe, allowing the infrared camera to easily identify pockets of damp insulation. This method easily identifies areas of porous insulation which allow the ingress of moisture to settle within contact of the pipe surface. Reliably identifying moisture, one of the main components of corrosion, is one way of detecting CUI at or just before onset. [28]

Further to detecting water as a precursor to corrosion, it is also possible to determine defects and areas of rust due to their varying thermal conductivity when compared to the healthy baseline pipe material. Laaidi et al [29] use advanced 3D numerical analytics to determine defects areas of rust within a pipe section. The infrared images are analysed to determine the presence of circular rust areas. It was noted from this piece of research that the rust was easiest to detect when it covered larger areas or if the rust was depth

was increased. It is suggested that the technique would work best on pipes with a small wall thickness.

Although infrared technology plays a key part in the detection of CUI, it does have limitations. The technique only identifies water ingress on hot process lines and cannot give any insight into the condition of the structural integrity of the pipe. It does however allow for a timely spot measurement of a pipeline, meaning that large areas could be monitored quickly giving a better insight into problem areas in the plant. The fact that the technique detects water rather than corrosion is also positive as it could detect there is a problem before the corrosion process begins.

2.2.3 Ultrasonic

Ultrasonic is a widely used sensing medium for CUI due to its ability to inform the wall thickness of the pipe. It can be utilised in two forms. The first, pulse-echo, can be used to determine the wall thickness of the pipe at a single point. The issue with this is that insulation needs to be removed to allow access for the ultrasonic probe to make contact with the metallic surface. Some assets have been fitted with Insulation Inspection Ports (IIP) to reduce the amount of insulation removal and therefore reduce downtime overall. Although this is a solution to the repetitive removal of insulation layers on assets susceptible to corrosion, it is not a viable option for large areas of plant in existing operations. There is also the issue of ensuring that the IIPs do not contribute to water ingress if they are not correctly sealed or replaced after use.

The second ultrasonic technique, guide wave, allows the inspection of long lengths of pipeline from one measurement point. [30] This technique utilises a ring-array of transducers around the circumference of the pipe at one fixed position. The transducers propagate low-frequency ultrasonic signals along the length of the pipe. The signals are reflected off of defects, such as cracks or corrosion patches, enabling their detection due to the change from the incident signal. Commercially developed systems from Olympus (Ultrawave LRT) can detect defects up to a distance of 90m either side of the transducers. This enables significant lengths of pipe to be monitored without the insulation being removed except for where the array of transducers needs to be fitted. An example of this type of system is shown in Figure 2-7.

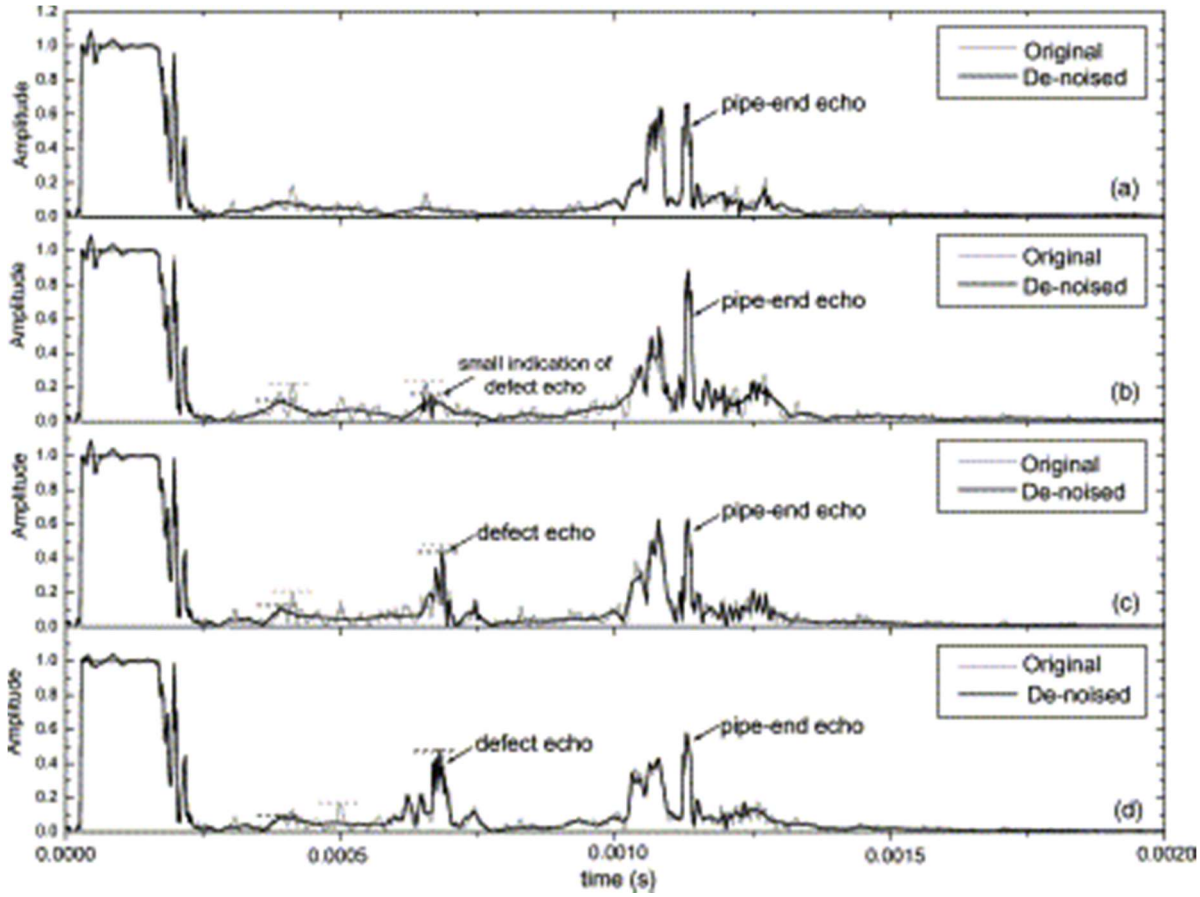


Figure 2-7 – Ultrasonic Pulse showing defect [31]

In summary, ultrasonic techniques have the advantage of being able to determine the wall thickness of the pipe. This allows for an assessment of the internal and external corrosion to be made. The guide wave approach specifically allows for long sections of pipe to be assessed in one measurement making this a quick assessment method due to the wave travelling in both directions from the point of contact.

The disadvantages when using ultrasonic techniques come from the fact that both techniques require the removal of the outer insulation layer. This presents the issue of having IIP's which potentially allow for water ingress, as well as the time and expense of removing and replacing the insulation. The technique is therefore not mobile and therefore can only perform assessment on areas of pipe within its capability. Ultrasonic techniques also require a couplant in order to transmit the signal from the device to the pipe with minimal losses. This in itself incurs a cost, but also a potential area for failure e.g. poor application.

2.2.4 Eddy Current

Pulsed eddy current inspection is another widely used technique to monitor the presence of CUI. This method is able to detect the wall thickness of the pipe with the insulation and cladding in place. Eddy current uses a probe coil wrapped around the outer cladding to induce a pulsed magnetic field. This magnetic field penetrates through any non-magnetic materials between the probe and the pipe, i.e. the insulation and cladding when non-metallic. The varying magnetic field induces eddy currents on the pipe surface. The behaviour of the eddy currents are relative to the pipe line materials as well as the pipe wall thickness in that area. The resultant eddy current signal is processed and compared with the reference signal to determine changes in the overall wall thickness in the immediate area of the coil. Figure 2-8 details a suggested setup for this technique.

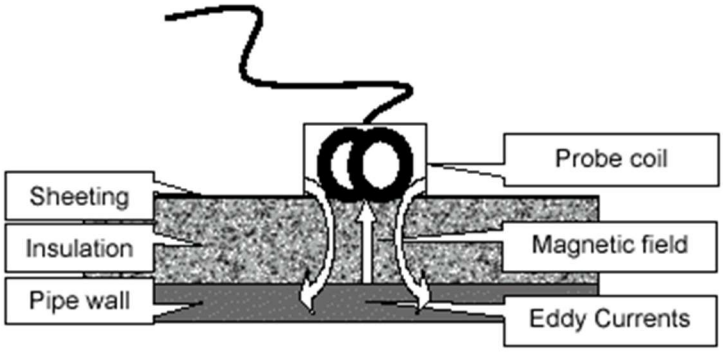


Figure 2-8 – Eddy Current Setup

The major advantage of this technique is it requires absolutely no insulation removal to monitor the pipe. This technique can also be carried out when the pipe is in service, so no loss of production when monitoring. The limited inspection area is a disadvantage along with the inability to detect through external cladding which is metallic. [32]

2.2.5 Neutron Backscattering

Neutron backscattering is an indirect approach to determine the presence of CUI. This technique detects the presence of water ingress similar to thermography as outline previously. Neutron backscattering uses high energy neutrons which are emitted from a radioactive source to inspect the insulation for the presence of water. If water is present in the insulation layer this is identified by the reduction in energy of the reflected neutrons, as water attenuates the energy of the neutron. The amount of water present in the insulation can be detected, unlike the thermography technique, as the amount of water present is proportional to the number of low energy neutrons.

The major advantage in using this technique is that the pipeline can remain in service and there is no need for insulation to be removed for the measurement.

The main disadvantage for using such a technique is personnel safety and training. Whenever a radioactive substance is required, this in itself becomes an issue due to its very nature. Additional training may be required for the staff member to use the technology safely, increasing costs, but limiting its usage to trained staff only.
The technique also only achieves a limited field of view into the pipe surface, meaning other techniques may be required for a better overview of the asset condition. [33]

2.2.6 Real-Time Radiography

Real-time imaging radiography has been developed extensively in recent year allowing for a real-time image of the asset to be displayed on site. This technology allows for images of the pipeline to be displayed meaning that the user/operator can assess the pipes in real-time and in-situ.
This technique utilises a low level X-ray source to image the outer pipe wall. The system is biased by a variable high voltage, typically 75KV, to allow adjustment of the image resolution. The electricity creates the radiation, so the device is considered safe when the power is off. [34]

The main advantage to using this technique is that the system gives a real-time view of the steel surface of the pipe, giving a clear and precise image of the defects or corrosion on the surface.

The disadvantage comes when using the device. A high voltage power supply is require, so this on its own introduces a logistical challenge. Secondly there is still the need to remove outer metallic cladding, meaning partial insulation removal which incurs time and cost penalties.

2.2.7 Microwave Sensing

A second method for detecting water ingress in a pipeline is a microwave technique. This technique uses the pipeline structure as a coaxial waveguide to propagate electromagnetic waves along stretches of the pipeline, shown in Figure 2-9. [35]

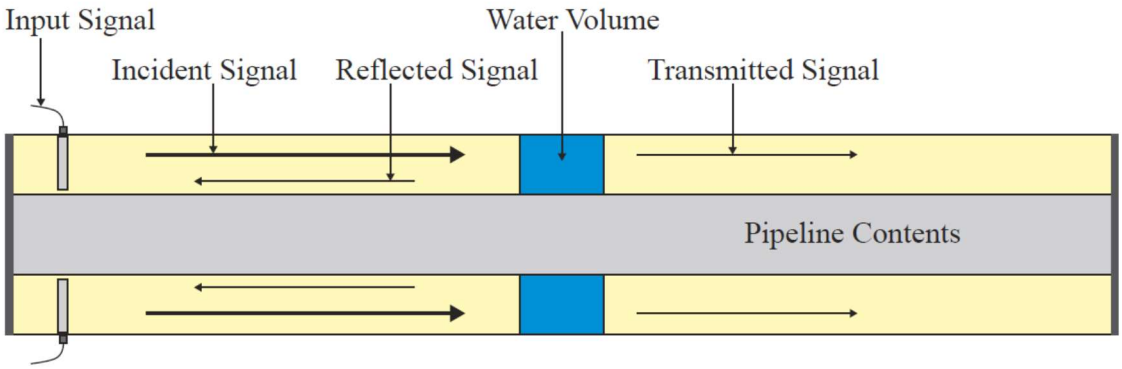


Figure 2-9 – Guided Wave Microwave Technique [35]

The microwave signal is attenuated by the wet pockets of insulation due to impedance mismatches in the waveguide (pipe structure). The impedance mismatches cause reflections of the incident microwave signal, allowing for the wet patches to be detected shown in Figure 2-10.

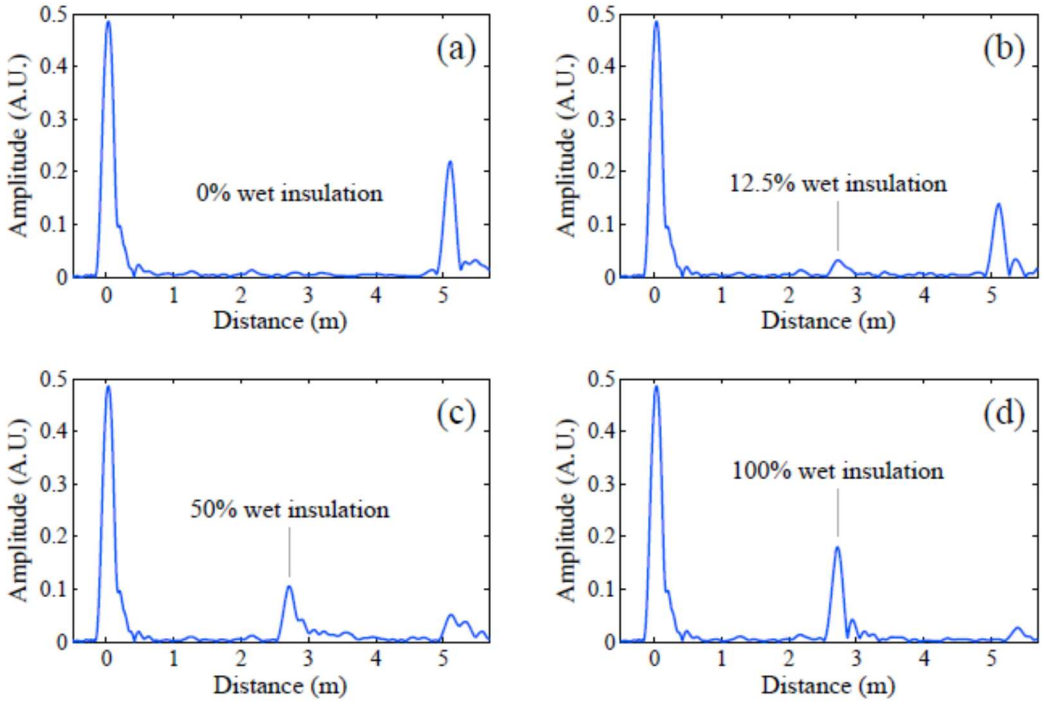


Figure 2-10 – Results Showing Wet Insulation [35]

As can be seen from the data, this technique is able to identify concentration of water in insulation based on the magnitude of the central peak. It can also be noted that as the reflection from the water increases, the reflection from the end of the waveguide decreases.

This technique has the advantage of being able to measure large sections of pipe at the same time up to 6m as tested in either direction from the antenna probes. It also has the ability to determine the concentration of water within the insulation which gives an idea of the scale of the issue.

The disadvantages of the system are that it can only work on pipes with external metal cladding to contain the waves. Although many historic assets do still operate with this type of cladding, many new assets have reverted to polymer and painted coating rendering the technique useless. Additionally when the technique does detect a patch of damp insulation, it will be unable to detect any further patches within the pipe section due to most of the energy being reflected from the first damp patch. Also the technique requires physical contact with the pipe outer surface which means removal of insulation and making the device fixed. This technique is also unable to give any indication of the health of the pipe itself.

2.3 Summary

As discussed in section 2.2 there are many techniques available to determine the presence of CUI, all with their advantages and disadvantages. Several of the techniques mentioned; thermography, neutron backscattering and guided wave microwaves, use the detection of water as the detection metric. Using this has the advantage of being an early warning precursor, but they do not inform the condition of the pipe.

Not all the techniques described give an assessment of the condition of the pipe. Visual inspection, Thermography and Guided wave can only give a precursor indication to signify the possible presence of corrosion.

Table 2-2 summarises the findings from the state of the art review and combines there with the points of the propose microwave sensor.

Technique	Couplant	Contact Pipe	Hazardous	Mobile	Instant Measurement	Pipe Condition	In Service	Handheld
Thermography	No	No	No	Yes	Yes	No	Yes	Yes
Ultrasonic	Yes	Yes	No	No	Yes	Yes	Yes	No
Eddy Current	No	No	No	Yes	Yes	Yes	Yes	No
Radiography	No	No	Yes	Yes	Yes	Yes	No	No
Visual Inspection	No	No	No	Yes	Yes	No	Yes	Yes
Guided Wave	No	Yes	No	No	Yes	No	Yes	No
Microwave Sensor	No	No	No	Yes	Yes	Yes	Yes	Yes

Table 2-2 – CUI Technique Summary

In the author's opinion a tool which requires no external stimulus or couplant and is truly handheld with an instant measurement is considered to be ideal CUI identification technique. The table reflects these points as the key metrics for comparison with other technologies.

As can be seen, the microwave sensor provides all the key advantages over the other proposed technologies. The microwave sensor is a very flexible technology and the deployment could be either as a handheld diagnostic device or as sensor array within a collar configuration to provide a 3D assessment of the asset, as illustrated in Figure 7-3.

In the following chapter, all relevant microwave theory will be discussed to provide an understanding of the concepts used in the design of the microwave sensor. Theory surrounding the critical microwave components used in the sensor design will be explained as well as principles of radar and more specifically Frequency Modulated Continuous Wave (FMCW).

Chapter 3- Theoretical Background

3.1 Maxwell's Equations

Maxwell equations are a set of four equations which define the behaviour of both magnetic and electric fields. [36, 37] Maxwell equations are named after James Clerk Maxwell who added the displacement current terms to the final equation, the steady state equations had been in existence longer. The equations below are listed in their differential form as follows:-

$$\nabla . \boldsymbol{D} = \boldsymbol{\rho_v} \tag{3-1}$$

$$\nabla . \boldsymbol{B} = \mathbf{0} \tag{3-2}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial \boldsymbol{t}} \tag{3-3}$$

$$\nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial \mathbf{t}} + \mathbf{J} \tag{3-4}$$

D is the electric flux density; $\boldsymbol{\rho_v}$ defines the electric charge density in Gauss' Law, **3-1**. In **3-2** (Gauss' Magnetism Law), magnetic flux density (**B**) is equal to **0** due to the nonexistence of monopoles (magnetic charges). Both these equations are used to calculate the divergence ($\nabla\cdot$) of the associated field either electric (**D**) or magnetic (**B**).

Faraday's Law (**3-3**) effectively states that a time varying magnetic field will always induce a spatially varying electric field. In real terms, when a current flows along a wire a magnetic field (B). The equation defines the curl ($\nabla\times$) of the electric field (**E**) by the partial differentiation of the magnetic flux density (**B**). **3-4**, known as Ampere's Law, defines the curl of the magnetic field (**H**) by partially differentiating the electric flux density (**D**) and adding **J** the current density. Ampere's Law dictates that an electric

current (**J**) gives rise to a magnetic field (**H**) when it flows. We can also see that an electric flux density (**D**) when varying with time can induce a magnetic field (**H**).

Essentially Maxwell's equation define the behaviour of all electromagnetic waves and declare how one field can induce another. To summarise, an electric flux density (**D**) varying in time will induce a magnetic field (**H**). Faraday's law also states that a varying magnetic field (**H**) induces an electric field (**E**) fundamentally declaring how electromagnetic waves propagate.$\boldsymbol{\nabla . D = \rho_v}$

3.2 Horn Antenna Theory

Horn antennas are simplistic structures which provide high gain and wide bandwidth for varying frequencies. There are three different types of horn antenna discussed in this thesis, the pyramidal horn and two sectoral horn antennas, E-plane horn and H-plane horn.[38]

The pyramidal horn antenna (Figure 3-1) is widely used in this thesis. Pyramidal horn antennas utilise a rectangular waveguide and are flared in both the direction of the electric field (E-Plane) and that of the magnetic field (H-Plane). [39]

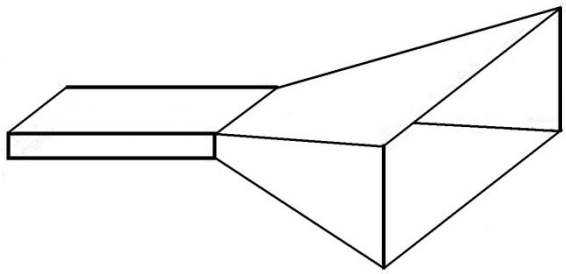
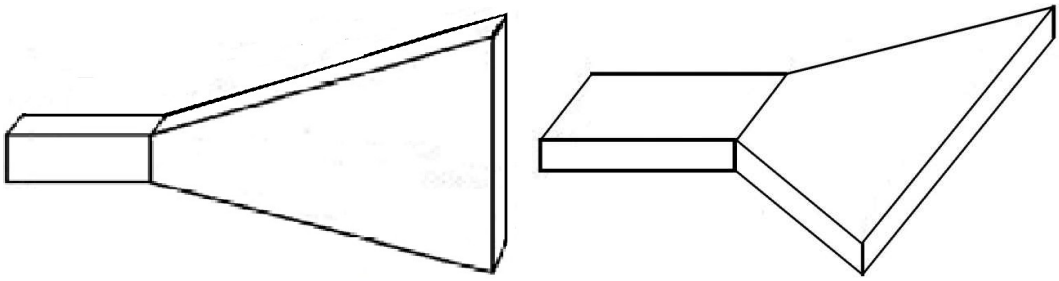


Figure 3-1 – Pyramidal Horn Antenna

This flaring in both directions ensures that the magnitudes of both the magnetic and electric fields are similar.

The sectoral horn antennas also utilise a rectangular waveguide but these horns are only flared in one direction, either the electric or magnetic field. An E-Plane horn antenna is flared in the electic field and narrow in the magnetic field, the H-Plane is the reverse of this.



E - Plane H - Plane

Figure 3-2 – Sectoral Horn Antenna (E-Plane and H-Plane)

Flaring the horn antenna in one plane has the effect of increasing the magnitude of the field in that plane. An E-plane horn has a larger electric field and the H-plane has a larger magnitude magnetic field.

Horn antennas typically have a directional radiation pattern with a high antenna gain. Gain of a horn antenna can tend towards 25 dB, but typically values in the region of 10 – 20 dB can be expected. The gain of the antenna increases along with the frequency of operation and the 'beam width' decreases. At high frequencies the horn antenna is electrically larger due to the high frequency having a smaller wavelength. The physical aperture of the horn antenna is therefore many time larger than the RF wavelength. [40]

The key parameters when looking at horn antenna design are radiation pattern, radiation power density, radiation intensity, beamwidth and directivity.

3.2.1 Radiation Pattern

A radiation pattern is generally a graphical representation of the radiation properties as a function of space coordinates. [40] Radiation patterns can be classed in 3 different ways. Firstly a pattern can be thought of as 'isotropic', this is when the radiation pattern is equal in all directions. [41] The next classification of radiation pattern is known as 'omnidirectional'. This type of radiation pattern is associated with slot antenna or dipole antenna and is defined as being isotropic in a single phase, an example of this is shown in Figure 3-3. [42]

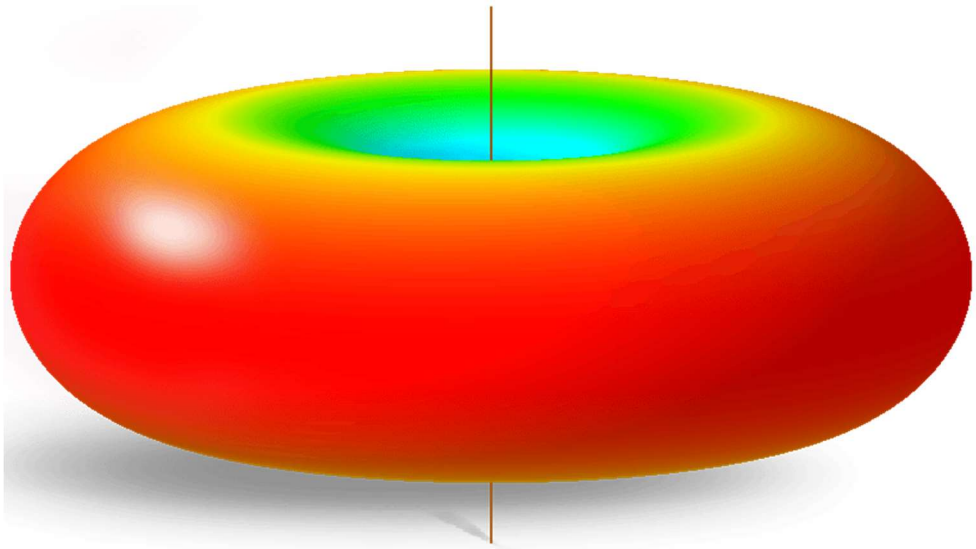


Figure 3-3 – Isotropic Radiation Pattern from Dipole Antenna [42]

Finally a directional radiation pattern, as used in this thesis, typically have a single peak in one direction where the majority of the radiated power travels. The radiation pattern of this type is shown in Figure 3-4, with a 2D representation discussed in the next section concerning key measurements to metric horn antenna performance.

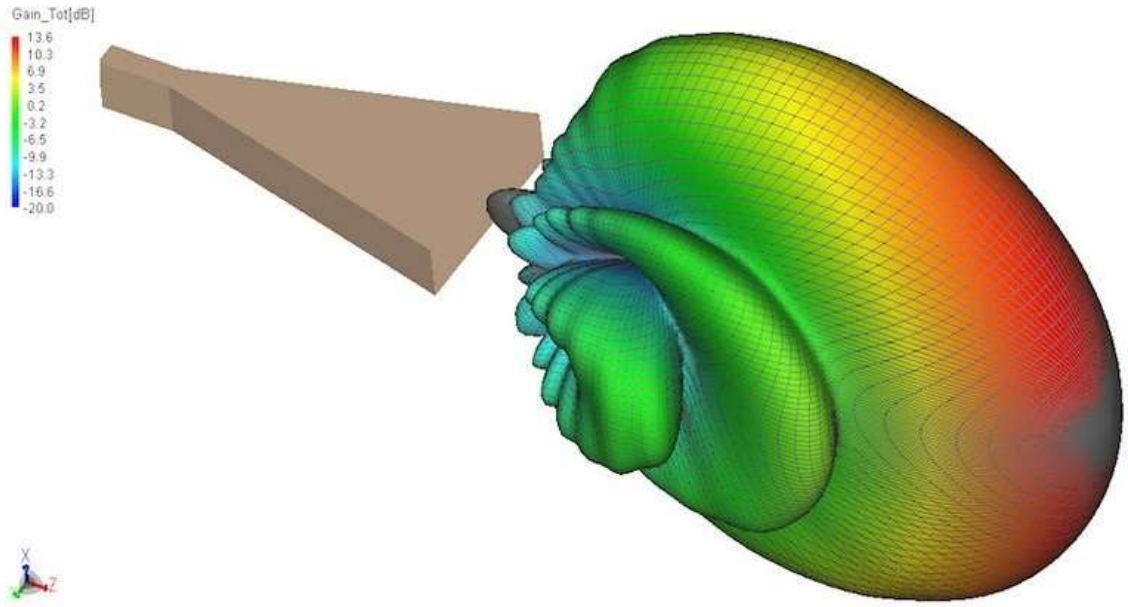


Figure 3-4 – Horn Antenna Radiation Pattern [43]

3.2.2 Beamwidth

Beamwidth is a parameter which is determined from the radiation pattern of the horn antenna. Most commonly, beamwidth is defined as the Half-Power Beamwidth (HPBW). The HPBW is calculated as the -3 dB (half power) points taken from the maximum radiated power on the main lobe as shown in Figure 3-5.

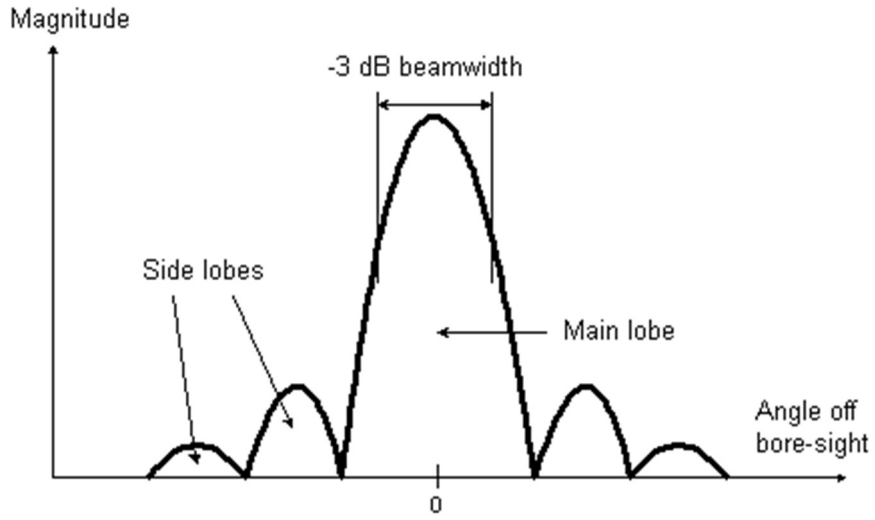


Figure 3-5 – Horn Antenna Beamwidth [44]

The beamwidth is typically measure in degrees and is a quoted parameter for all off the shelf horn antennas. This parameter relates to the side lobes of the horn antenna, in that, when the beamwidth is decreased the side lobes are increased. Beamwidth is also an important parameter when capturing the resolution capability of the horn, i.e. ability to distinguish between two targets.

3.2.3 Directivity

The directivity of a horn antenna defines the ratio of the radiation in a specific direction in comparison to the intensity averaged in all directions. In general the direction of most radiation intensity is presumed to be the proposed direction. The directivity of the horn can be calculated as follows:

$$D = \frac{4\pi U}{P_{rad}} \qquad \textbf{3-5}$$

D is the directivity of the horn antenna, **U** defines the radiation intensity and $\mathbf{P_{rad}}$ is the total radiated power. [45] An antenna with a higher directivity suggests more focussed antenna, which is what is required for small defect detection in terms of this thesis. To measure larger areas in one measurement it would be useful to have a horn with a lower directivity, although a lower resolution larger defects could be detected.

3.3 Waveguide Theory

Typically, horn antennas are fed by waveguides. A waveguide is quite simply a hollow metal cavity which is usually fed using a dipole, i.e. SMA Connector. As the name suggests, waveguides are used to direct electromagnetic waves from one place to another. Most commonly, they are fitted to horn antenna structures to direct the electromagnetic wave through the antenna and ensuring that any reflected energy is again captured and focussed back at the dipole.

Waveguides are essentially coaxial lines without a centre conductor. They are constructed from conductive materials which can be fashioned into a variety of shapes, most commonly and in the case of this thesis, rectangular.

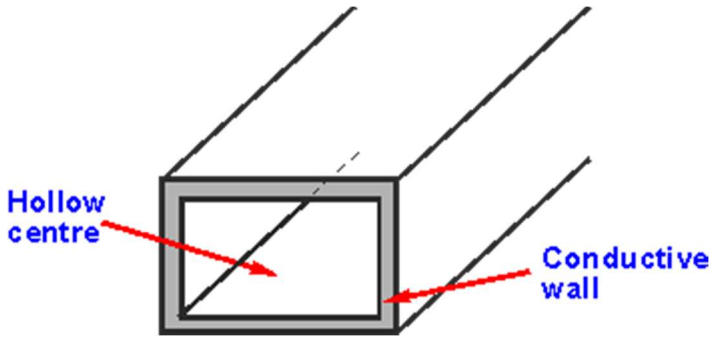


Figure 3-6 – Rectangular Waveguide [46]

There are many advantages for using waveguides over conventional two wire transmission or coaxial wires, the first being the reduction in copper losses. Due to the large surface area of the waveguide the copper losses are vastly reduced in comparison to a two wire transmission set up. This is due to the very small surface area of the wire; a coaxial line offers a larger surface area on the outer conductor, but the inner centre conductor has a very small surface area and at microwave frequencies the current carrying area of the inner conductor is restricted to a very small layer at the surface due to the skin effect.

Dielectric losses in waveguide are also lower in waveguides in comparison with the other two methods outlined. These losses are caused by the heating of the insulation between conductor lines due to the voltage potential between them. This heating effect results in power loss and therefore lowers efficiency. In a waveguide the dielectric material is air. Air has a much lower dielectric constant than the materials commonly used as insulation. [47]

There are a number of different types of electromagnetic waves which can propagate within a waveguide. The first type of wave is known as Transverse Electric Waves (**TE**). TE waves are determined by the fact that the electric vector is perpendicular to the direction of propagation within the waveguide. Transverse magnetic Waves, **TM**, are determined so due to the magnetic vector being perpendicular to the direction of propagation. Lastly **TEM** waves allow both magnetic and electric vectors to be perpendicular to the direction of propagation. This mode however cannot operate within a waveguide, only in a coaxial feed. [48]

Waveguide can only propagate signals above a certain frequency and this is known as the cut-off frequency, $\mathbf{f_c}$. The cut-off frequency is essentially the frequency at which the waveguide will not operate. To calculate the cut-off frequency for a rectangular waveguide equation **3-6** can be used. It should be noted that as stated above there are many transmission modes which operate in waveguide, but TE_{10} is the assumption used for the equation below.

$$\boldsymbol{f_c = \frac{c}{2a}} \qquad \textbf{3-6}$$

From this equation, **c** is the seed of light and **a** represents the largest internal dimension of the waveguide. [48]

3.4 Radar Basic Principles

The fundamental basic electronic principle on which radar operates is similar to the basic principles of sound-wave reflection. Instead of using sound-waves, radar transmits electromagnetic pulses. These pulses of energy are transmitted and then reflected off of the target being monitored. The reflected energy captured by the radar is known as the echo in the same way an audio wave would be. The echo can be used to determine both the distance and direction of the object. [49]

All radar basic radar systems follow the same transmission path principle as shown below:

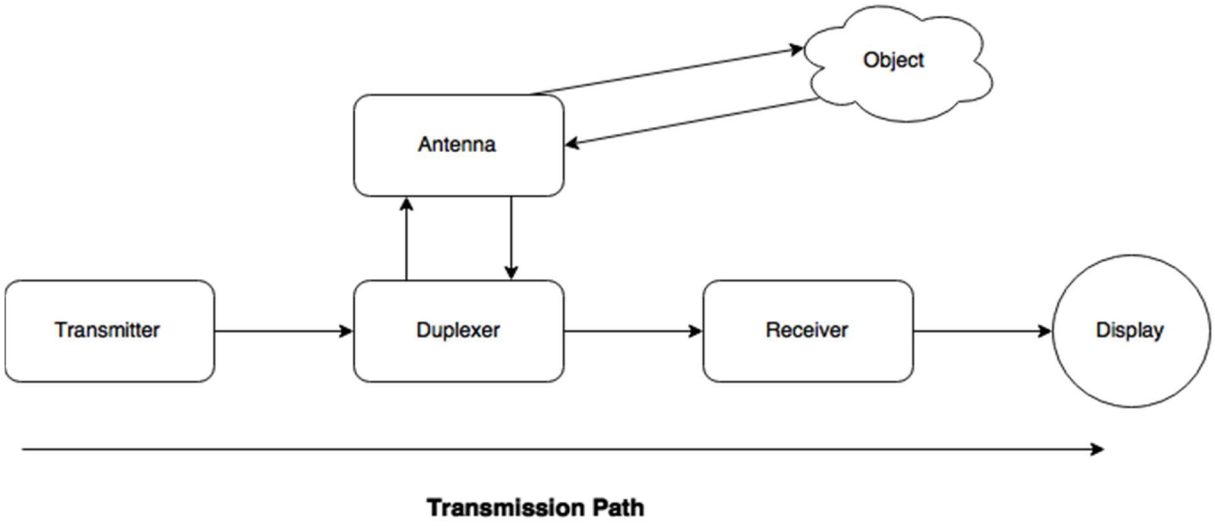


Figure 3-7 – Radar Transmission Path Block Diagram

The functions of each of the components are as follows. **Transmitter** generates the pulse of RF energy sent into 'space' by the antenna. The **Duplexer** acts as a switch to alternate the antenna between transmit and receive mode. This ensures only one antenna is required. The returned signal is amplified and demodulated by the **Receiver.** [50]

The transmitted signal is passed through the duplexer and transmitted by the antenna. The signal will be returned as a reflection off of any object there is contact with. The returned signal is focussed by the antenna and then passed through the duplexer and sent to the receiver before being displayed on the display module. [51]

3.5 FMCW Theory

Frequency modulated continuous wave radar (FMCW) differs from conventional radar systems as it continuously transmits an electromagnetic signal. The transmitted signal is varied over time, by sweeping the frequency linearly over a predetermined bandwidth. The reflected signal is mixed with the transmitted signal to calculate the frequency difference due to delay caused from the time of flight. Due to the nature of FMCW the resultant signal is in the low frequency domain as both high frequency components have been subtracted from one another, the resulting frequency is known as the intermediate frequency (IF). [52]

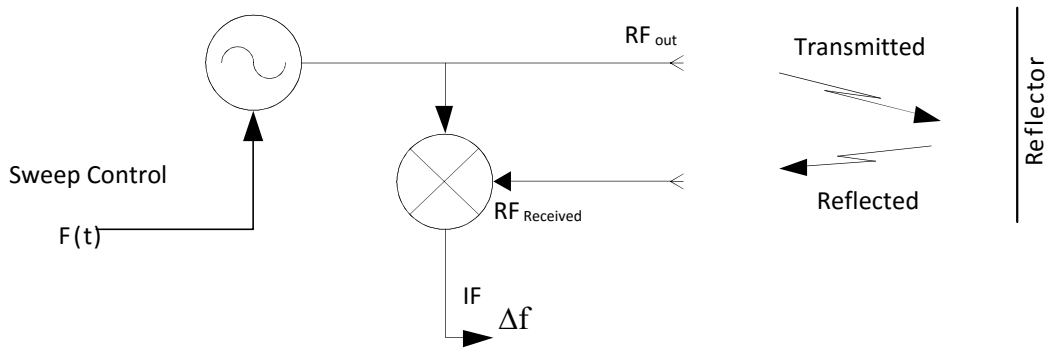


Figure 3-8 – FMCW Flow Diagram

Figure 3-8 shows the signal flow through the whole FMCW radar setup. As can be seen, the output signal from the radar unit is declared as the Intermediate Frequency (IF) which can be derived as:

$$\Delta f = \frac{BW}{T} . 2\frac{d}{c} \qquad \textbf{3-7}$$

From equation **3-7**, $\Delta \boldsymbol{f}$ defines the Intermediate frequency, the bandwidth of the frequency range is represented by **BW**, **T** is the amount of time required to sweep the bandwidth, **d** is the distance to the target or reflection point and **c** is the speed of light constant.

FMCW radars use different frequency modulation waveforms such as saw-tooth, triangular, and square wave for different measurement purposes. Saw-tooth modulation is used to measure objects at large distance from the detector. Triangular modulation is widely used as it enables easy separation of the difference frequency and the Doppler frequency (caused by change in speed). This modulation type can be used at a wide range of distanced from the detector. Square wave modulation is used for close range measurements with high precision. This requires more signal processing and proves difficult to separate out echoes form multiple objects.

3.5.1 FMCW Theory Specific to Hardware

The hardware used in this thesis used a specific type of modulation to determine targets. Unlike traditional FMCW, synthesised modules, as used throughout the work in this thesis, do not sweep the frequency continuously. The frequency is stepped using a set of discrete frequency points.

The sensor output corresponds to the cosine of the phase difference between the reflected signal and the transmitted signal. In real terms, the signal generated is a cosine signal representing the electrical distance which the radiated signal has travelled. Thus,

$$s = \cos(2\pi \frac{2d}{\lambda})$$

3-8

Where **d** represent the distance to the reflector and **λ** the electrical wavelength of the RF signal. The multiplication by 2 is to take account of the total trip of the signal. [53]

The phase difference (**φ**) between the transmitted and reflected signal can therefore be written as,

$$\varphi = 2\pi \frac{2d}{c} f_{RF}$$

3-9

Where **c** is the speed of light and $\mathbf{f_{RF}}$ is the frequency of the signal.

From equation **3-9** it is reasonable to say that φ will increase linearly with f_{RF} thus the detector signal will produce a cosine signal. A small value of **d** (close object) will produce a slowly varying signal, where a large value of **d** (distant object) will produce a fast varying signal. As this thesis has used close objects the signal has represented a slow changing cosine which can be seen in the various graphs of the results section 5.3. [53]

3.6 Microwave Sensing Theory

This section aims to outline how microwave sensing works by explaining what manipulates the electromagnetic wave to cause a change in response. There are two key parameters which cause the electromagnetic wave to respond to the materials being sensed.

Firstly, permittivity (**ε**) effects the propagation of electromagnetic waves. Permittivity specifically effects the propagation of electric field within a material. As shown in Figure 3-9, when a positive and negatively charged plate are placed opposite one another with an equal charge at a set distance, **d**, an electric field is generated between

them. The electric field propagates from the positive to the negative plate. There is currently no material between the plates, i.e. a vacuum.

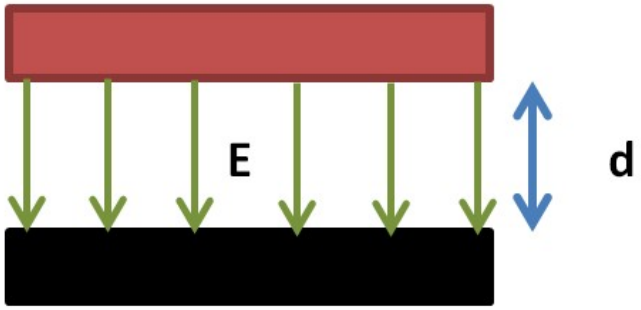


Figure 3-9 – Electric Field between two plates

Materials are made up of various atoms and molecules which tend to contain some kind of dipole. A diploe is a molecule with two atoms with opposite polarity, i.e. one negative and one positive. As shown in Figure 3-10, these dipoles are randomly orientated within the material when no electric field is present.

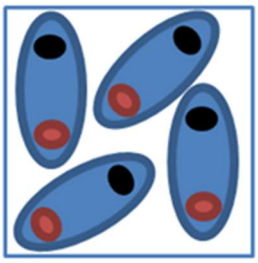


Figure 3-10 – Material Electric Dipoles with no Electric Field

When the material is placed between the plates with the propagating electric field, the dipoles of the material align with the corresponding oppositely charged plate as shown in Figure 3-11.

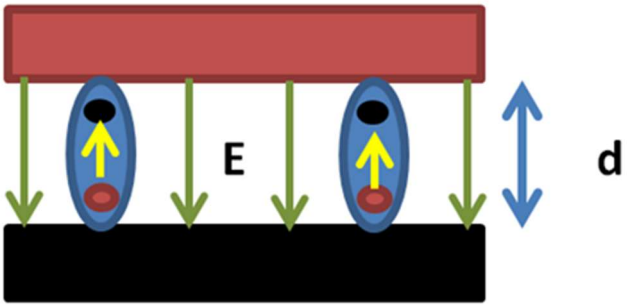


Figure 3-11 – Molecules aligned with Electric Field

The image shows that the electric field due to the dipole molecules opposing the externally propagating electric field. This results in the overall net effect of the external electric field being reduced, therefore any material placed within an electric field will ultimately reduce net effect, altering the electromagnetic wave propagation.

In summary, permittivity quantifies how much the molecules of the measured material oppose the electric field. In relation to Maxwell's equations:

$$\boldsymbol{D = \varepsilon E}$$

3-10

D the electric flux density is directly proportional to the electric field **E**, which is scaled by the permittivity **ε**.

The second parameter which effects the propagation of electromagnetic waves is permeability (**μ**). Permeability relates the magnetic flux density **B** to the magnetic field **H.**

$$\boldsymbol{B = \mu H}$$

3-11

As shown in **3-11**, the permeability is the multiplier which directly relates the magnetic field with the magnetic flux density.

Similarly to permittivity, permeability is associated with the orientation of bound 'magnetic' particles in a material. Therefore the permeability is in essence a measure of how well a magnetic field can pass through a material. The permeability of a material is most often quoted as the relative permeability, $\boldsymbol{\mu_r}$. The relative permeability is the permeability relative to that of a vacuum:

$$\mu_r = \frac{\mu}{\mu_0}$$

3-12

The relative permeability of a material can be less than 1 which is known as Diamagnetism. This is where a material creates an internal magnetic field which is opposite to the externally applied field. Therefore a diamagnetic material would therefore reduce the net effect of the magnetic field applied by the microwave sensor.

Material with a relative permeability greater the 1 are known as paramagnetic materials. These materials will create a small internal magnetic field equal to that of the external magnetic field. Paramagnetic materials present the same behaviour defined by equation **3-11**.

Finally, materials with a large relative permeability are known as Ferromagnetic materials. These materials are typically used as magnets due to their high magnetic attraction. These materials can retain the magnetic properties of the external magnetic field once it has been removed.

Table 2-1, details various elements and their typical relative permeability as well as their classification as examples of each type of magnetic material.

Element	Relative Permeability	Classification
Copper	0.9999906	Diamagnetic
Silver	0.9999736	Diamagnetic
Lead	0.9999831	Diamagnetic
Magnesium	1.001	Paramagnetic
Air	1.00000037	Paramagnetic
Aluminium	1.000021	Paramagnetic
Iron	280000	Ferromagnetic
Nickel	600	Ferromagnetic
Cobalt	250	Ferromagnetic

Table 3-1 – Magnetic Material Properties [54]

In summary, the permeability and permittivity are the two main properties which effect the propagation of electromagnetic waves and are therefore the two properties which are of interest when analysing results from the microwave sensor.

Chapter 4– Sensor and System Design

This chapter summarises the sensor design as well as the overall system design process. In following sections the design process of the overall system is described including the hardware setup, listing the components used to make-up the system as a whole and their relevant specifications. Following this, the next section will describe the system software for both control and user interface. Finally the sensor design is described including the reasons for the choice of horn and waveguide as the sensor.

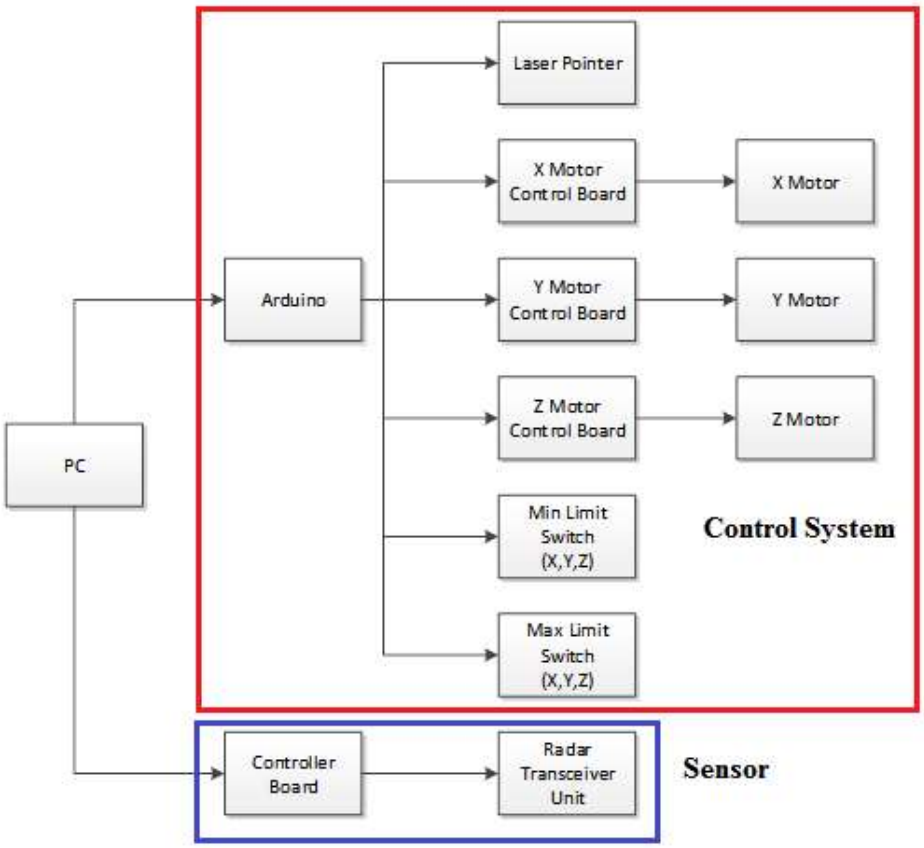


Figure 4-1 – System Block Diagram

The system designed within this thesis was for the initial lab trials of the proposed sensor. A system which would be used in a real life situation would not require a translation stage as this would be a limiting factor in terms of mobility of the device. The current system was designed with the primary focus of achieving repeatable results for the proof of concept.

4.1 System Hardware

The complete control system for the sensor is outlined in this section. The section will focus on the hardware and software used to create a stable and reliable platform for the sensor measurements.

Figure 4-1 shows the system block diagram as a whole, listing all of the components required to operate the system. The block diagram is split into two separate sub systems. The first is the control system which can be defined as the translation stage and the components required to operate it. This includes all the physical hardware components as well as the microcontroller used to control and drive the individual parts. The second is the sensor section which includes only the electronic modules of the sensor in terms of electronic hardware which is outlined in section 4.1.3. The complete sensor includes a horn antenna which is explained in section 4.4

4.1.1 Translation Stage

Ensuring that the data from the sensor was accurate and dependable was a key part of the design process. The hardware used to drive the sensor is fundamentally a radar system, so ensuring the distance between the sensor head and asset was critical. A stable and controllable platform was required to guarantee the exact positioning of the sensor over an asset as well as a smooth and stable operation to reduce interference with the measurements.

To achieve this a Reprap 3D printer kit was used. This kit was already available within the university and could be reutilised with minimal cost. The setup included many of the key components and attributes needed to support the sensor. The platform area allowed for samples up to 200 mm x 200 mm. The height of the platform could be adjusted through 140mm. This provided an ideal sample base and fulfilled the needs of most of the experiments conducted; the stage is shown in Figure 4-2

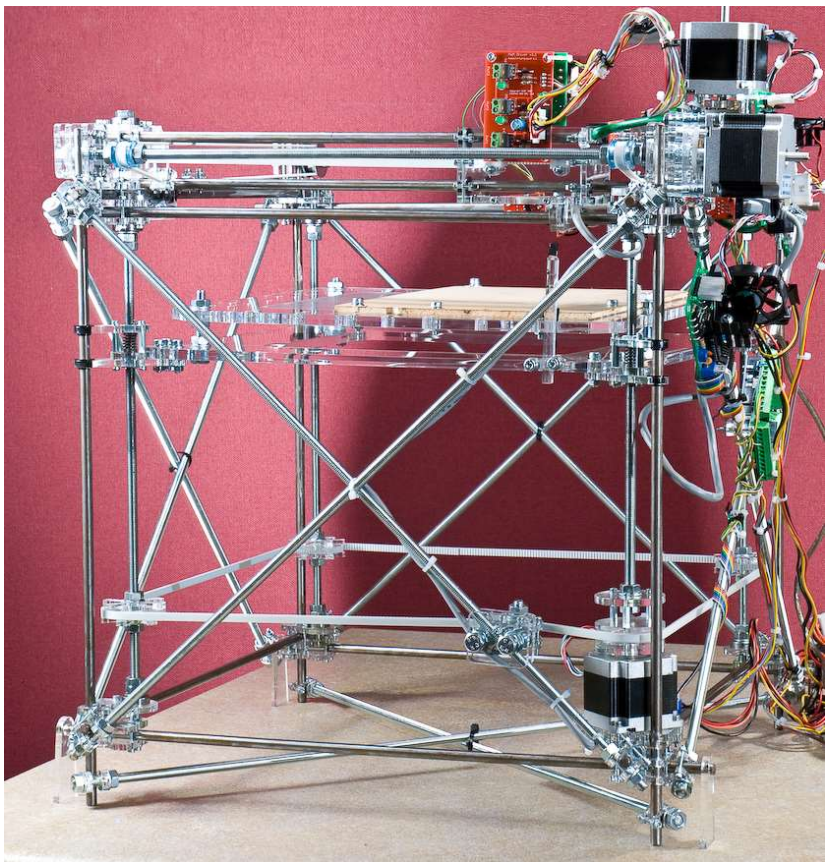


Figure 4-2 – Translation Stage

The kit also included 3 servo motors. The motors were NEMA 23 high torque steppers. The motors were well suited for the application due to the high torque and are were recommended for CNC applications due to the accuracy and controllability. The motor has 400 steps to one rotation realising 0.9º step. This combined with the drive belts allowed for movements of just 0.1 mm per axis. The nature of a servo motors ensured that movements to the stage could only be made electronically, minimising any chance of movement from accidental external forces.

The driver boards for the stepper motors translate the signals from the microcontroller to stepping signals to drive the motor. The driver board utilises electronics to aid in supplying the 2A per coil required to physically move the motor.

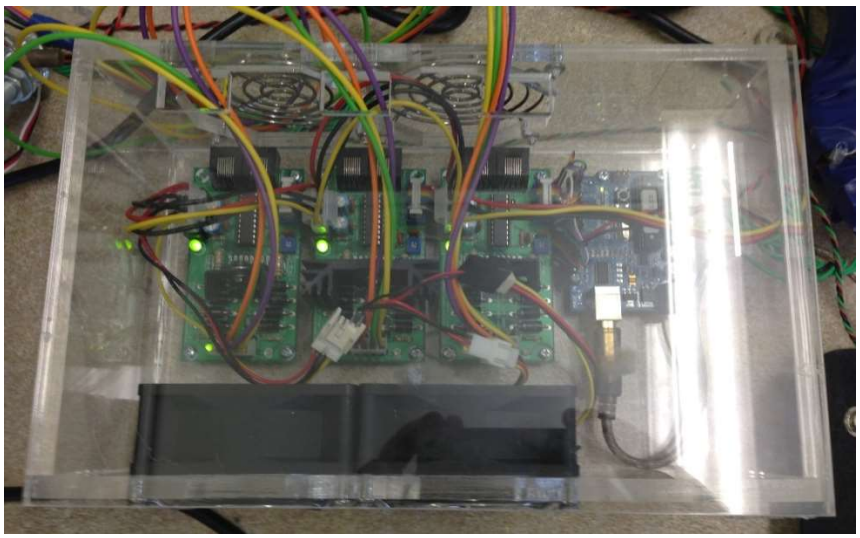


Figure 4-3 – Translation Stage Electronics

All the electronics components for the control of the translation stage were housed in a custom box with PC cooling fans to ensure that the motor control boards did not overheat during operation. The box was constructed from clear acrylic sheets which could be easily manufacture on the departmental laser cutter.

The translation stage had optical limit switches which proved to be unreliable. These were replaced with standard three pin micro switches which were more simplistic, reducing the risk of failure.

The switches were mounted so that the moving section of the translation stage would mechanically operate the switch when the maximum or minimum distance was reached on any of the axis.

4.1.2 Arduino Microcontroller

All of the electronic components were interfaced through the Arduino Duemilanove microcontroller. The Arduino microcontroller has built in USB port which allows for easy interfacing with a PC for control purposes. It also aids the upload of the firmware onto the chip. Arduino also has a native software program to write the firmware using the native language compiled with a C/C^{++} compiler.

The Arduino microcontroller also had the relevant connectivity for all the electronic components. The I/O pins are female 2.5mm headers allowing for easy connectivity with single core wires.

All electronic components were interface using the digital I/O pins and were assigned the following:

Component	Pin Assignment	Component	Pin Assignment
Steppers		**Limit Switches**	
X Step	3	X Min	A5
X Direction	2	X Max	A4
Y Step	10	Y Min	A1
Y Direction	9	Y Max	A0
Z Step	7	Z Min	A2
Z Direction	6	Z Max	A3

Table 4-1 - Arduino Pin Assignments

A full code listing of the Arduino firmware is given in Appendix A-E.

4.1.3 Sensor Electronics

The sensor electronics consisted of two off the shelf components sourced from Sivers IMA, an industry leader in millimetre wave products. The first element of the sensor was the FMCW transceiver module shown in Figure 4-4.

Figure 4-4 – RS3400K/00 Transceiver Module [55]

The transceiver unit is a synthesized FMCW radar front end for K-band operation. It operates across a maximum bandwidth of 1500 MHz from 24 GHz to 25.5 GHz. The frequency of the unit is controlled by digitally programming the embedded synthesiser

circuit via the 3-wire serial interface or through RS-232 when used in conjunction with the controller board as in this case.

The controller board shown in Figure 4-5 provide a simple interface between the PC and radar module via RS-232.
Figure 4-1 shows that the sensor hardware is interfaced directly with the PC rather than through the Arduino control board. The sensor hardware could in theory be interfaced with the Arduino board, but the I/O limit was already reached due to the other hardware outlined in section 4.1.1. The control board has a 3 wire Serial Peripheral Interface (SPI) bus alongside the standard RS-232. The Arduino board supports the use of SPI and has a built in library to support the interface between peripheral devices. [56] The board suppliers also provide a document to support the code development for full communication for both control and data retrieval. [53]

The board has an inbuilt microcontroller to control the FMCW frequency sweep and a 16 bit ADC to measure the Intermediate Frequency (IF). The measurements are stored on the board until the PC requests the data to be transmitted.

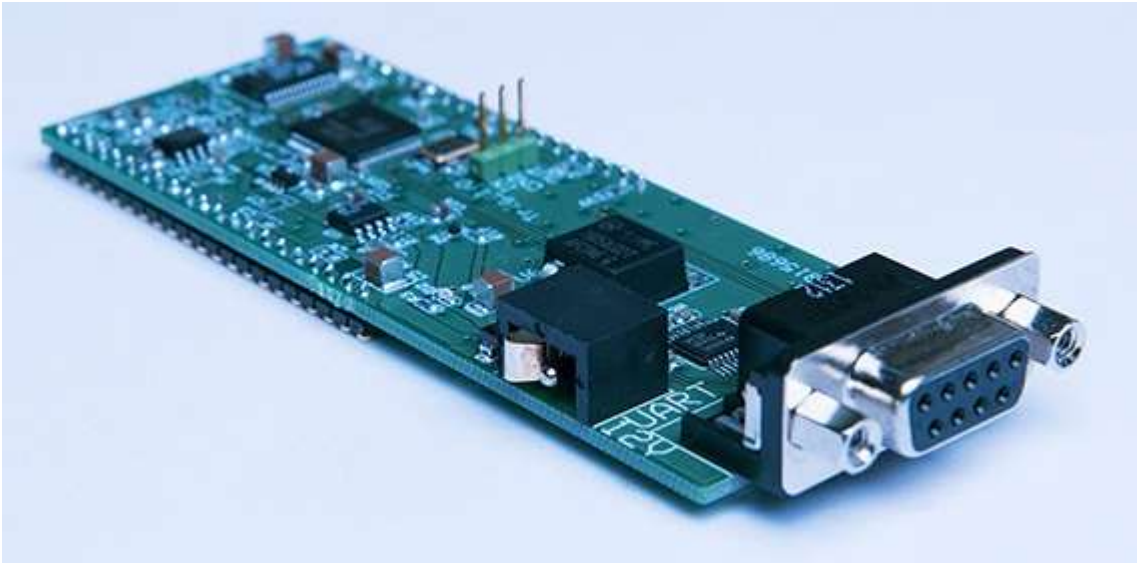


Figure 4-5 – CO1000A/00 Controller Board [57]

Both of these components have been designed to be intrinsically safe due to their low power consumption and low thermal emissions. This ensures they can be used in certain hazardous environments without risk of causing an ignition. The hardware has been graded as military standard, meaning it can withstand significant shock levels without causing any damage to the electronics.

Although these are not set requirements for the proposed sensor system outlined throughout this work, they are desirable attributes for future working prototypes if required to enter hazardous environments for testing.

4.2 Overall System Software

As discussed in the previous section, the firmware to allow the control of the hardware components was written and developed in the Arduino software suite. The next step in the design was to create a Graphical User Interface (GUI) to allow all the elements of the hardware to be controlled via the PC. The GUI had to be able to request data from the sensor and display it graphically for the user. The data would require processing and storage options for future analysis. The following subsections will describe the software and outline the full user interface.

4.2.1 Data Acquisition

In order to acquire the data from the sensor, a series of predetermined serial commands from the manufacturer had to be sent to the control board. The commands were transmitted over the RS-232 interface included on the board and the data would be received from the board after each measurement command sequence. Upon starting the system, the board would send an initial initialisation message which confirmed the communications between the control board and FMCW module. This message confirmed the FMCW module connected and programmed the control board to the correct frequency band. Failure to detect the correct operation band would prevent the sensor from operating.

The command sequence required to retrieve data from the sensor was as follows:

INIT
SWEEP:MEASURE ON
SWEEP:NUMBERS 1
TRIG:ARM
TRACE:DATA ?

Anytime a question mark could be used after any command to query the control board and request for the value of that parameter to be read back to the PC, it also was the trigger to request the measurement data to be returned from the device.

The ‘SWEEP:NUMBERS’ command acted as an averaging command. When the number was set higher than one, the board would request the module to perform the set number of sweeps across the whole frequency band. The resultant IF frequency from each of the sweeps is stored on the control board and when the data is called back to the PC only the average of all the sweeps is given. This proved to be an extremely useful command to increase the repeatability of measurements as any external noise was simply averaged out.

4.2.2 Graphical User Interface

As mentioned, the GUI was implemented using the Matlab software suite as it allowed for data capture, manipulation and presentation, as well as full control of all the hardware. The Matlab GUI design tool is known as ‘GUIDE,’ Graphical User Interface Design Environment. GUIDE is an interactive layout editor which automatically generates code for each element added to the screen window. [58] Features such as buttons, sliders, images, etc. can be added to the display. Figure 4-6 shows the main window of the GUI for the system.

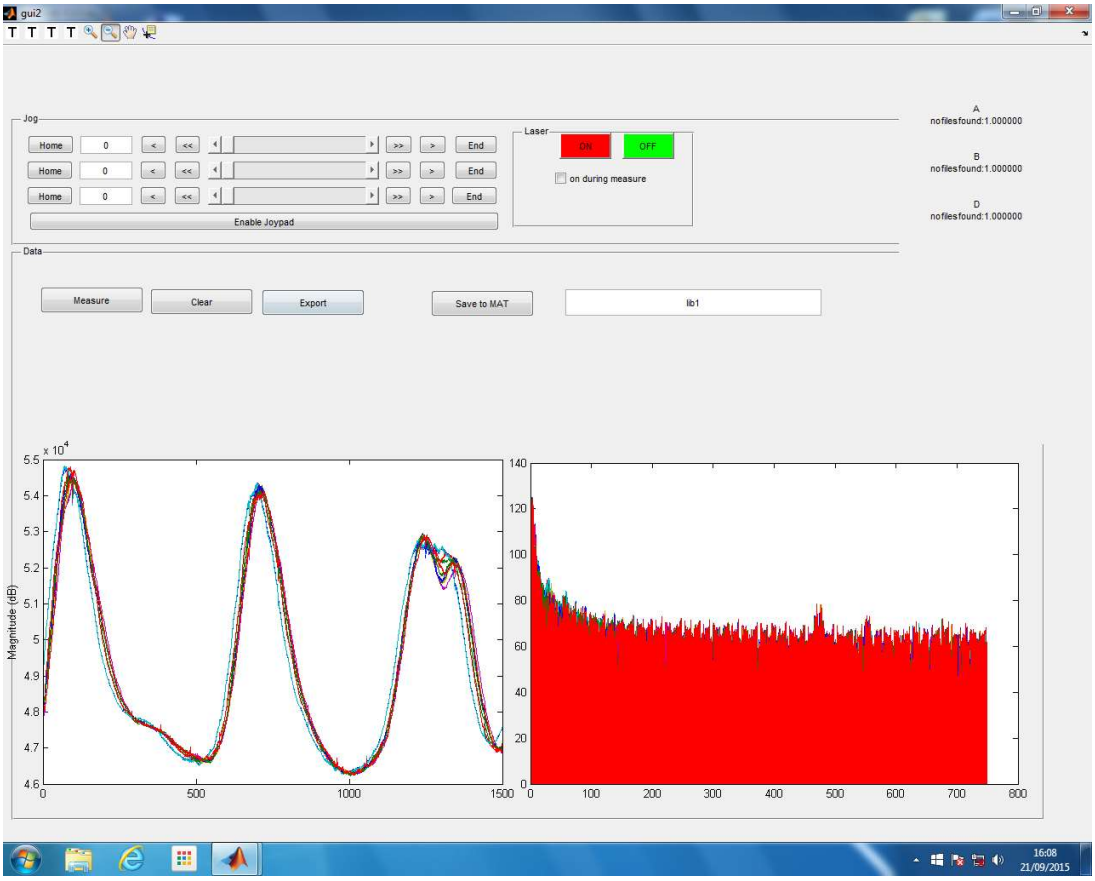


Figure 4-6 – Measurement Tab Graphical User Interface

As shown, the main GUI window provided much of the functionality. The main area of the window was for the presentation of the data once captured. The left hand graph plotted the raw sensor data every time the measurement button was pressed. On the right hand graph, the Fast Fourier Transform (FFT) data was presented. This proved to be the major benefit of using the Matlab interface as the FFT function within the Matlab software could be called into the GUIDE and used to perform the real-time data analysis. The top of the window contained the controls for the movement of all three axis of the translation stage. The sliders allowed for large movements of the motors, while the trim buttons at either side of the sliders allowed for either 1 step movements or 100 step movements from one press. The laser enable was also included on this page so it could be used to align the sensor over a specific area.

Figure 4-7, was a second tab from the main window. This included both the initial communications selection as well as the more complex automation of the sensor. The left hand side of the figure contains the communication selection. A dropdown box contained all the available communications port on the PC. The correct port for the sensor and the translation stage could be selected from the list and then by clicking the appropriate button assigning that as the correct port for the hardware.

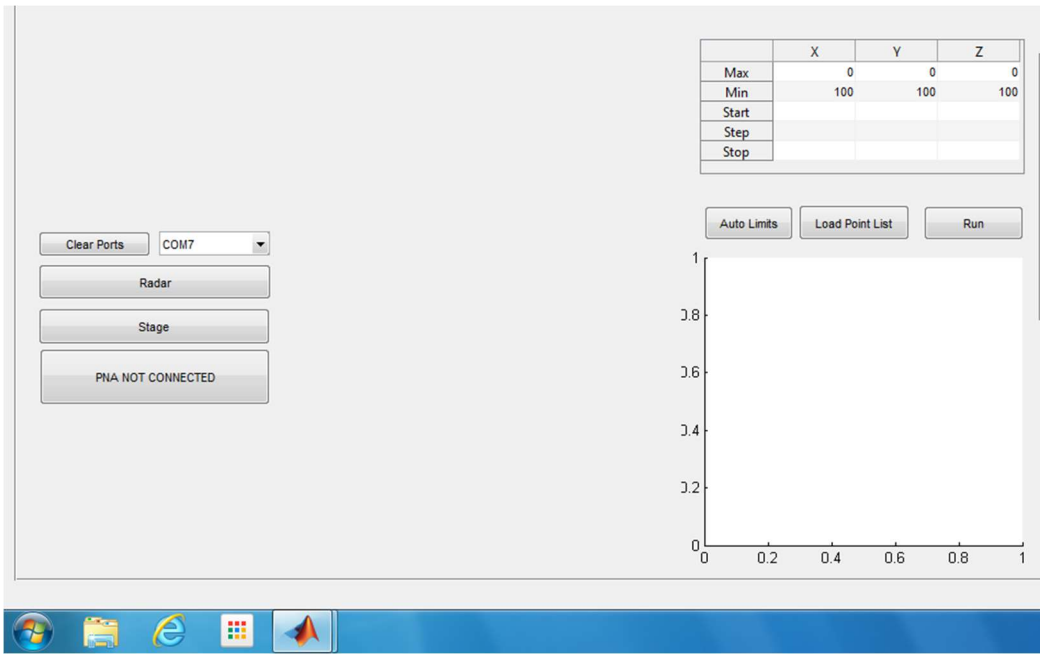


Figure 4-7 – Com Tab and Auto Tab Graphical User Interface

The automation of the sensor was carried out by the controls on the right side of the figure. To perform a single measurement over a variety of points across the

measurement area, the start and top point of each axis could be defined. The distance between each measurement was also required and the plot of the proposed measurements was displayed below. This proved to be useful when running scans through one axis to take a measurement across a set distance of a sample.

The GUI software was written over a period on few months, with additional functionality added when required. Initially the GUI only provided data capture and basic control of the stage. The complete code listing for the fully developed GUI can be seen in Appendix F.

4.3 Signal Processing

Several functions were developed and written in the Matlab software to aid with data capture or to provide feature extraction. The main functions are outlined below in separate subsections with the key code.

4.3.1 Peak Function

The function 'peak' determines the presence of peaks in the FFT data based on the rate of change between points in the x-axis. The function has two threshold values defined by the user to tune the function to locate peaks. The code is as follows:

```
function [out] = peak(x,y)
%Peak function determines peaks in data set based on rate of change
%between points
%    If roc between point A and B is greater than +/- T then a a peak
is
%   determined.

out = [];
T=850;
thresh1 = T;
thresh2 = -T;

samples = 10

for i = 2:length(y)-1
    roc1 = y(i) - y(i-1);
    roc2 = y(i+1) - y(i);
    if roc1 > thresh1 && roc2 < thresh2
        out = [out; x(i), y(i)];
    end
end

%Change value for the number of outputted peaks
if(size(out, 1) > samples)
    out = out(1:samples,:);
end
```

As can be seen from the code listing, the threshold values ‘thresh1’ and ‘thresh2,’ are determined by the user value ‘T.’ T determines the rate of change between one point and the next. The function uses the data from the FFT and determines a peak when the proceeding value has a value of a greater magnitude than ‘T’.

In the case shown, the function would note a peak where any proceeding value had a magnitude change of 850 or more. The function was self scalling as defined by the ‘out’ parameter. This ensured that the the function could be utilised on any size of dataset without change.

4.3.2 Correlate Function

The ‘correlate’ function was implemented to run within the GUI software for the overall sensor system. The main purpose was to compare the live data with data stored in a user defined library. The function allowed for a real-time comparison of data with historic data stored locally.

```
function [ output_list ] = correate(directory, samplea, sampleb,
sampled)

directory = strcat(directory, '\**\*.mat');
list = rdir(directory);
%list = list.name;

output_list = struct();

for loop=1:numel(list)
    data = load(char(list(loop).name));
   % fft = abs(data.b);
    data.b(:,end+1) = sampleb;
    [br,bp] = corrcoef(abs(data.b));  % Compute sample correlation and
p-values.
    br = br(end, 1:end-1);
    bp = bp(end, 1:end-1);

  %  raw = abs(data.b);
    data.a(:,end+1) = samplea;
    [ar,ap] = corrcoef(data.a);   % Compute sample correlation and p-
values.

    ar = ar(end, 1:end-1);
    ap = ap(end, 1:end-1);

    data.d(:,end+1) = sampled;
    [dr,dp] = corrcoef(data.d);
    dr = dr(end, 1:end-1);
    dp = dp(end, 1:end-1);

    [~,index,~] = fileparts(char(list(loop).name));
    output_lista.(index) = mean(ar);
    output_listb.(index) = mean(br);
```

```
    output_listd.(index) = mean(dr);

end
if(numel(list) > 0)
    output_list = {output_lista, output_listb, output_listd};
else
    output_list = [59), struct('nofilesfound',1)];
end
```

As shown above, the function would compare both the raw data (A), FFT data (B) and the FFT Peak Locations (D) with the live measurement and any other measurement stored within the library defined in the GUI. The results were displayed in real-time in the measurement window as a percentage for each of the outlined parameters. This gave the user the ability to compare the resemblance with three parameters to help confirm the result.

4.3.3 Image Script Function

The image script function was run separately from the GUI in a standalone script. This script loaded an image of the asset on screen so that the user could click the positions they wished the sensor to measure. These were then used to create a list of points relative to the translation stage measurement area and loaded into the GUI for the sensor to measure. To ensure the accuracy of the point list, the laser module was used to align the asset under measurement marks, so that the image could be scaled to fit the screen correctly. The full code listing of the function follows.

```
function [output] = imagescript(x1, y1, x2, y2, file)

if(nargin == 0)
fprintf('Please align the laser over the first alignment mark\n');
x1 = input('X = :');
y1 = input('Y = :');
end

if(nargin < 4)
fprintf('Please align the laser over the second alignment mark\n');
x2 = input('X = :');
y2 = input('Y = :');
%%
end

point3 = [x2 - x1, y2 - y1];

if(nargin < 5)
file = input('Please enter the name of the file incluing extension and
relative path:\n','s');
end

image = imread(file);
```

```
imshow(file, 'border', 'tight');

height = size(image, 1);

fprintf('Click on first alignment target\n');
waitforbuttonpress
point1 = get(gcf,'CurrentPoint'); % button down detected
fprintf('Click on secund alignment target\n');
waitforbuttonpress
point2 = get(gcf,'CurrentPoint'); % button down detected

if(point1(1,1) > point2(1,1))
   tmp = point1(1,1);
   point1(1,1) = point2(1,1);
   point2(1,1) = tmp;
end

if(point1(1,2) > point2(1,2))
   tmp = point1(1,2);
   point1(1,2) = point2(1,2);
   point2(1,2) = tmp;
end

plot_point1 = [point1(1,1), height - point2(1,2)];
plot_point2 = [point2(1,1), height - point1(1,2)];

plot_wh = plot_point2 - plot_point1;
wh = point2 - point1;

%output = [point1; point2];
rectangle('Position', [plot_point1 plot_wh])
hold on;

fprintf('Click to add measure. Press ANY key (any of the keys) to
EXIT\n');

output = [];

while(waitforbuttonpress == 0)

point = get(gcf,'CurrentPoint'); % button down detected
plot_point = [point(1,1), height - point(1,2)];
plot(plot_point(1,1), plot_point(1,2),'r.','MarkerSize',20)

point = point - point1;
%point(1,1) = wh(1,1) - point(1,1);
output = [output; round((point) ./ wh .* point3)];

end

name = input('Please give a variable for exporting:','s');
setappdata(0,name,output);
end
```

4.3.4 Point List Function

A ‘point list’ algorithm coded and run through Matlab defined the 9 points and stored the co-ordinates into a matrix variable. The matrix variable could then be loaded and run directly through the GUI for the sensor system. The sample could therefore be

clamped in the same position each time, ensuring that the same 9 areas were being measured by the sensor.

4.4 Sensor Design

The operational frequency band was the Ka-band, 18 – 26.5GHz. This was the rated frequency band of the FMCW radar module chosen for the driving the sensor as outlined previously. As the frequency bandwidth of the hardware is fixed, the design characteristics of the waveguide and horn antenna are somewhat reduced. Each frequency band has set dimensions for rectangular waveguides to ensure minimal losses. This therefore reduces the number of design freedoms for the sensor as the horn antenna of course must have the same dimensions as the waveguide at one end, meaning that the only notable changes are the dimensions of the opposite aperture and indeed the entire length of the horn flute. These two dimensions can be used to tune the horn to provide the best signal response within the set frequency band. For this the Ansoft HFSS software package was used to plot the radiation pattern shown in Figure 4-8 as well as the S1 parameter response of the horn antenna in Figure 4-9.

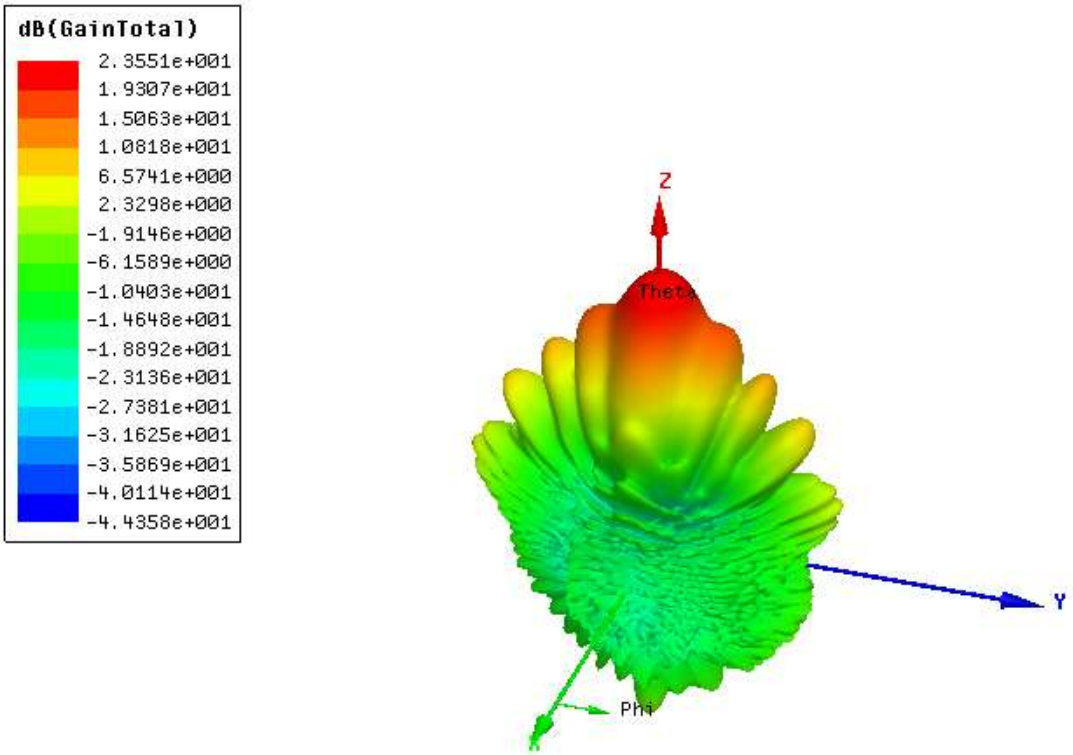


Figure 4-8- HFSS Simulated Radiation Pattern

By changing the dimensions noted above, it was possible to change the response of the horn antenna into the correct frequency bracket. Changing the horn aperture of the E

and H planes of the horn adjusted the location of the horn response. Changing the length of the horn flute determined the width of the response on the frequency bandwidth.

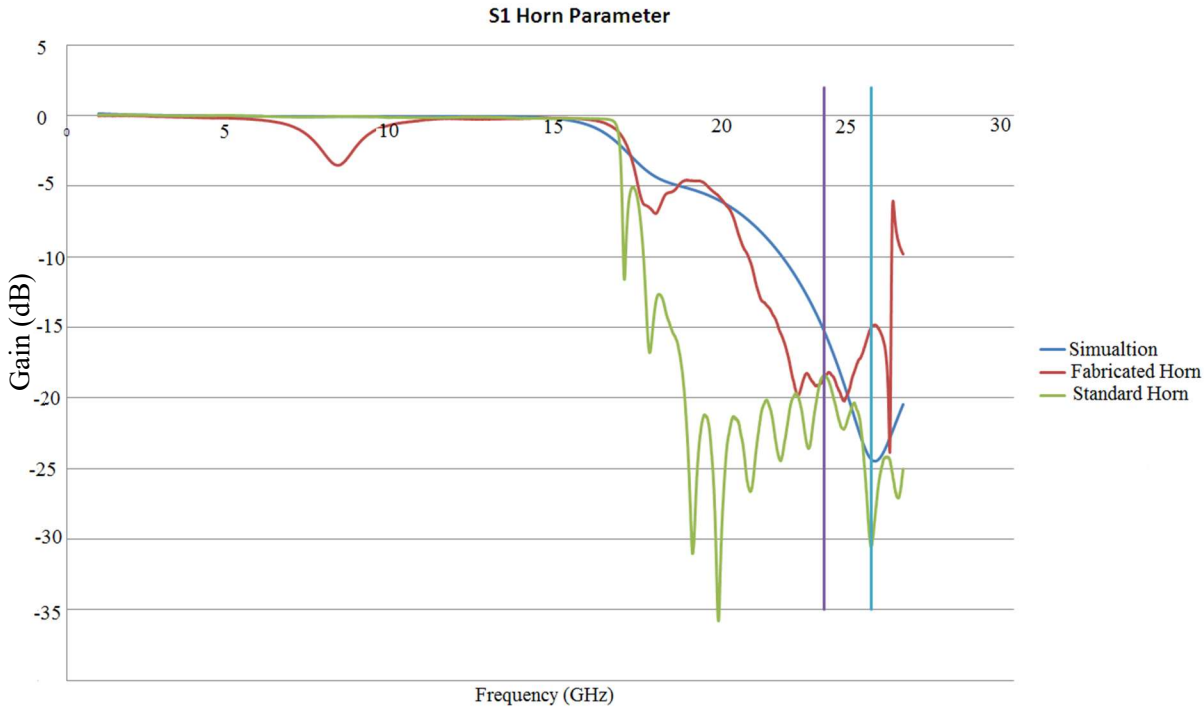


Figure 4-9 - Horn Response

Figure 4-9 shows the horn antenna response of the simulated horn versus both the standard gain off the shelf horn and the prototype fabricated horn. The figure shows the S11 parameter in dB plotted against frequency in GHz. The S11 parameter is the most commonly used parameter in terms of antennas. It represents how much power is reflected from the antenna and is known as the reflection coefficient. The S11 data for the simulated horn is exported directly from the Ansoft HFSS software, where the data for the standard and fabricated horn was collected using the Programmable Network Analyser (PNA).

When looking at the data it should be noted that where S11 tends toward zero, this indicates that the antenna will radiate nothing as all the power is being reflected. Where S11 is less than -10dB or in this case -20dB this shows that the antenna radiates well.

It can be seen that the simulated results show that the horn radiate marginally outside the proposed operational bandwidth, but S11 is still less than -10dB across the whole operational band. The standard gain horn, as expected, has a large operational band width over the whole range of the Ka-band and is ideally suited to this operation. The

prototype horn antenna however, behaves differently than the simulated results suggested. The physical experimental results show that the horn actually operates as intended in the 1500 MHz bandwidth required. This is due to the physical imperfections in the production and the manufacture of the individual components. The fabricated horn also uses a fabricated waveguide. The exact dimension of the waveguide may be slightly out with tolerance due to mechanical manufacturing limitations, creating a different frequency response and in this case it has changed the antenna behaviour toward what was required. The final dimensions of the horn antenna are defined in Figure 4-10 and listed in Table 4-2 – Horn Antenna Dimensions.

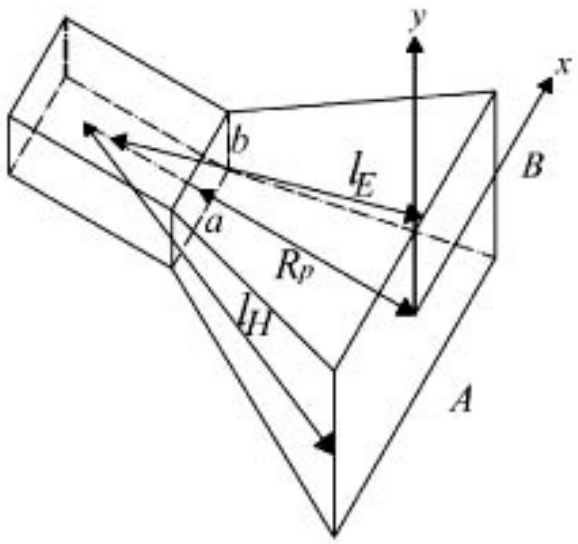


Figure 4-10 – Geometry of Pyramidal Horn Antenna

Dimensions	
A	**77.3 mm**
B	**61.5 mm**
R_p	**137 mm**
a	**10.688 mm**
b	**4.318 mm**

Table 4-2 – Horn Antenna Dimensions

It was deemed following initial tests that both the standard horn and the fabricated horn performed to a similar level, measuring similar sized defects with similar responses. The only difference between the horns was the spot size of the antenna. The standard horn antenna had a more direct beam pattern in comparison with the fabricated horn due to the perfect matching between horn and waveguide and the accuracy of the

manufacturing. The horn was sourced from Flann Ltd and was calibrated for the specific frequency band, further improving its response.

4.4.1 Sensor Manufacture

The horn antenna was constructed using Poly Methyl Methacrylate (PMMA) sheets due to its availability and easy of manufacture. The modelled horn was exported into CorelDraw technical drawings software and deconstructed. The horn was spilt into four different faces of the pyramid and cut on a laser cutter.

PMMA is technically 'invisible' to microwaves at this frequency due to its low dielectric constant (2.6), therefore it is unable to store electrical energy from the electric field which allows the electromagnetic wave to pass straight through. [60] The cut pieces of PMMA therefore had to be lined with a Copper Wire Mesh with aperture smaller than that of the wavelength. The wavelength of the centre frequency (24.75 GHz) is 12.1 mm; therefore the aperture had to be smaller than this. The chosen mesh had an aperture of 0.263 mm, much smaller than required to minimise leakage.

Once the mesh was secured to the internal side of each of the sections, the horn was fixed together using M3 steel screws. The screws did not affect the performance of the horn as these were on the outside of the mesh material, meaning that the MW energy did not come into contact with them. The mesh was cut to overlap slightly on all joining edges to once again minimise the leakage along the joins. The completed horn is shown in Figure 4-11.

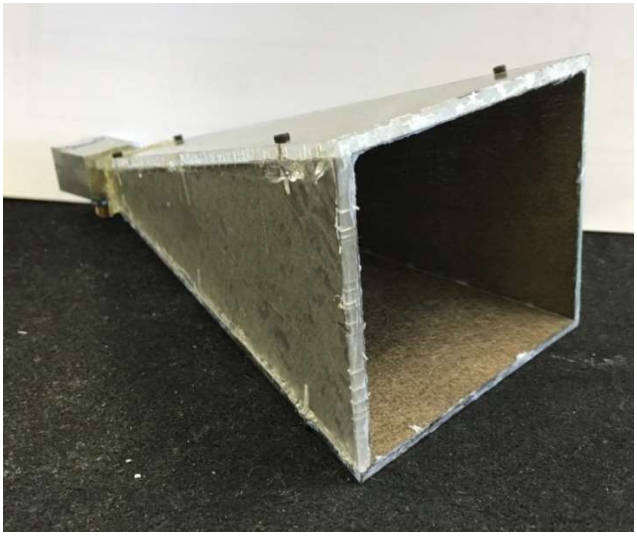


Figure 4-11 – Fabricated Horn Antenna

Chapter 5- Experimental Results and Validation

5.1 Experimental Protocol

All experiments discussed within the thesis had to follow the same protocol to ensure that results were accurate and repeatable. The translation stage discussed in the previous sections was a major part of ensuring the repeatability of the experiments. In using the translation stage samples were fixed in place meaning no external factors could affect the sample to sensor distance. The translation stage itself did not move the sample; it moved the sensor in relation to the sample surface. The distance between sample and sensor could be adjusted using the z-axis, but this axis was not included in the automatic calibration at the beginning of every measurement to ensure that the critical distance between the sample and sensor was not changed accidentally. The air-gap created in the z-axis was must remain fixed when looking at the same sample. Differences in this gap would cause a signal change due to the increased volume of air between the sensor and the sample and therefore effecting the E filed due to the exposure of the permittivity.

Throughout the following section, the sample construction will be described using the layer structure as shown in Figure 5-1. For the purpose of the following experiments, no more than 3 layers have been considered, even though it is possible for pipelines with greater number of layers to be used in industry. Layer 1 will refer to the metallic outer surface of the pipeline. Layer 2 represents the thermal insulation layer and layer 3 the outer protective layer to seal out external conditions.

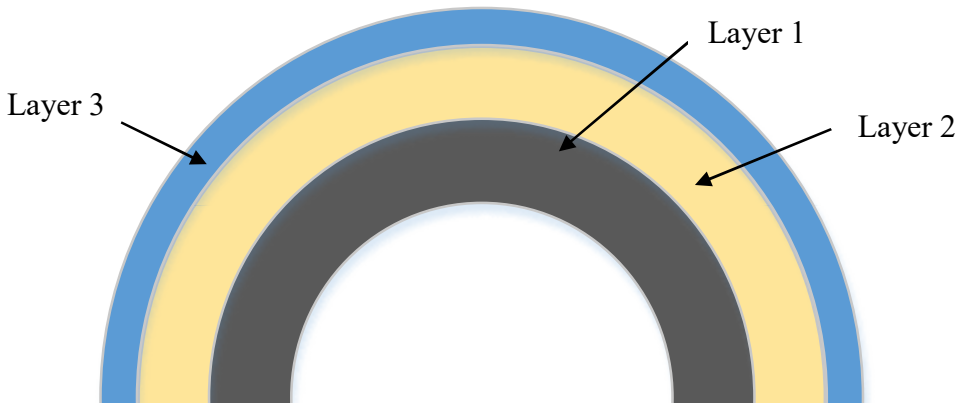


Figure 5-1 – Pipe Cross Section

5.2 Experimental Setup

The experimental setup for each of the experiments is detailed in relevant sub sections below. Each experiment shared the same fundamental experimental protocol as outlined in the previous subsection, but the specifics of each of the varying experiments are given below.

5.2.1 Copper Sheet

The first experiment conducted used machined defects in a copper sheet. The purpose of this experiment was to validate the sensors ability to detect defects compared to a health baseline and by varying the feature size of the defects establish the sensitivity of the sensor when assessing metallic surfaces (layer 1). The experiment would also form the beginning of the defect library for the sensor historical database which could be used in conjunction with 'correlate' function outline in section 4.3.2.

The copper sheet shown in Figure 5-2 was of 3mm thickness and had dimensions of 150mm x 150mm. Defects of various size and depth were milled into the surface of the sheet equidistantly along the surface.

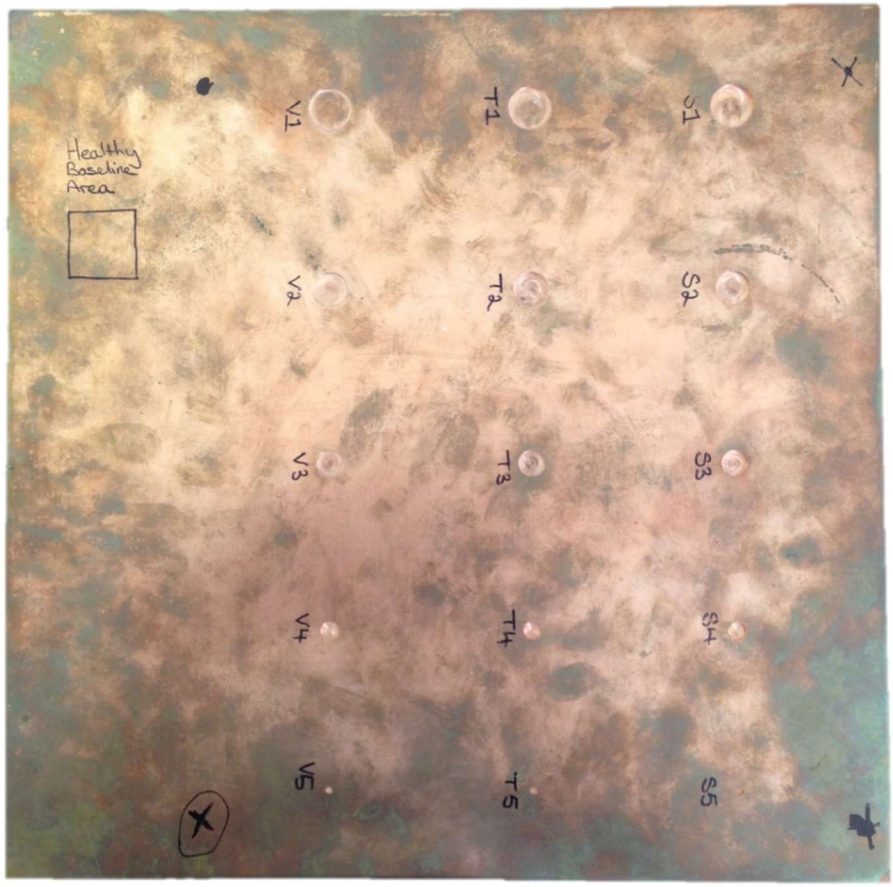


Figure 5-2 - Copper Defect Sheet

The defects were distributed in rows of corresponding depth 'S, T and V'. The deepest defects were 1.5mm and labelled 'S', the shallowest group of defects were labelled 'V' and were 0.5mm deep. The centre row labelled 'T' were 1mm deep, a cross section of the defects depths is shown in Figure 5-3. Each row of defects had identical surface area sizes with the largest being 15mm diameter and the smallest having a diameter of 2mm.

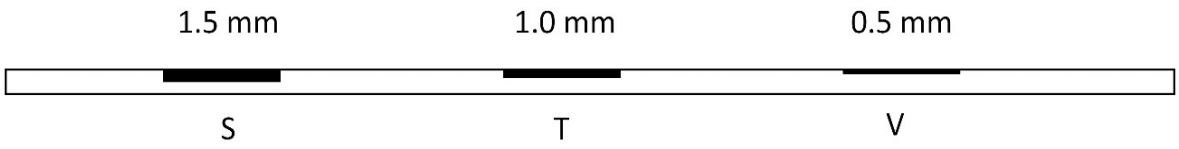


Figure 5-3 – Cross Section of Copper Defect Sheet

To gain an understanding of the sensors ability to determine defects, the sample was placed below the sensor with an air-gap, between sensor and sample, set at the maximum z-axis distance of 45mm.

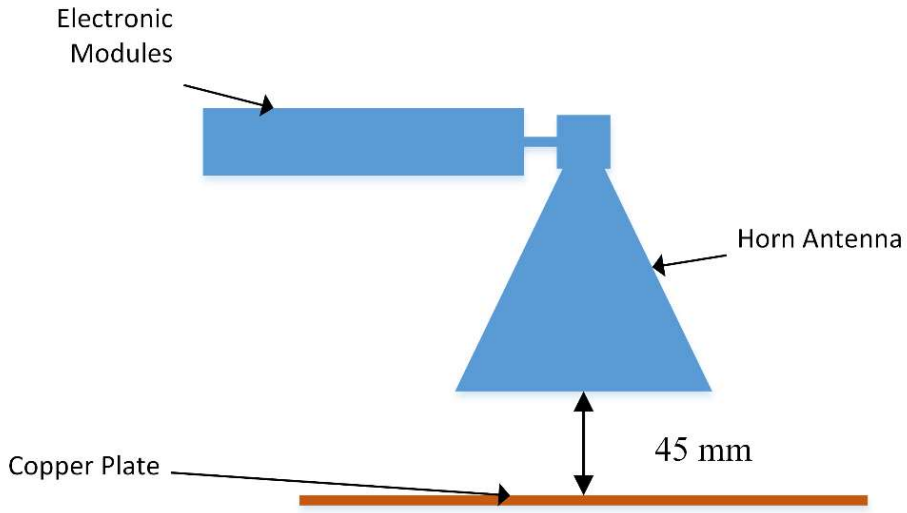


Figure 5-4 – Copper Sheet Test Setup

Using the setup as shown in Figure 5-4 meant the sensor spot size would be at its largest, maximising the opportunity of locating the large defects. The laser alignment module ensured the sensor was located over the centre of each of the largest defects, 'S1, T1, V1' in turn so as one measurement could be taken of each.

5.2.2 Copper Sheet with Dielectric Layer

This experiment looked to establish the ability of the microwave sensor to penetrate – see through – a dielectric coating as per traditional CUI preventative asset protection. To validate this, the sensor had to be able to detect the same changes in the copper sheet

as in the previous experiment, but with the addition of a dielectric material acting as an insulating layer (layer 3). To simulate this, a single 5mm thick piece of PMMA was placed on top of the copper sheet. The sensor was then aligned over each of the large defects in turn using the laser module as in the previous experiment. The air-gap between sensor and copper plate remained the same to identify the exact effect the dielectric layer would have on the sensor response. Even though PMMA is considered invisible as described in section 4.1.1, the experiment was required to prove that this would work in practice. PMMA and HDPE have very similar dielectric properties, therefore are both technically invisible to electromagnetic waves, making PMMA the ideal test piece for insulation. [61]

The experiment was completed with multiple layers of PMMA up to 20mm due to the limitation of the z-axis movement of the translation stage.

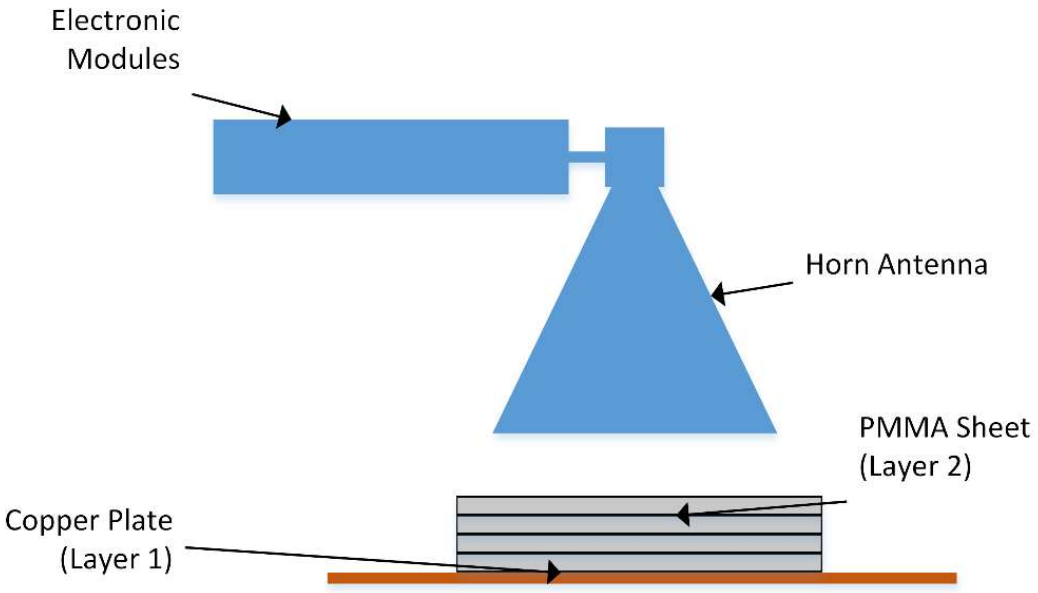


Figure 5-5 – Copper Sheet with Insulation Layer Test Setup

5.2.3 Insulation Water Ingress

The following experiment was conducted to determine the feasibility of using the developed sensor to detect moisture in insulation as a precursor to CUI onset. As discussed in section 2.2, some techniques for CUI identification rely on the presence of water as a marker that corrosion may be present. Although effective, these experiments do not quantify the amount of water in the insulation. The aim of this experiment was to firstly determine whether water could be sensed within the insulation and secondly to check the feasibility of quantifying the amount of water absorbed into the insulation.

This experiment was conducted using a 3 layer sample to simulate a fully constructed insulated pipe similar to Figure 5-1.

As can be seen in Figure 5-6, the experiment was conducted with layer 1 represented by a 3mm thick carbon steel plate. Layer 2 of the sample used Phenol Formaldehyde (PF) foam to represent the internal thermal insulation. Finally, the outer protective coating was simulated with a 5mm thick uniform sheet of PMMA.

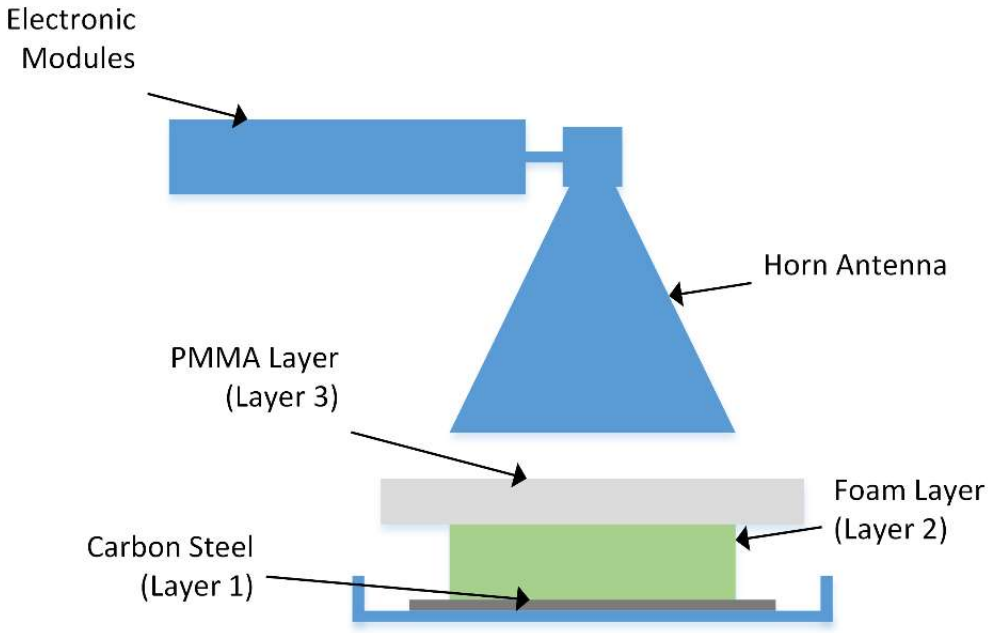


Figure 5-6 – Wet insulation Setup

The sample was placed in a moisture collection tray where it was clamped and marked in position beneath the sensor. The air-gap between the sensor and the third layer of the sample was fixed at 5mm. The air-gap was changed to accommodate the increased height due to this samples construction, within the limited z-axis of the translation stage. The z-axis motor of the translation stage was disconnected during this experiment to ensure that no change in this air gap could be made by either hardware or software glitches.

Moisture was added to the sample using a high accuracy pipette at 5 mL intervals. The top layer of PMMA was removed and the water added in the centre of the sample to allow for uniform absorption. After the addition of moisture the sample was allowed to settle for a period of 5 minutes to allow the moisture to be absorbed throughout the

foam section. The type of foam used is specifically designed to absorb and hold water so was ideal for this experiment.

After the predefined settling time was reached, 3 measurements of the sample were taken at 30 second intervals. In taking the measurements at these intervals it was possible to confirm if the moisture had settled within the sample. If variations were seen in the 3 measurements then the settling period would be run again. Once the 3 samples had the same signature, then a final measurement was taken.

To begin the next measurement, layer 3 was removed from the sample and moisture added. The layer was then reinstated in the same position which had been marked on the PMMA and the process repeated.

5.2.4 Polymer ageing

The polymer ageing experiment was conducted with the support of the polymer chemists based within the school of Engineering and Physicals Sciences (EPS) within Heriot-Watt University. The experiment aimed to determine whether the sensor could detect changes in the polymer condition as the sample was aged using heat.

In line with advice from the polymer chemists it was assumed that ageing the polymer would results in physical changes. Firstly the polymer would potentially suffer from contraction due to the heat cycling during the ageing process. This would be due to the bonds which make up the sample breaking. Secondly the polymer would begin to breakdown becoming more porous and brittle. This would affect the electrical characteristics of the material, i.e. the dielectric constant. Although very low when health, an aged polymer can exhibit increases due to physical changes. [62]

Samples were prepared using a hot press, in a mould which was specifically designed with 75mm diameter and a thickness of 1mm to fit within the sensor footprint. 4.42g of Styrene Acrylonitrile (SAN) powder was used to make the sample. The same material had been used to create smaller experimental samples to perform Dynamic Mechanical Analysis (DMA) and Differential Scanning Calorimeter (DSC) and these results would be used to determine the extent of the ageing on the polymer.

DMA is used to determine material characteristics of polymers. The measurement studies the viscoelastic behaviour of the material which is an assessment of the viscosity and elasticity of the material while a stress is being applied to deform the material. [63, 64] DSC is a thermal analytical measurement. This technique is used to determine the thermal transitions within polymer materials. [65, 66]

For the purpose of this experiment the accelerated ageing of the samples was conducted in a vacuum oven at 14.7 psi and 75ºC ± 5ºC. The prepared samples were measured with the sensor before being place in the oven for varying periods of time. After each specified duration in the oven, the samples were given a settlement period of one hour to return to ambient temperature and pressure before being measured again.

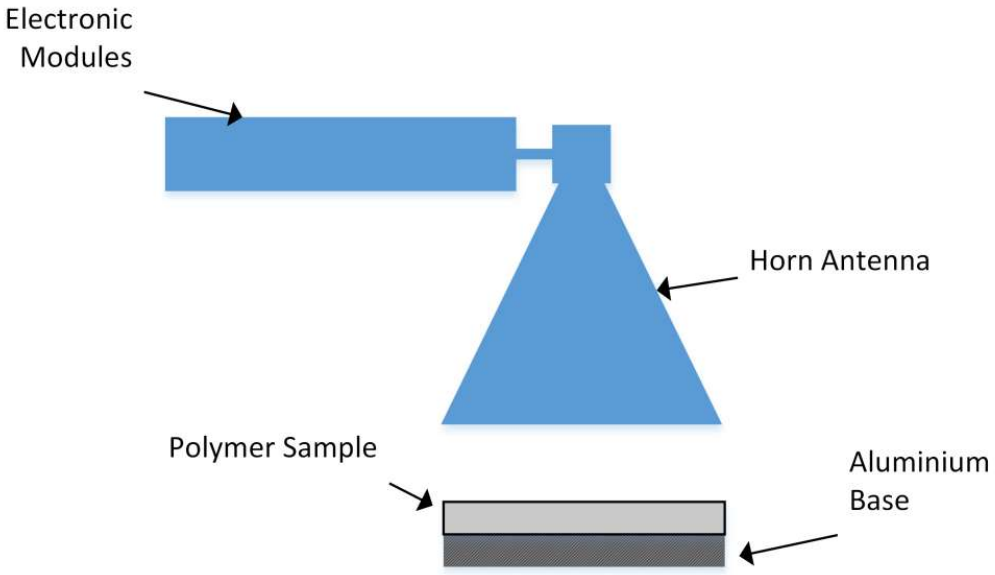


Figure 5-7 – Polymer Ageing Setup

As shown in Figure 5-7 the base from the aluminium mould was used as a reflection platform for the sensor. The sensor was aligned above the centre point of the sample and the co-ordinates of this point were used to determine a 3 x 3 point matrix. Taking measurement using the nine points around the centre of the sample confirmed that the beam of the sensor did not take any measurements out with the profile of the sample, ensuring that the signal response was linked only to changes in the polymer material rather than other external conditions.

5.2.5 Accelerated Corrosion Steel

To determine the sensors ability to measure gradual CUI onset an accelerated corrosion experiment was designed based on the on the reverse concept of an impressed current cathodic protection system (ICCP).

An ICCP is basically an electrochemical cell which protects the metal asset from corrosion by assigning it as the cathode. The anode will be made of a 'sacrificial metal' that is more readily corroded. The electrolyte solution is moisture or surrounding media such as soil. The ICCP is completed with the addition of a DC current flowing between the anode and cathode as shown in Figure 2-3.

Based on this, Figure 5-8 shows the setup used to increase the corrosion rate of the carbon steel plate chosen for the experiment.

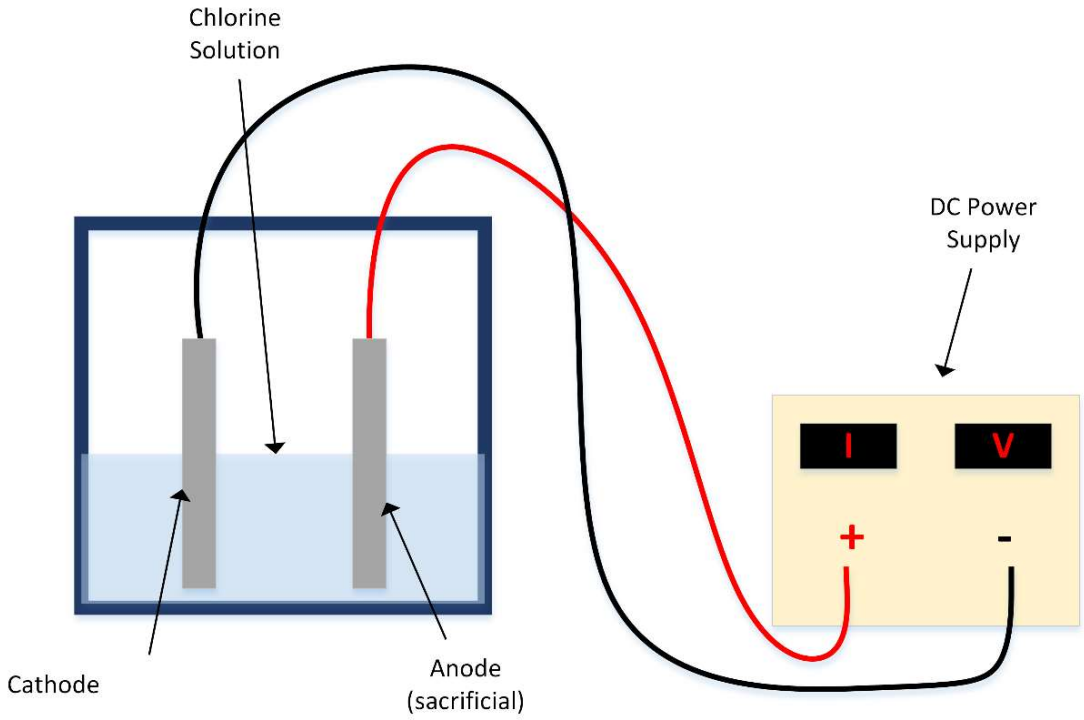


Figure 5-8 – Accelerated Ageing Setup

The 'sacrificial metal' was exactly the same metal as the cathode, but it had been placed in a chlorine solution for 24 hours to begin the corrosion process. By starting the corrosion process before being used as the anode, this made the plate easier to corrode meaning it would respond well to the accelerated ageing.

The electrolyte solution was formulated using high strength chlorine tablets used to clean equipment for sterile environments. The chlorine tablets dissolved in 1 litre of water making a 10000 ppm strength solution which was highly corrosive. This coupled with the 840mA current at 20V proved to be a very effective accelerated corrosion environment.

The carbon steel plates were exposed to varying time durations to achieve a wide range of corrosion effects. Once completed, half of the plate was cleaned using fresh water to remove the oxide layer as shown by Figure 5-9. The sample was then left to dry overnight before measurements were taken.

Figure 5-9 – Measurement Areas of Accelerated Ageing on Steel Plate

A measurement of the area unexposed to the solution was taken as the reference healthy baseline condition, and one measurement of both the clean and uncleaned exposed area was taken.

5.2.6 Painted Coating

AkzoNobel provided a new painted sample containing an area of corrosion beneath the paint layer from their pain coatings business. The company specialises in coatings for all industries to protect assets from potentially hazardous external factors. This particular coating is used to protect pipelines from the onset of CUI, sealing out external parameters to reduce the risk of moisture coming into contact with the external pipe surface.

The sample provided was a 30cm x 10cm steel plate with a predefined area of corrosion measuring 3cm x 3cm underneath the applied coating. The coating used contained aluminium pigmentation flakes throughout. As the coating contained metal particles it was not known if the sensor would be able to determine the area of corrosion due to reflections from the surface.

The sample had been prepared by hand therefore the painted coating was not evenly distributed across the whole of the sample surface. The changes in the paint thickness can be clearly seen in Figure 5-10. This changing surface finish could potentially mask the corrosion due to the undulations in the coating.

Figure 5-10 – Aluminium Coated Sample

This experiment was conducted in a similar fashion to those done previously. The sample was secured to the centre of the translation stage with a 5mm thick piece of carbon steel providing a splint effect to remove the curvature in the measurement plane as shown in Figure 5-11.

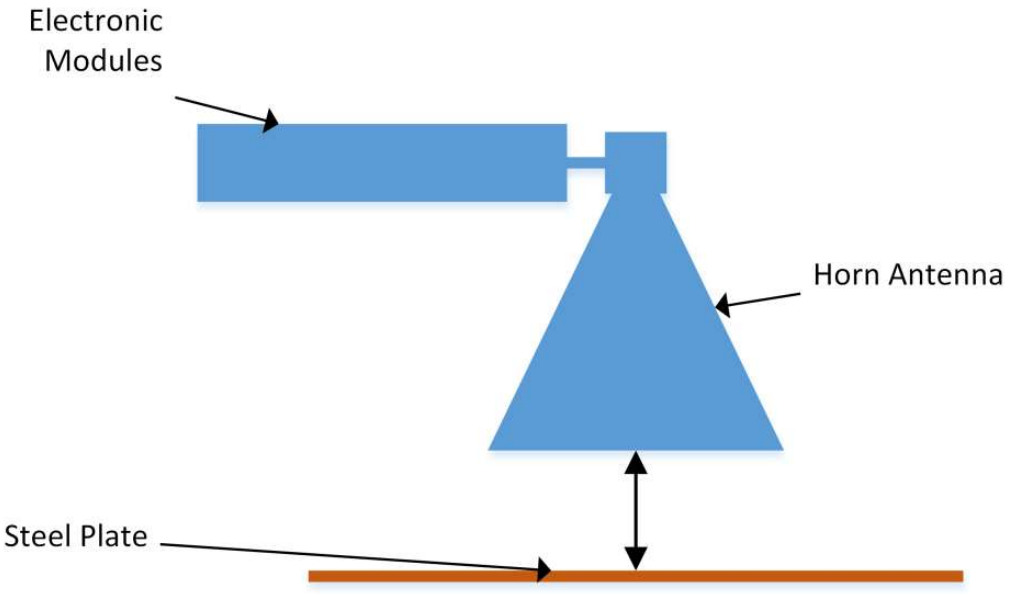


Figure 5-11 – AkzoNobel Sample Setup

The sensor was programmed to take a measurement every 50 steps along the centre line of the sample in its longest dimension. The air gap between the sample and sensor was 30mm and no insulating layers were added.

5.3 Experimental Results of Corrosion Using Horn Antenna

The results from the various experiments have been outlined under the relevant subheadings, with full analysis and discussion of the results. The results from the experiments are displayed in graphical form, with all the graphs plotted on the same axis. A single data stream has been shown in Figure 5-12 as an example of the typical data.

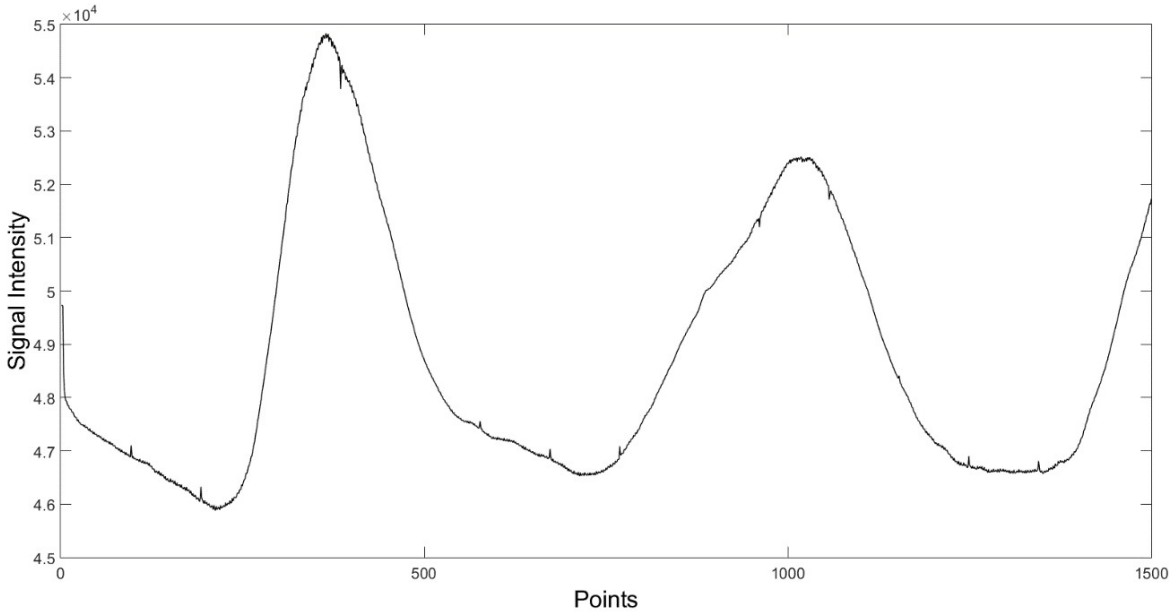


Figure 5-12 – Example Data Plot

As can be seen the x-axis is labelled as points. The 'points' axis is the number of points taken specified when programming the board as explained in section 4.2.1. Each point on this axis is representative of a frequency in the sweep, i.e. sweeping the whole 1500 MHz bandwidth with 1500 points equates to each point representing 1MHz. Therefore, point 0 is effectively 24GHz and point 1 is 24.001GHz. The final point '1500' would represent 25.5GHz, the maximum frequency in the hardware's capability.

The y-axis represents the signal magnitude in terms of phase difference. As discussed in section 3.5, the difference between the sent and received signal is known as the IF signal. The FMCW hardware being used for the experiments represents the IF signal as a phase difference (**φ**) between the sent and received signal. Therefore each data point on the graph is the magnitude of the phase difference between sent and received signals plotted against the corresponding frequency for that transmission.

5.3.1 Copper Sheet

The results of the measurements taken from the copper sheet are outlined in this section. Figure 5-13 below is a simple plot of the three largest defects as discussed previously.

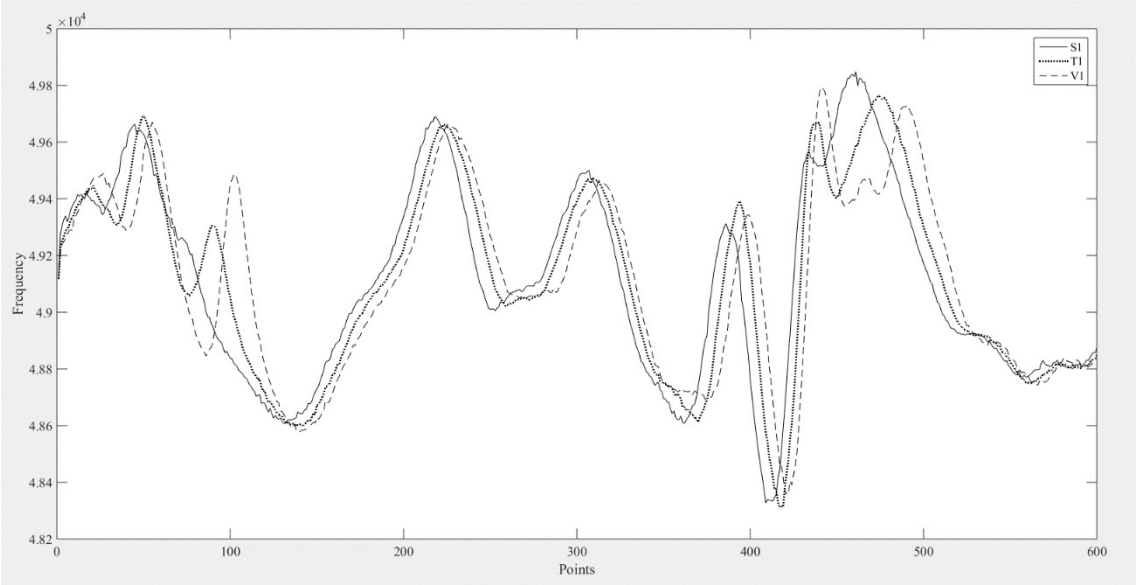


Figure 5-13 - Three Depth Defects

As can be seen from the results above, the sensor can clearly define between the depths of the largest defects in the way of a phase shift. As the FMCW radar has a primary purpose of detecting distance, this was the anticipated result. The phase shift seen above are due to the time of flight difference between the defects. As the depth of the defect is increased, the time taken for the EM wave to travel to and from the target is also increased. As explained in section 3.5.1 and at the beginning of this section, the data points on the graph are representative of the phase shift between the sent and received signal at each swept frequency.

In terms of EM parameters, the shape or signature of the signal is defined by the permeability of the sample as outlined in section 3.6. As there is no dielectric material present in this experiment other than air, permittivity has no effect on the signal.

Other small features are visible from the plots, but these are not always repeatable and can be due to minor variations in the horn movement. Adjusting the sensor averaging function, removes most of these discrepancies by taking multiple measurements of the chosen area and only sending back the averaged data to be plotted.

5.3.2 Copper Sheet with Dielectric Layer

The following results show the effect of increasing the layer thickness of PMMA when measuring the three largest defects as in the previous experiment.

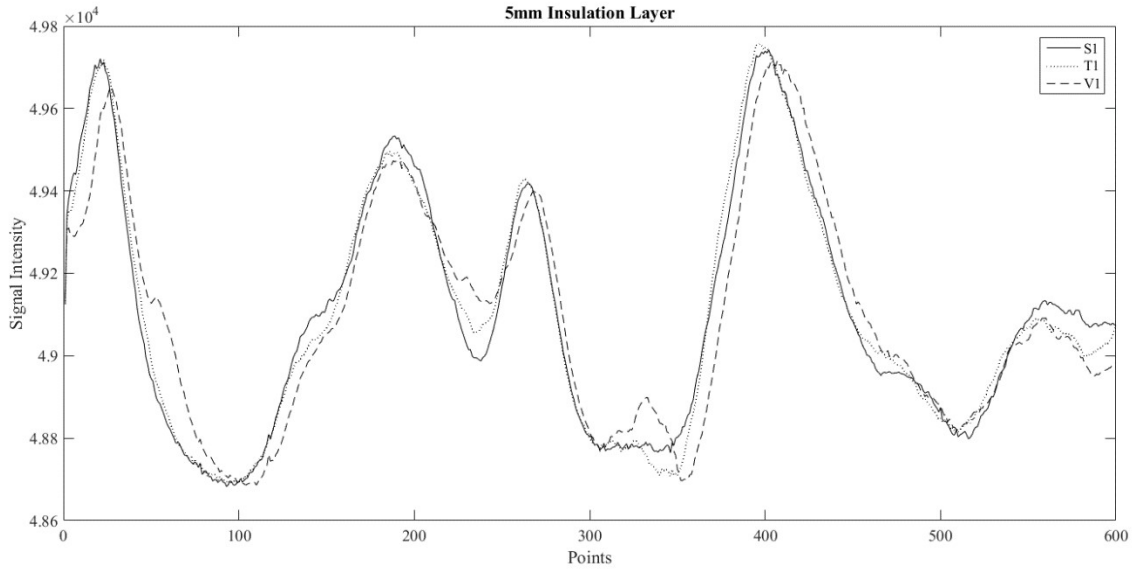


Figure 5-14 - Defects with 5mm Insulation

As shown above by Figure 5-14, the addition of the insulation layer changes the signal response/ shape quite dramatically. Although a phase shift is clear between the defects due to the time of flight measurement as seen in the previous experiment, the overall signal features have been reduced. Figure 5-13 contains 8 notable signal peaks which are common to all of the defect traces, but the addition of the 5mm layer of PMMA has reduced the number of to 5 which are common to all traces. This reduction in features is due to the permittivity of the dielectric layer changing the E field of the sent signal. At this stage this feature reduction could be used to characterise the presence of the insulation layer, with the original phase shift able to determine the defect depth.

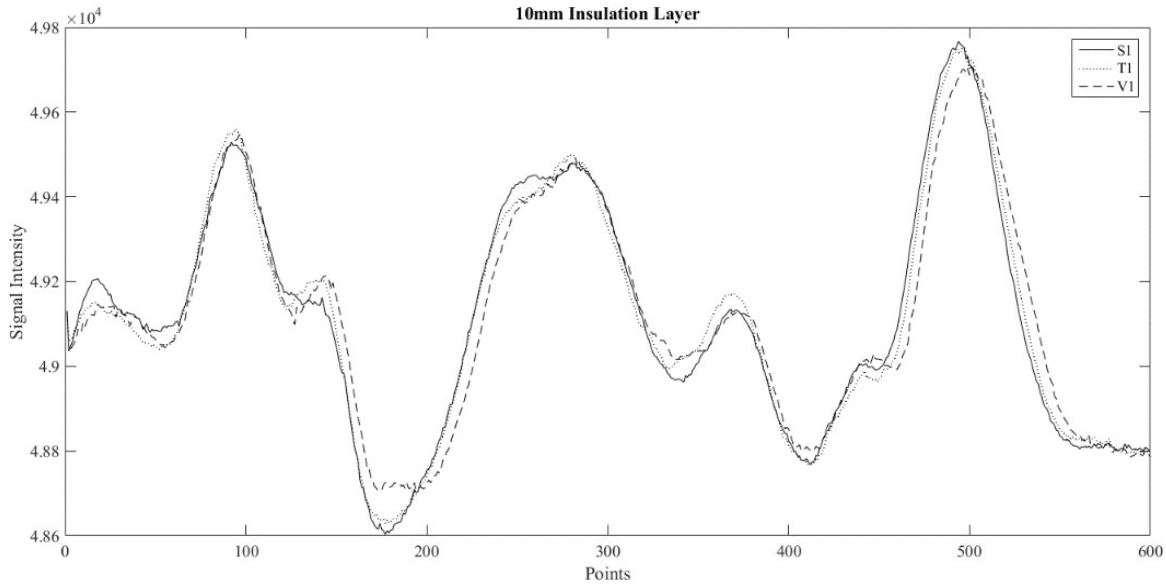


Figure 5-15 – Defects with 10mm Insulation

Figure 5-15 contains the results of the defect measurements with the addition of a second PMMA sheet, increasing the overall thickness to 10mm. As can be seen from the results the magnitude of the whole signal has not been significantly reduced, as expected as the magnitude of the signal is only representative of the phase or time of flight. As the defects are common to all the results, the intensity of the phase change should remain largely constant. The signal response has however changed somewhat in terms of the position of the peaks and number of peaks. Many of the peaks have moved position and shape. This change in response is due to the change in the time of flight which is increased by thickening layer of insulation. The addition of the PMMA as layers also creates more boundaries between air and the insulation material. As the wave travel through the insulation, it interacts with the air gap between each layer. This interaction creates a small reflection of the signal which effects the overall signal response.

The increased layer thickness also increases the amount of energy exposed to the dielectric constant. As explained in section 3.6, the dielectric constant of the material being measured reduces the net effect of the electric field therefore reducing the intensity of the returned signal, which in turn changes the phase of the signal and thus the shape of the plot.

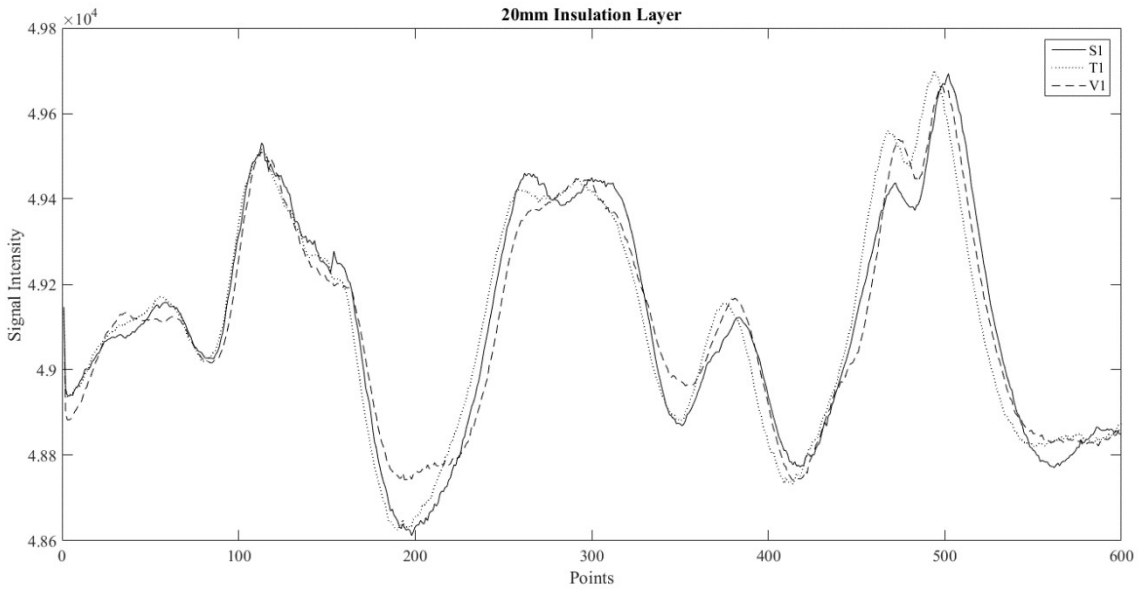


Figure 5-16 – Insulation with 20mm Insulation

The final result from this experiment, shown in Figure 5-16, further confirms the signal changes described previously. The signal magnitude remains high as with the other change in thicknesses, showing that the PMMA causes little attenuation to the sensor signal. At each of the peaks in the signal, small changes in amplitude and phase can be seen between the measured defects which are constant with the results of the copper defects without insulation layer.

This experiment has shown that the insulation will become a limiting factor in determining defects. As the insulation thickness increases, the net dielectric property of the insulation material will out-weigh the electric field supplied from the sensor. This will therefore limit the sensors ability to detect corrosion through insulation layers.

In summary this experiment has shown that the addition of insulation layers effect only the shape and features of the signal due to the permittivity of the material affecting the E field supplied from the sensor. As expected, the phase difference between the defects remained as these were constant throughout.

5.3.3 Water Ingress

Once the experiment was conducted the results shown in Figure 5-17 showed a clear signal change in both magnitude as well as phase when comparing the insulation with and without moisture content.

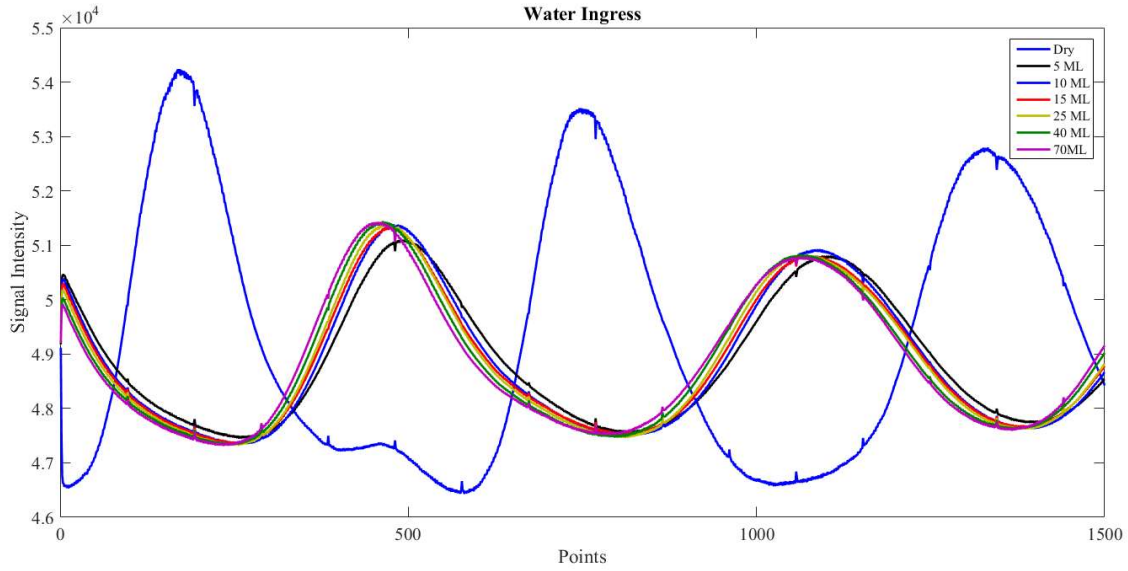


Figure 5-17 – Wet Vs Dry Insulation

The dry insulation shows a much larger signal intensity as this is the reflection from the base steel plate. The insulation has such a low permittivity that it has no effect on the E field, similar to the copper sheet experiment with the insulation layers. Therefore the signal is purely based on the time of flight to and from the steel plate. The permeability of the copper has an effect on the shape and features of the signal.

As water is added to the sample there is a large drop in the signal intensity. This drop in intensity represents a reduced time of flight measurement due to the signal now reflecting of the water rather than the steel plate. The water has increased the permittivity of the insulation meaning that the incident wave is reflected earlier. The similar shape of the signal is still due to the permeability of the steel plate.

Figure 5-18 below gives a concentrated view of the results, focussing on the response due to the level of moisture in the sample.

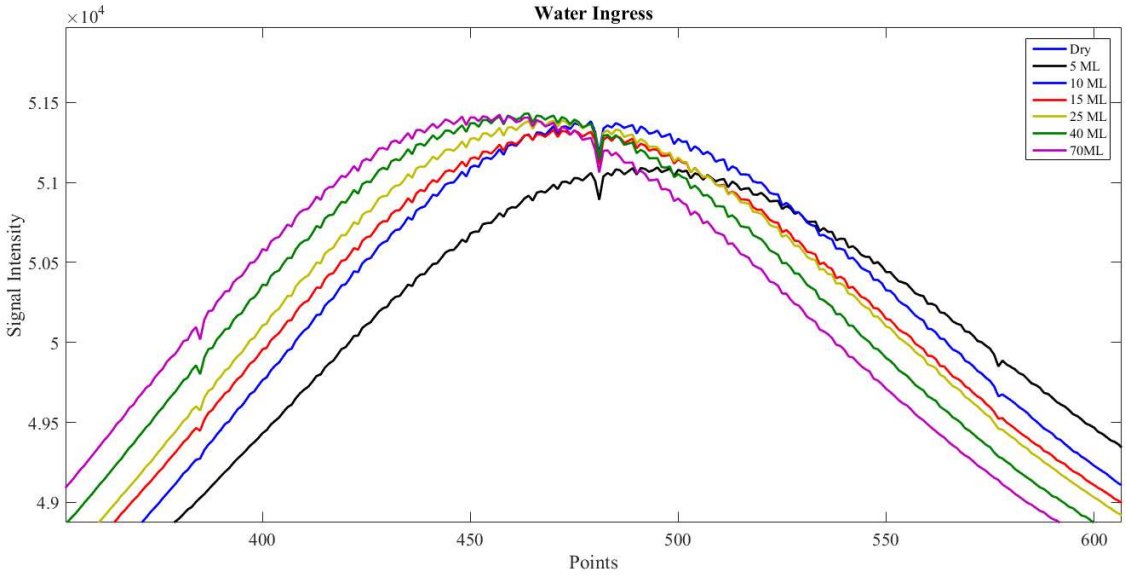


Figure 5-18 – Zoomed Water Ingress

The results above confirm that the sensor can not only determine when moisture is present in the insulation, but also has the feasibility to determine the amount of water present down to relatively small quantities. As the concentration of the water is increased so is the amount of signal reflected by the foam. This will continue until the foam becomes saturated. The small phase shift is attributed to the time of flight of the signal changing relative to each addition of moisture.

To put the level of sensitivity into context, a study by Yu et al, [59] determined that polyurethane foam absorbed 8% of its weight in water. The sample used in this experiment weighed 13 grams when dry. Adding 5ml of water took the sample weight to 18 grams in total and therefore the water – weight ratio was 27%. Although this is much higher, the foam type used in the experiment was designed to absorb water, is less dense and weighs less when dry. The polyurethane foam used by Yu is not designed to hold water and is therefore more dense and heavier when dry.

5.3.4 Polymer Ageing Results

The results below, Figure 5-19, show the sensor response to the polymer sample after being aged in the oven up to 23 days total time.

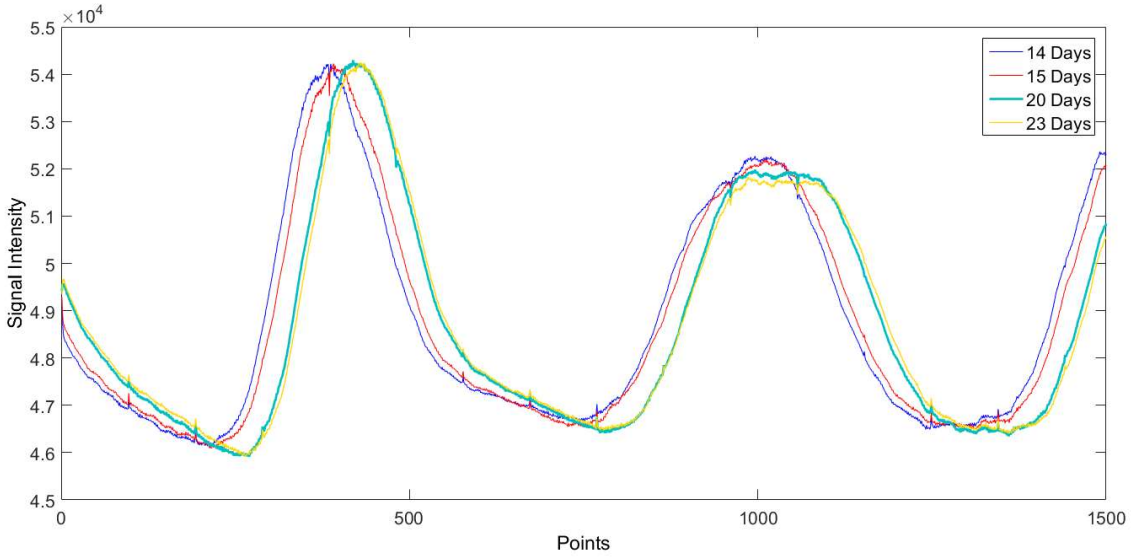


Figure 5-19 – Aged Polymer Results

The sensor response remains very similar throughout the measurements, but does contain slight changes which signify the process of ageing between the two groups, but also individually as well. The 14 and 15 day ageing result and the 20 and 23 day results grouped together as expected.

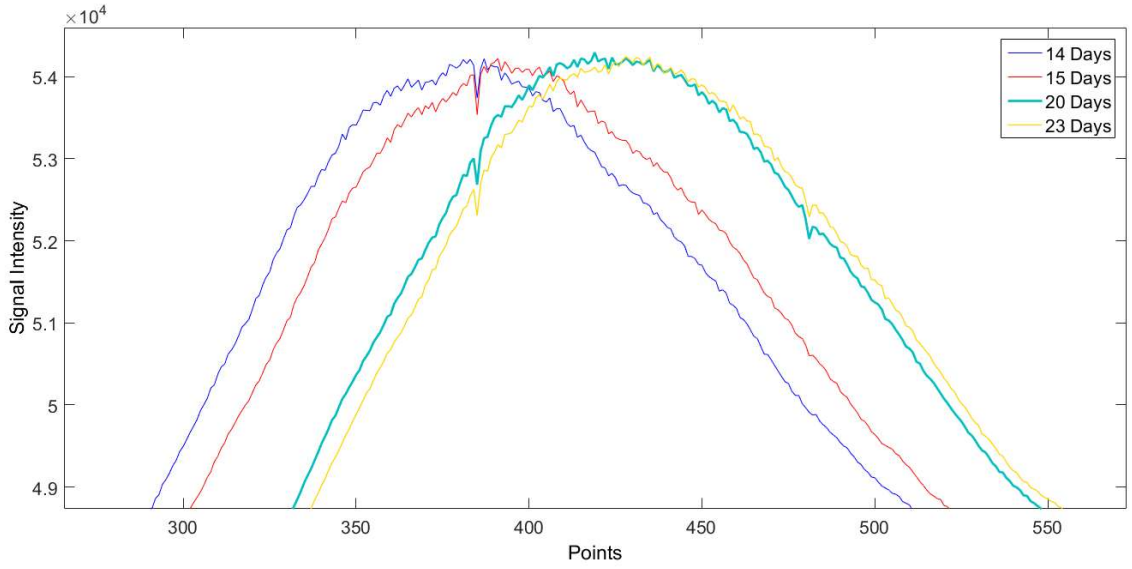


Figure 5-20 – Zoom of Aged Polymer 1

Figure 5-20 looks closer at the initial peak of the signal. The maximum value for each of the traces moves in terms of phase to the right. This could be used as a classification to determine the age state of the polymer along with a secondary classification using the second peak.

The signal changes observed in the results are primarily the effect of the dielectric constant changes in the polymer. As the polymer aged the dielectric constant of the material also changes. This therefore is altering the E field from the sensor.

As shown in Figure 5-21, the maximum value of each trace in the second peak decreases as the sample is aged.

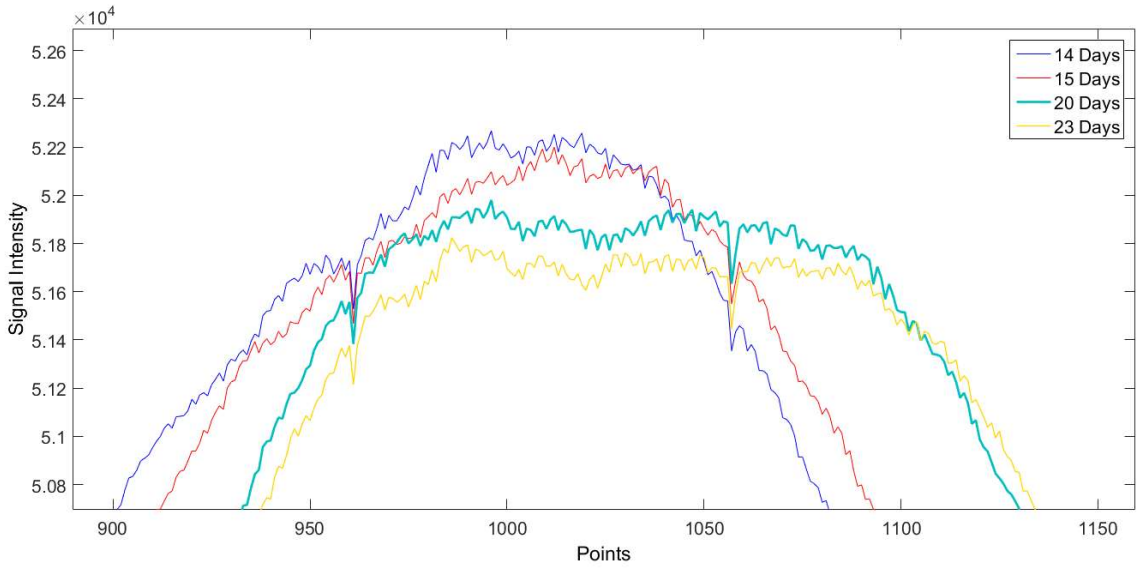


Figure 5-21 – Zoom of Aged Polymer 2

This change in intensity of the signal in line with other experiments should represent a change in time of flight measurement. This is most likely due to a mechanical change in the polymer, i.e. a change in thickness. It is not known what specifically happened to this sample during the ageing process and therefore further experiments are required to determine whether this is an isolated or repeatable case. It was the opinion of the polymer chemists that this sample was not prepare properly in the first instance meaning that the heat ageing was merely completing the curing process. Future work will look at using the sensor to determine curing level in polymers to check this thought.

5.3.5 Accelerated Ageing Steel Results

The following results displayed show the sensors ability to determine more realistic corrosion defects. This section will present the results of the various carbon steel plates corroded using the accelerated ageing process.

The sample shown in Figure 5-22, was exposed to the chlorine solution for 2 hours with an 880 mA DC Current applied. Under these conditions the sample formed a oxide layer and was subject to a very small amount of metal loss

Figure 5-22 –Accelerated Sample with Early Rust Formation

The result of measuring the healthy baseline material, which was not exposed to the solution, along with the corroded area is shown in Figure 5-23.

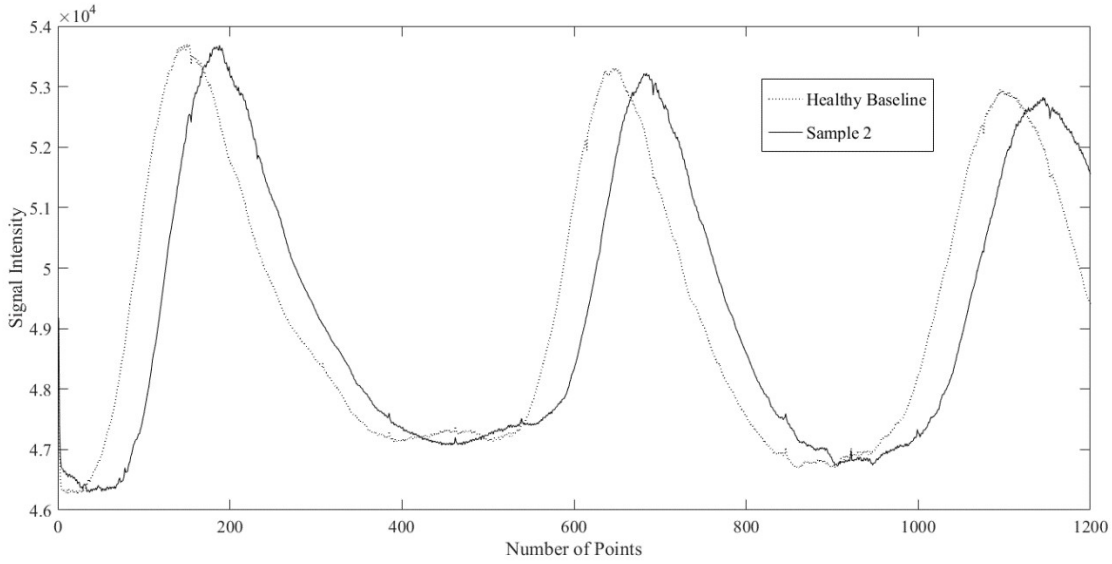


Figure 5-23 – Rust Formation Results

It can be noted that the healthy condition achieved a larger magnitude than the corroded area. This is due to the slight metal loss in the corroded area of the sample due to the increased time of flight measurement. The phase shift between the traces is significant enough to clearly identify the presence of the corroded area. This phase shift can be linked to both the permittivity and permeability of the sample being affected. The permittivity of the sample has changed with the introduction of the rust layer, forming a dielectric layer and the permeability of the sample being altered by the reduction in overall mass through metal loss.

Figure 5-24 – Accelerated Sample with Excessive Corrosion

The sample in Figure 5-24 was exposed to the same conditions as the previous sample, but for 10 hours duration. The sample suffered an increased amount of metal loss as well as some localised pitting to the surface. The results below (Figure 5-25) echo the results of the previous sample.

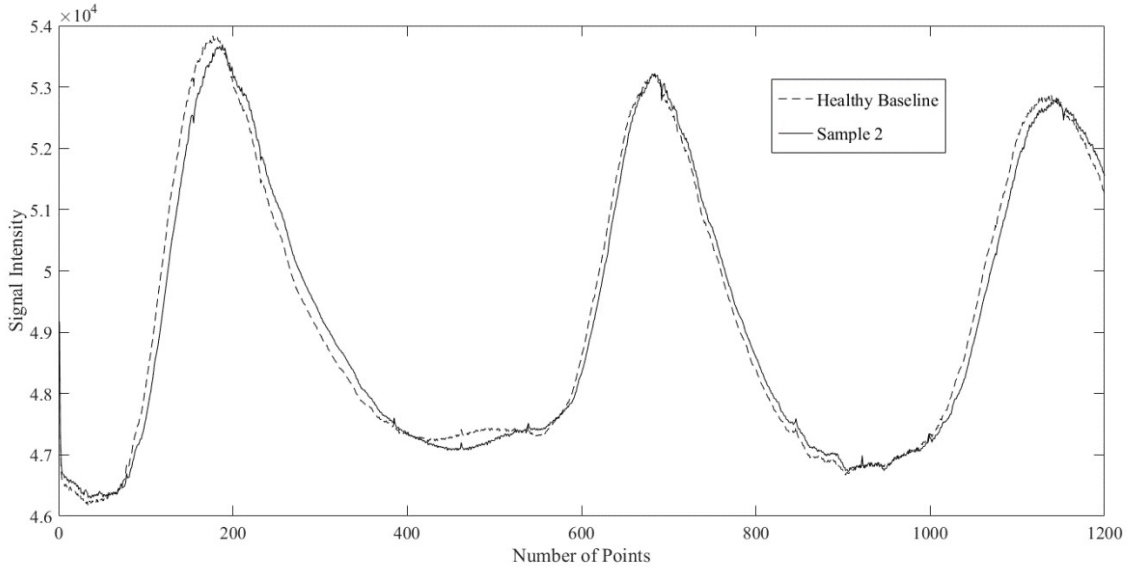


Figure 5-25 – Excessive Corrosion Results

The healthy condition material once again achieved higher reflection intensity. This sample had the rust layer removed, therefore removing the dielectric layer meaning that no permittivity effect was observed. This result therefore determines that the permeability change due to metal loss is negligible as the phase change is minor compared with the previous result.

In future it would be useful to age the same sample further to obtain results of corrosion progression. It seems that the change in sample could mask some of the resolution in the sensor measurement.

5.3.6 Painted Coating Results

The results of the painted coating experiment, shown in Figure 5-26, gave a clear indicator of the presence of corrosion beneath the painted coating.

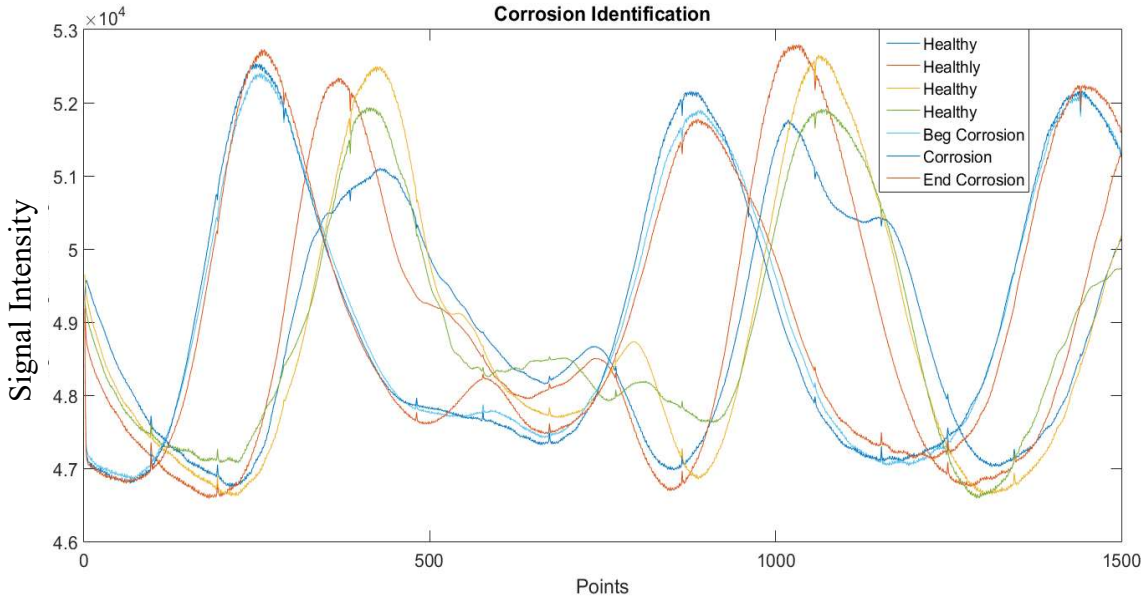


Figure 5-26 – Sensor Corrosion Identification

As can be seen from the results, a definitive phase change can be seen between the healthy signal responses and the corrosion area. The variations in the healthy data contain some phase shift and a large variation in amplitude. This can be accounted for due to the aluminium flakes in the paint as well as the poor coating application. Due to the nature of the application there is no way of determining the spread of the aluminium flakes throughout the sample. The varying thickness throughout the finish further attributes to the signal variations.

The area of corrosion provides the most stable results as it is over a much smaller surface area. The phase of all three signals associated with this area is almost constant as expected. The small variations in magnitude are caused as the sensor is overlapping both the corroded and healthy area of the sample. The area of corrosion identified by the sensor has been highlighted in Figure 5-27 and confirmed by AkzoNobel. The large change in signal to signify corrosion area is due to the permittivity change caused by the corrosion. The permittivity has changed the E field from the sensor.

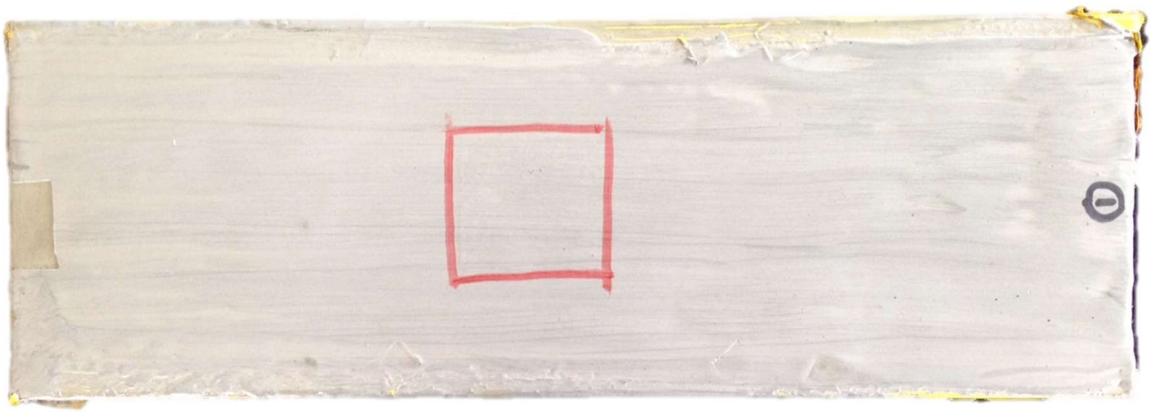


Figure 5-27 – Corrosion Location under Painted Coating

5.4 Sensitivity to Environmental Conditions

To determine the sensor reliability due to environmental conditions and the various temperature ranges of the measurement media, a simple experiment was conducted on a steel plate. The 150mm square plate was placed on a variable temperature hot plate in a fixed position. The sensor was the positioned and clamped 80mm from the surface. The temperature of the plate was set at 30 degrees to ensure it was above ambient conditions. Once temperature had stabilised two measurements were taken with the sensor and the surface temperature of the plate was recorded. The temperature was subsequently increased by 15 degrees centigrade after each measurement up to 160 C.

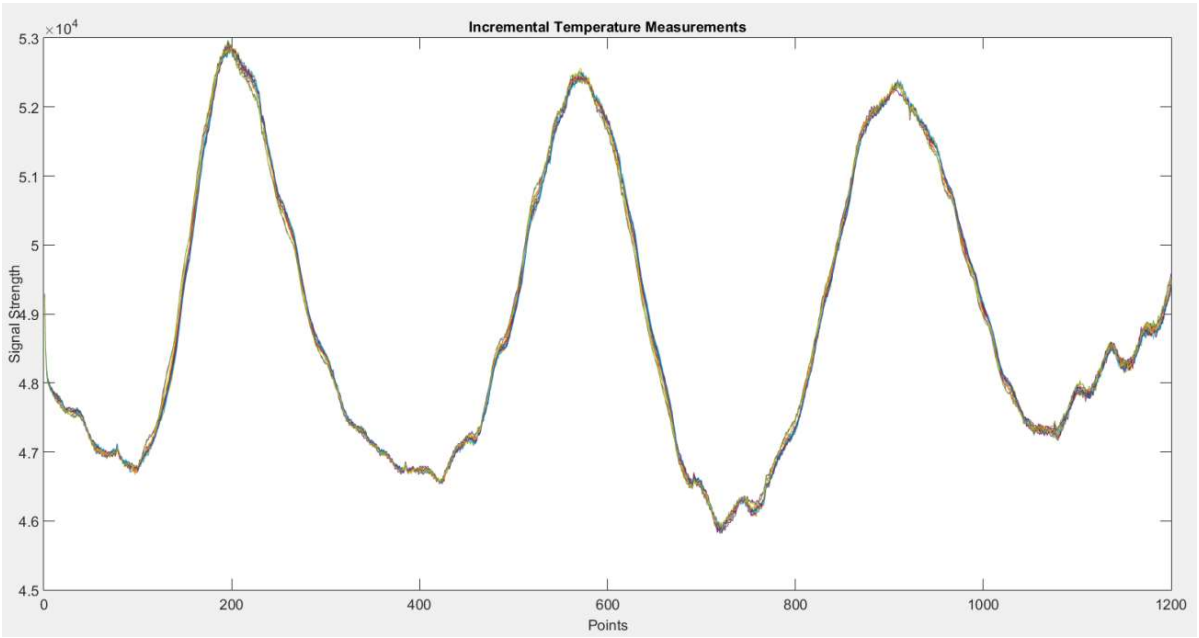


Figure 5-28 – Incremental Temperature Measurements

In order to show the sensor behaviour with temperature, the signal obtained was averaged for each temperature measurement resulting in a single value of the sensor

intensity. These values from these measurements were plotted against temperature as shown from Figure 5-29. There was very little change in the signal response due to the varying temperature. The same conclusion was drawn when directly comparing the results of the ambient measurement with the extreme temperature of 160C.

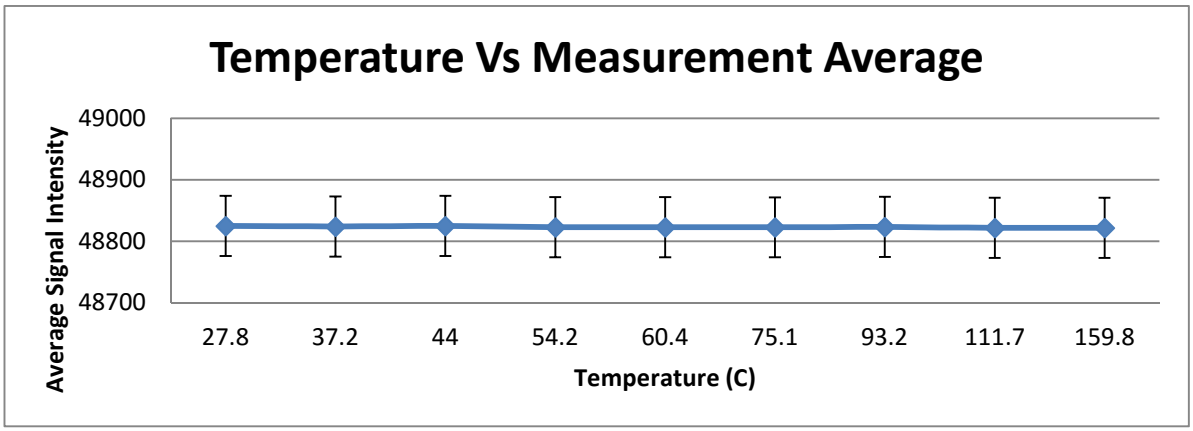


Figure 5-29 – Temperature vs Measurement Average

Figure 5-29 shows the data presented with error bars which are representative of a 0.1% error. This error was calculate in terms of the variance in the signal between 3 measurements of the same temperature.

5.5 Repeatability

As can be seen from the data shown in Figure 5-28, the data from the sensor is very repeatable, even in conditions where the asset surface has an increasing or drifting temperature measurement.

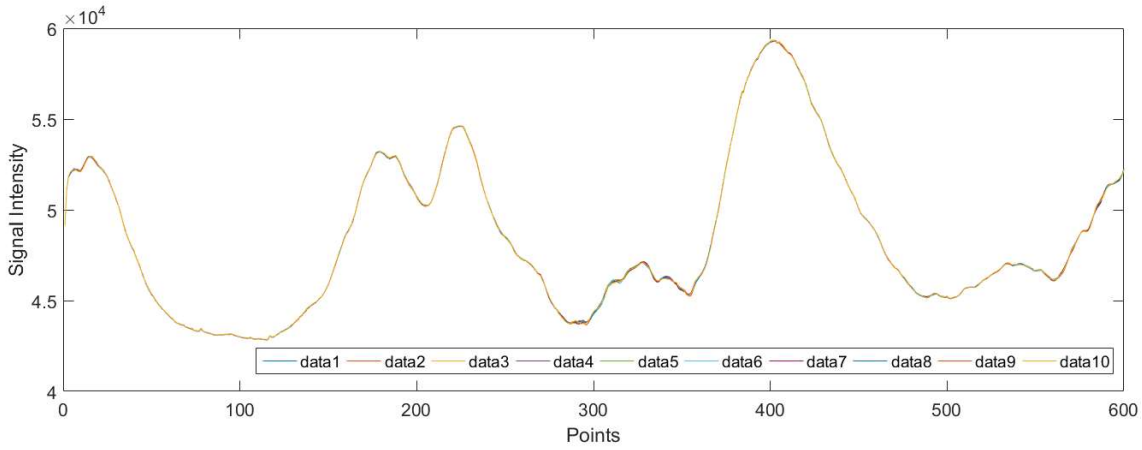


Figure 5-30 – Repeatability of Measurements

As shown in Figure 5-30, the data of 10 traces are imposed on top of one another with very little variance across the whole trace. Each traces was a single measurement taken at the point of sample at a fixed distance. The measurements were taken at 30 second intervals.

5.6 Validation of Non-Destructive Technique

The sensor was designed primarily as a new non-destructive method to determine the presence of corrosion under insulation. During use no change, whether it is physical or chemical, should occur to the monitored asset either temporarily or permanently.

The main concern while developing this sensor was the generation of heat during measurements. As the sensor uses microwave energy, it is a common misconception that heat is generated when an asset comes into contact with the waves. In order to disprove this, an infrared camera was used to monitor both the surface of the asset being monitored and the transmission end of the horn antenna. On both occasions no increase in ambient temperature could be seen by the infrared camera.

According to the supplied literature of the radar module, the maximum power from the unit is 400 mW. Power at this level is not deemed to be significant enough to cause any localised heating effect in the area of the measurement. It is considered that no other effects will be created by the sensor to cause temporary or lasting damage to the asset.

Chapter 6 – Prognostic Health Management

6.1 Condition Monitoring Overview

Condition monitoring (CM) is the function of monitoring the condition of a piece of machinery while in operation to assess the current health status in real-time. Active CM centres on the principle of monitoring specific parameters with known failure signatures. These failure signatures are based on parameters such as vibration, temperature, acoustic leakage etc. and will inform the user of upcoming issues with specific components.

There are several approaches taken when implementing CM. These range in cost and complexity to give a wide range of results in any application. The most cost effective approach is that of corrective maintenance. This maintenance program runs on the premise of replacing (or repairing) after the failure has taken place. This approach is extremely easy to implement and is commonly known as Run-To-Failure. Using this maintenance program requires no sophisticated monitoring system or decision making technology. It is often used on assets which have been in use for some time or items considered to have a relatively low risk (high reliability) of failure, i.e. considered safe when in a failure state.

The main criticism of this method of maintenance is the requirement to stop production for replacement or repair. Stopping the equipment for long periods of time can be due to a number of factors, many out with control of the facility, e.g. lead time for spare parts. This unknown period of downtime can be further increased due to lack of personnel to conduct the repair work or limited accessibility to the asset.

The second more informed CM approach is preventative maintenance. This approach has been developed to reduce the impact in industries where downtime is less tolerable. Preventative approaches such as replacing key components susceptible to failure on a regular basis reduce the risk of downtime. Stocking components with long lead times or high failure rates further reduces the downtime disruption. Strategies such as replacing high failure rate components during other downtime further reduce the likelihood of unplanned maintenance occurring. [67, 68]

This approach does risk over maintaining key components slightly increasing operational cost, but this does not require large capital expenditure to implement once a stock of components have been built upon.

The final, most comprehensive approach to CM is that of condition based maintenance (CBM). Condition based maintenance is most effective in reducing maintenance cost through reduction in over maintenance as well as maximising production through the reduction of downtime. Figure 6-1 shows a suggested system block diagram for the implementation of the CBM approach to CM. [69]

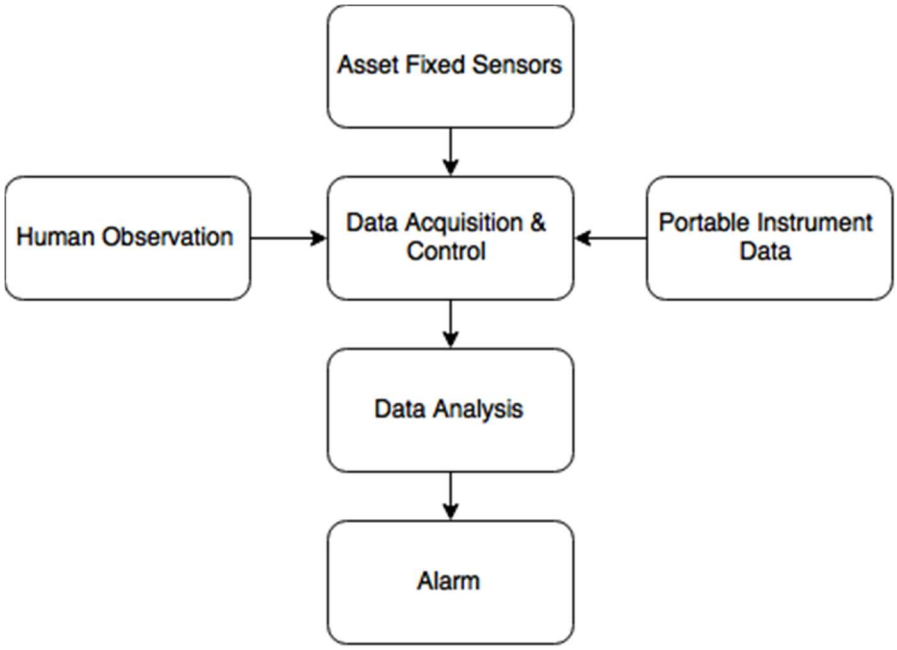


Figure 6-1 – Condition Monitoring System Design

The system uses sensor data "Asset Fixed Sensors" from the chosen monitored parameters at the selected key components, as well as the option of imported data from handheld sensors (Portable Instrument Data) to inform decisions on the asset health. Defects or performance irregularities from personnel maintenance assessments (Human Observation) can also manually be inputted to inform the health of the system. Data analysis can consist of various levels of complexity, from simple healthy baseline comparisons to form fault detection, or more complex signature and pattern recognition to inform trends toward failure. Which every type of data analysis which is performed, the user can be notified by alarm conditions to warn of impending failures.

6.2 PHM Summary

Prognostic Health Management (PHM) is a more comprehensive approach to improve and build on the monitoring technologies already in use in CBM programs. This approach can be considered as a total overview at not only individual components in real time, but the affect a failure of an individual component failure would have on the whole system health. The main aim of PHM is to give the user 'useful' information instead of mountains of data. [70, 71]

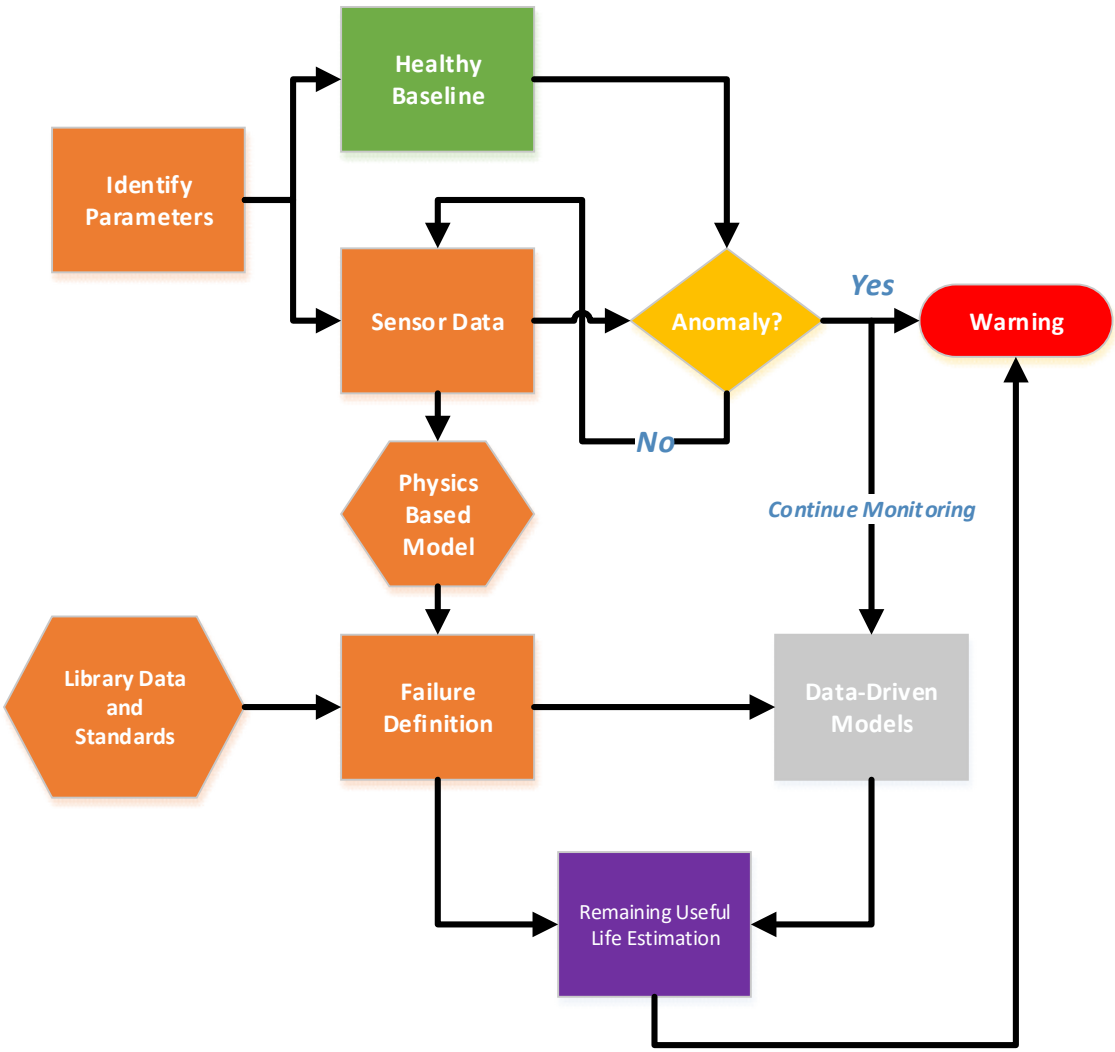


Figure 6-2 – Prognostic Health Management System Design

To make PHM a useful tool to aid management in optimal maintenance planning, it is suggested that a remaining useful life (RUL) of the asset should be available. This would give managers and personnel an ideal view of the overall system health, allowing a unique insight into the asset availability to maximise production. Increased awareness of the asset availability allows careful planning for repair so that replacement part and

personnel are available when required. Suggested PHM system design shown in Figure 6-2.

New sensor technology and data management techniques have emerged in recent years which allow for the deployment of multiple low cost sensors in one large data system to enable such an approach to be implemented. The increased sensing allow for more individual components of the overall system to be monitored. Advancements in sensing technology permits earlier fault detection due to increased sensitivity. Reduction in component cost increases the feasibility of redundant sensors, enabling the implementation of built-in self-test to guarantee sensor data. [72]

6.2.1 PHM Approaches

There are different approaches to be used when considering the implementation of a comprehensive monitoring system. To ensure the best implementation, it is normal to first of all utilise Failure Mode, Mechanism, and Effects Analysis (FMMEA) to determine potential faults and how they manifest themselves within the system. [73] Using this type of analysis allows the correct sensors to be determined in order to detect the fault at the earliest possible stage. The process also insures that the correct PHM approach is selected reducing the risk of over engineering the system.

The data driven approach is considered to be a quick and easy method to implement. This type of implementation does not require historic data to being monitoring, learn from scratch using either Bayesian or Neural Networks. Bayesian networks are made up of two separate networks, one of which is used to detect faults and second is for fault isolation. The detection network uses sensor data to check for erroneous values which are wildly different from proceeding measurements to determine whether or not a fault has developed. [74-76] Neural networks are useful to do initial analysis on big data often associated with condition monitoring applications. As the data is often depicting the healthy condition of asset, i.e. no fault, most of the data can be discounted. This means the data representing the healthy condition is only stored once as a reference and the other data stored will only relate to faults. This therefore results in efficient detection of faults within datasets. [77-79] The ideal data driven approach will combine both of these networks to ensure the best results as discussed by Chaochao et al. [80]

Model based approaches use mathematical data and algorithms to predict the propagation of a fault through a system, i.e. Physics of failure (PoF). Physics of failure models are power tool which take into account the load stress and environmental conditions of the asset to give a comprehensive understanding of the asset behaviour as it is operated under different conditions. This type of model can infer where it is possible to increase pressure on the asset without causing lasting damage to the asset. This model is expensive to implement as it requires a lot of time to understand each individual component within the system and how a change in this component will affect the asset as a whole. [81, 82]

	Advantages	Disadvantages
Data Driven	• Does not require detailed failure mechanism data. • Can be implemented quicker than other methods.	• Requires retraining if load/environmental conditions change. • Only as accurate as the data it receives • Extensive training and human intervention required to maintain accuracy • Large amount of historical data
Model Driven	• Can give an accurate prediction of failure for set failure mechanisms. • Comprehensive knowledge of investigated failures. • Method is relatively autonomous.	• Costly to implement on large complex systems. • Extensive knowledge of failure mechanism required. • Can only detect faults that system has investigated.
Fusion Approach	• Offers the advantages of both a model and data driven methods. • Can provide a more accurate RUL estimate.	• Can be very difficult and costly to implement on large complex systems

Table 6-1 – PHM Method Comparison

Table 6-1 summarises the advantages and disadvantage between the three main approaches to PHM. The method focussed on in this thesis is that of the fusion approach. This approach utilises both data and PoF as shown in Figure 6-2. This methodology provides accurate results through the sensor data and the real-time PoF informing the current health of the system. These benchmarks are compared with historical faults and failure definitions to search for data which signals early indication of faults developing. This data is compared with both models to estimate an accurate RUL of the system. [83, 84]

6.3 Comsol Model Implementation

As discussed in the previous section, the implementation of an accurate RUL estimate is based upon dependable sensor data to inform the current asset health, but more importantly the model data to infer the RUL of the asset. This section of the thesis

outlines the development of the model 'CorrosionSim,' which models the effect of corrosion on the external wall of an insulated pipe. As shown in Figure 6-3, the pipeline model consisted of three layers which could be defined by the user.

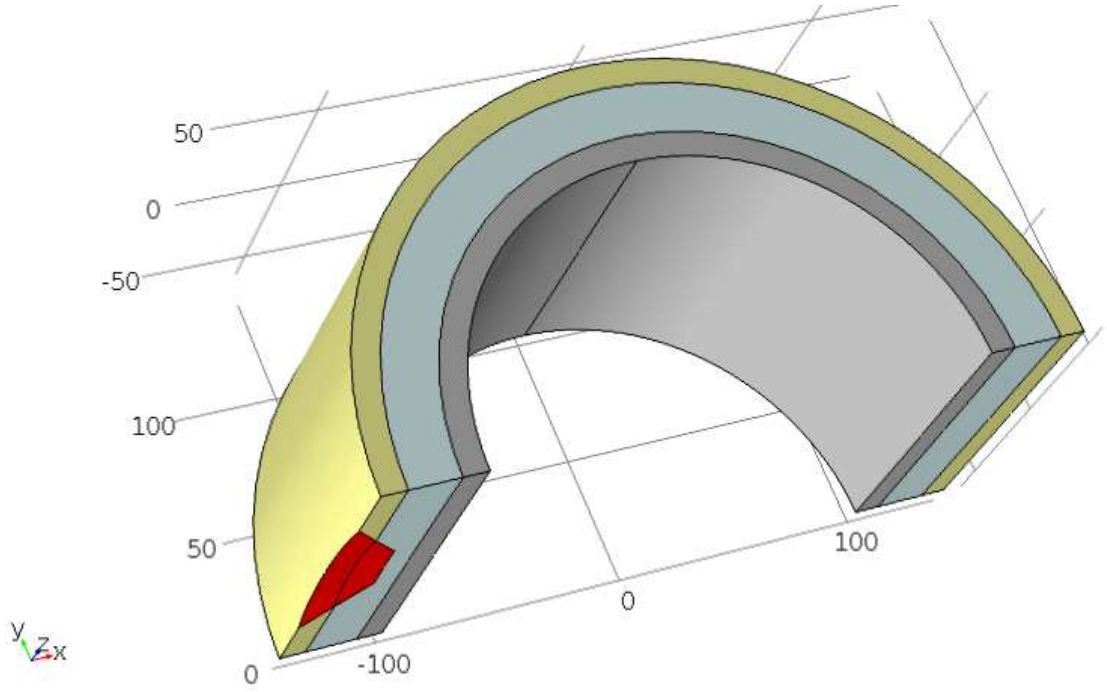


Figure 6-3 – Cross Section of Pipeline Model

The outer layer (yellow) was a High Density Polyethylene (HDPE) polymer and the second layer was a porous foam layer. The thickness of both layers could be defined in the model, but the properties of the porous foam layer could be defined further. The porous layer parameters could adjust the mean pore size and distribution as well as the overall porosity of the insulation as a percentage.

The metallic pipe was modelled as a Steel Alloy (ASTM A 106 grade B). This was a widely used grade of steel for pipelines with a good amount of documentation to inform the model parameters.

The model was developed with the assistance of Continuum Blue Ltd who specialise in developing Comsol based models. The interface of the model is shown in Figure 6-4.

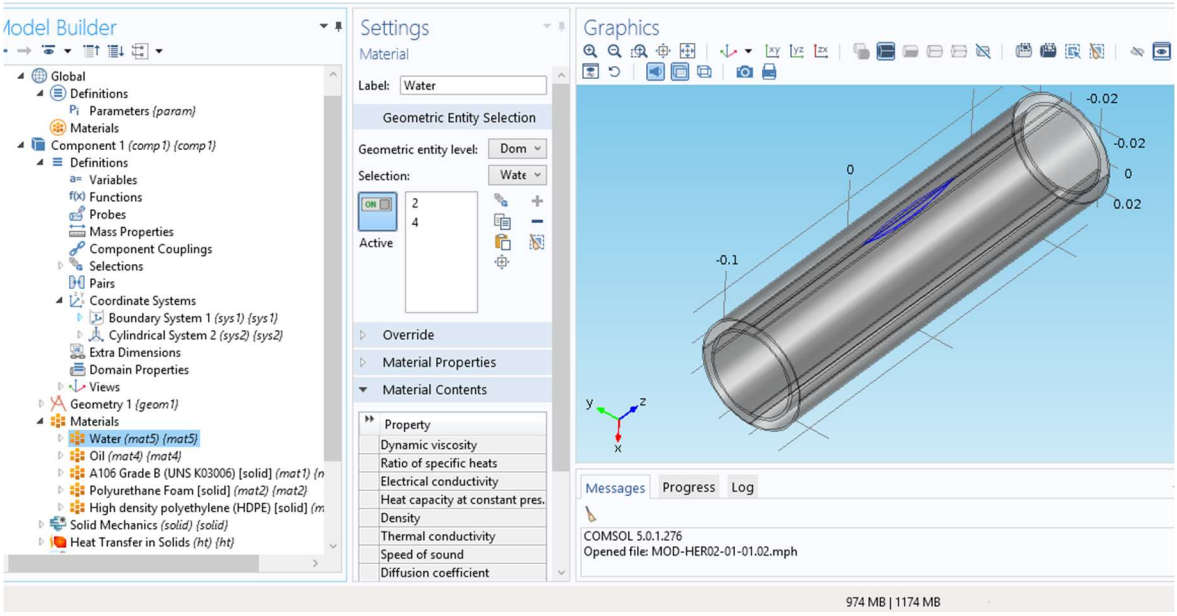


Figure 6-4 – Screenshot of 'CorrosionSim' Model

CorrosionSim utilises several coupled physics modules from the Comsol software suite to model the corrosion of the pipeline overtime, where the pipelines dual insulation layer has a defect, or crack, allowing for moisture and salts to corrode the pipeline itself.

The crack was another user defined parameter for the model. The length, width and height of the defect could be defined down to a minimum of 0.5mm. At this time it was deemed sufficient to model at this resolution, although improvements can be made in the future.

The user could define the pressure and temperature of the external environment surrounding the pipe as well as the pressure and temperature inside the pipe. Although the model did not take into account the corrosion internally due to production fluid flow, it was important to know what the internal condition were as the temperature of the inside of the pipe has a heating effect on the outside. As discussed in section 2.2.2 temperature cycling causes condensation to form under insulation, which can lead to the onset of corrosion or act as a catalyst to corrosion already in process. The internal and external pressure values were important when looking at the structural effect on the pipe as outlined in the next section.

6.3.1 Model Physics

Several modules of the Comsol Multiphysics software were used to model the CorrosionSim accurately. The most widely used was the 'electrochemistry' module. The physics from this module looked at the chemical species transport, fluid flow, heat

transfer and corrosion interfaces. The combination of all of these studies resulted in determining the potential in the electrolyte (moisture) as well as the corroding effect of the electrolyte on the metallic pipe surface due to leakage of chlorides etc. from the insulation layers.

The 'structural' module was used to determine the physical effects on the asset. Typically stress and strain were the main focus using studies such as Young's modulus, Yield Stress and Poisson's ratio. Young's modulus defines the 'stiffness' of a solid material which is the ratio of stress to strain with units of N/m^2. [85] Yield Stress is a material property which defines the stress at which a material, in this case steel, begins to deform plastically. [86] Plastic deformation is used to define when a material has deformed so much that it will not recover to its natural state. Poisson's ratio is a measure of the Poisson effect. The Poisson effect is the phenomenon which captures that when a material is compressed, it expands in two directions perpendicular to the compression direction. The same can be said when a material is stretched. The material will contract perpendicularly to the direction of stretching. Poisson's ratio is the percentage of expansion divided by the percentage of compression. [87] The density function

Finally the 'thermal' module looked specifically at the heat transfer from the internal pipe as well as the environmental temperature and humidity. These were considered in the corrosion estimation. Specific studies used were Specific Heat Capacity, Thermal Conductivity and Thermal Expansion Coefficient.

6.4 Remaining Useful Life Estimation Example

Using the CorrosionSim application it is possible to understand how the sensor could be used with a PoF model to determine an accurate RUL of an asset. The model uses several parameters to accurately determine the level of corrosion on the outer steel surface of the pipe. Using the sensor data to feed into the model the current health state of the asset, it would be possible to calculate the RUL of the asset under current working load and conditions.

For this example the specification of the pipeline was taken from an incident which occurred in March 2013, when an ExxonMobil pipeline ruptured and caused 200,000 – 420,000 gallons of heavy oil to be released. [88] Although not directly attributed to

CUI, the nature of the sudden spill fits with the nature of pipeline failures which are caused by CUI. The pipeline diameter was 20 inches (510 mm) and the wall thickness of the pipe is $\frac{5}{16}$ inch (7.9 mm). The pipeline at the time of failure was operating at 708 psi, 14% below the maximum 820 psi rating. The pipeline was insulated with 2 inches of thermal insulation (50.8 mm) and a 1 inch external coating (25.4 mm). The temperature of the product in the pipeline is considered to be 120ºC so that the product can flow. These parameters were inputted into the model with crack geometry of 350 mm length, 2 mm width and 78 mm deep to breach the insulation layers. The results from the model are shown in Figure 6-5.

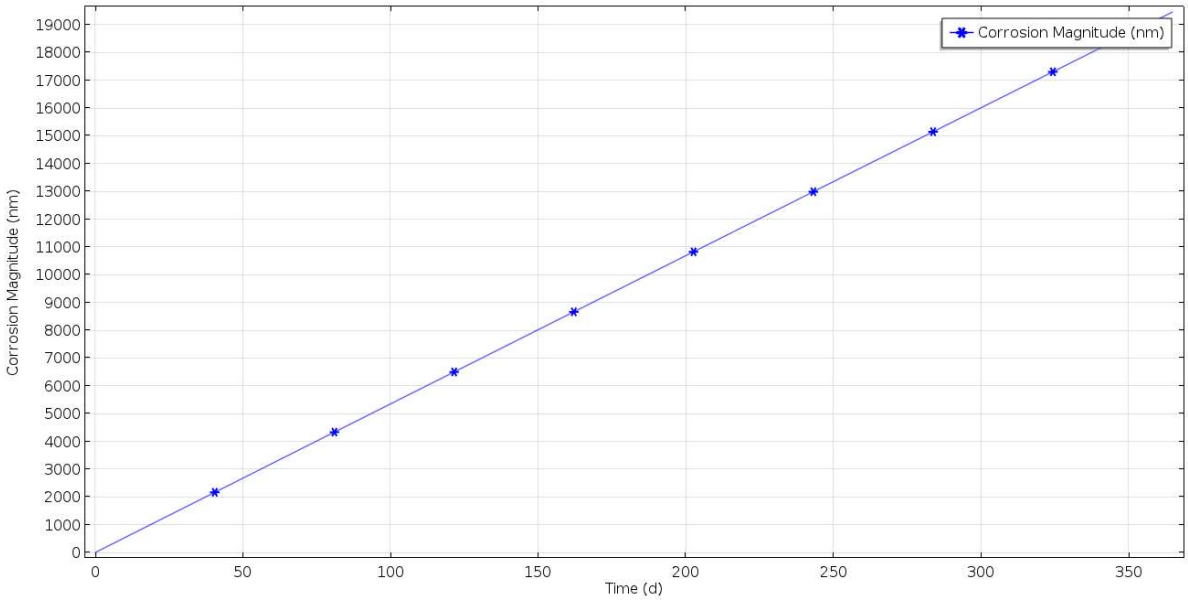


Figure 6-5 – Corrosion Magnitude Vs Time

As shown the corrosion occurs at a rate of 19,000 nm per year. This means that year on year if left untouched, the wall thickness of the asset will continue to corrode at 19,000 nm per year. Using the same parameter, the minimum wall thickness can be calculated using the ASME standard B31.4m, equation **6-1**.

$$t = \frac{Pd_0}{2(FES_y)}$$

6-1

Where

t = minimum wall thickness (inch)

P = Internal pressure (psi)

d_0 = Outer diameter of pipe wall (inch)

$\mathbf{S_y}$ = minimum yield strength for pipe (psi)

F = derating factor (0.72 for all locations)

E = longitudinal weld-joint factor

The value of **E** is assumed to be 1 as the section of pipe has no welds and $\mathbf{S_y}$ for carbon steel Grade 2 is 35000 psi. [89] Thus,

$$t = \frac{708\ x\ 20}{2(0.72\ X\ 1\ X\ 35000)}$$

$$t = 0.28\ inches$$

$$t = 7.11\ mm$$

According to the calculation, the minimum wall thickness for the pipeline under the conditions outlined is 7.11 mm. As the wall thickness of the pipeline was 7.9 mm as declared earlier, there is 0.79 mm of extra wall material. The corrosion per year value is calculated using the Comsol model and in this case is 0.019mm. Bearing this in mind it is possible to determine how many years of corrosion can be tolerated before the asset becomes unsafe and therefore determines a RUL using equation **6-2**.

$$RUL\ (years) = \frac{(Wall\ Thickness - Minimum\ Design\ Thickness)}{Corrosion\ Per\ Year} \quad \textbf{6-2}$$

$$RUl\ (years) = \frac{(7.9 - 7.11)}{0.019}$$

$$RUL\ (years) = 42\ years$$

Although the estimate above seems to be very high, this would be the design lifetime of the asset from the point of installation. Bearing in mind that this is based only on corrosion from the external wall thickness, this can be seen as rather realistic. If the same rate of corrosion occurred on the inside of the pipe due to fluid flow etc. then this RUL would be further reduced, bring the estimate toward the actual asset life cycle of 42 years.

Looking at this equation again using the yield strength ($\mathbf{S_y}$) for Carbon Steel Grade 3 we can see that the RUL off the asset increased due to the reduction in the minimum wall thickness required due to the increase in material strength.

$$t = \frac{708\ x\ 20}{2(0.72\ X\ 1\ X\ 45000)}$$

$$t = 0.22\ inches$$

$$t = 5.59\ mm$$

The minimum wall thickness is now 5.59 mm resulting in a significantly higher RUL for the asset of 122 years as shown below:

$$RUL\ (years) = \frac{(Wall\ Thickness - Minimum\ Desig\quad Thickness)}{Corrosion\ Per\ Year}$$

$$RUl\ (years) = \frac{(7.9 - 5.59)}{0.019}$$

$$RUL\ (years) = 122\ years$$

The sensor data could be used in two ways to inform the model. Firstly, the sensor data could be used to detail the current surface corrosion level on the asset. The model would then be used to determine how the corrosion would propagate within the pipe. If the corrosion were to further affect the wall thickness of the pipe, then this would significantly reduce the RUL of the asset as the wall thickness is the critical measurement when considering safety and availability of the asset.

Secondly the sensor data could be used to determine the geometry of the crack on the external protective layer of the pipe. This would not require any additional changes to the model and could be used as a preliminary tool to determine whether the corrosion would cause an issue with the asset integrity during the life cycle. If it is considered from the model that RUL is longer than then life cycle then maintenance could be reduced or bypassed when deemed not required.

The model still requires further user defined parameters to be included to make this a viable RUL estimation, but simply the addition of a parameter to define the current health state (surface corrosion level) would increase the accuracy of the estimation. Further parameters such as fluid flow could be added to model the corrosion effect on the inside of the pipe which also reduce overall wall thickness. Harmful gases such as Hydrogen Sulphide (H_2S) are naturally produce during oil and gas production and contribute to the corrosion of steel. [90, 91] Modelling such gases and their effects on steel would further increase the models accuracy in this industry.

Chapter 7 - Conclusions

7.1 Concluding Remarks

This thesis has outlined a novel sensor system capable of monitoring corrosion under non-metallic insulation layers. The concept is covered under a general Microwave Sensing Patent [92] as outlined in the author publication and a standalone UK patent is under review for the specific concept of CUI monitoring. The patentability of the designed sensor proves the novelty of the system and the commercial potential for such a device. Several petrochemical companies have declared interest in the designed sensor, increasing the likelihood that work outlined in the following section will be complete to take the system to market.

The thesis structure was written in such a way as to step through the design process to experimental validation. Each chapter provided an overview of the development process as follows.

Chapter 1 provided the motivation for the development of the sensor. Outlining the costs associated with CUI and industrial need for such a sensor.

Chapter 2 gave an overview of the current state of the art technology in the sector. The sensor was compared with the other techniques within this chapter, showing that the sensor was a time saving, flexible device which could be employed in a variety of situations in comparison to the other NDE techniques on offer.

Chapter 3 outlined the key horn antenna principles which were used in the design of the horn antenna. Relevant electromagnetism theory was stated along with the principles of the FMCW hardware used in the sensor. A brief overview of the hardware operation was given for the readers understanding of the experimental results.

Chapter 4 concentrated on the design of overall system. Firstly the control system hardware is outlined, defining the design of all the relevant components. The control hardware is described as well as the electronic components of the sensor. The section also included the system software for both the control and GUI. The design of each piece of software and the functions and reasons for their implementation were outlined along with the relevant signal processing features built into the software. Finally the

design of the sensor head (horn antenna) and the implementation of the manufacturing process were outlined.

Chapter 5 stepped through each of the experiments conducted to explaining the reason for each setup. The experiments were designed to simulate the various layers of an insulated pipeline and experiments were conducted with varying layer depths and configuration to test the sensor ability. The experimental work has demonstrated the success of the sensor concept to justify further research work in this area.

Chapter 6 summarised the concept of Condition monitoring as well as the Prognostic Health Management technique. This provided the reader with the background knowledge required to understand the concept of smart asset management for effective monitoring. Using this theory, the chapter went on to detail the developed Comsol model and its purpose in estimating RUL. Finally the chapter concluded with a case study example of RUL estimation to show how the model and sensor could be used in conjunction to determine RUL of pipelines using data from the sensor.

7.2 Key Points

Throughout this thesis the aim has been to outline the design process of developing the proposed sensor system. The key points to be summarised from this thesis are as follows:

- The sensor can determine surface defects within metallic surface with or without a dielectric insulation layer

- The sensor cannot measure defects through metallic (tin) cladding

- It is possible to determine physical changes in the insulation layer

- The sensor can sense varying degrees of corrosion beneath insulation layers

- Water ingress can be detected in insulation layer as well as the concentration

- The sensor can determine the presence of corrosion beneath painted coatings

7.3 Future Work

A multitude of future work has been identified during this research project which could not be addressed within limits of the PhD research project. Time, expertise and money were the main limiting factors due to the limited budget and set time frame of the programme of study. The future work can be split into two sections, additional research work and product to market work.

7.3.1 Additional Research Work

Additional research work which was not completed within the elected period of study consists of the following:

- Enhanced defect detection
- Enhanced Model implementation
- Precision Sensor Head Design
- Polymer Ageing

7.3.1.1 Enhanced Defect Detection

The main priority for future work should be the enhancement of defect detection. As discussed in section 4.3 the defect detection process worked on the concept of comparison with historical (library) data. Although this data driven approach is acceptable and provided valid results, it did not look at individual features within the signals. The experimental chapter has outlined many of the signal features for each of the different sample types, but these now need to be implemented in software for automated defect classification.

7.3.1.2 Enhanced Model Implementation

The Comsol model outlined in section 6.3 provided reasonable results from a research point of view, but would not be accurate enough for use in RUL estimation for working assets.

The model requires more parameters to be modelled to increase the accuracy of the corrosion estimation. By adding a parameter to define the current level of corrosion

determined by the sensor, it would be possible to determine the RUL of the asset from the current time positon rather than day zero as it currently stands.

The addition of internal corrosion due to fluid flow would be an advantage as this would allow an assessment of the overall corrosion which determines wall thickness. As it stand it is assumed that there is no corrosion to the internal pipe meaning that RUL is over estimated. The RUL example given in section 6.4 suggested the RUL was 84 years, when in actual fact the asset ruptured after 65 years. If the internal corrosion had been taken into account, the RUL would have tended closer to that figure, giving a fairer assessment of the asset.

7.3.1.3 Precision Sensor Head Design

The horn antenna designed and used throughout the thesis for the experimental work provided enough resolution to measure the large defects on the copper sheet as well as the corrosion samples under insulation layers. It is suggested that another horn antenna should be designed with the addition of a lens to focus the beam at a more concentrated angle allowing for much smaller defects to be detected. The suggested device is shown in Figure 7-1.

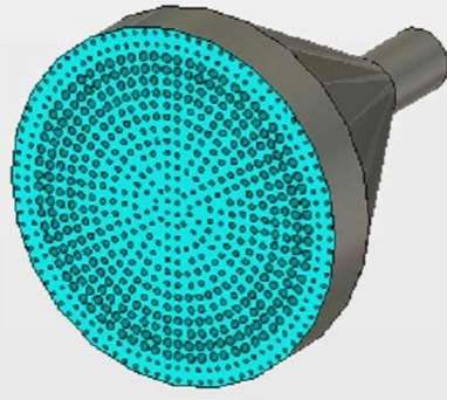


Figure 7-1 – Conical Horn and Dielectric Lens

Changing the horn design to a conical horn will allow for the addition of a dielectric lens to focus the beam. The beam will increase the directivity of the horn focusing the waves to a smaller degree angle as shown in Figure 7-2.

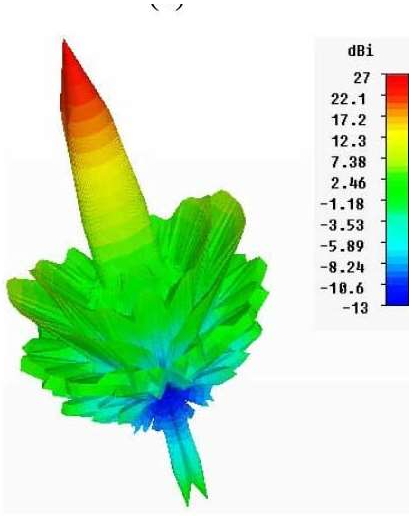


Figure 7-2 – Radiation Pattern from Conical Horn with Lens

7.3.1.4 Polymer Ageing

To determine the feasibility of using the sensor to detect ageing within polymer insulation further experimental work is required to confirm the changes which can be detected by the sensor. The experimental protocol of this experiment also needs to be increased as the same sample must be used from day 1 or ageing through to the end. The ageing parameters must be the same and the position on the stage must be fixed.

Ideally the sensor would be mounted within the oven and monitor the sample as it aged, limiting the interference from external factors.

7.3.2 Product to Market

In the future, further development of the sensor will be required to realise the potential of bringing a fully functional handheld device to market as shown in Figure 7-3.

Of primary focus, should be the method in which to monitor the distance of the horn end to the measurement surface. This will be a critical component of the device when deployed in the field as it will ensure that repeated measurements of an area are taken at the same distance as previously. Changing the distance of the sensor from the measurement surface risks falsely acknowledging an increase in corrosion or indeed, conversely falsely acknowledging the device as being healthy. There is also the potential of using the measured distance between asset and sensor to automatically correct the corrosion measurement, meaning that the corrosion measurement would be scaled depending on the offset distance. This in itself would require an algorithm to

determine the scaling factor for the distance measurement as it is most likely not a linear response.

Handheld Sensor Sensor Array Collar

Figure 7-3 – Proposed Sensors

Further to the handheld device, a propose sensor collar could be developed to allow automated testing of large lengths of pipe. Placing multiple sensors around the collar would allow for an assessment of the asset condition around the full 360^0 of the pipe. The collar could then be motorised to travel along the length of the pipe using GPS data to log areas of suspected corrosion.

7.4 Novelty

The new technique for determining CUI outlined within this thesis can be considered novel due to its multiple corrosion detection parameters. Unlike other techniques review in the literature survey in section 2.2, this method can determine various defects or precursors to identify the presence or future presence of corrosion.

Similar to other devices mentioned this device can determine water within insulation through the outer protective layer. Unlike other techniques, it can categorise the quantity of moisture present to as low as 6% concentration level as described in the experimental section 5.3.3.

The same device can also determine the presence of surface corrosion on the external pipe surface as shown in section 5.3.5 and 5.3.6. This can even be under a painted anti-corrosion coating. Being able to quantify levels of surface corrosion which can then be

fed into a RUL model as discussed in 6.4 provides valuable information concerning the assets remaining availability.

Lastly, although only an initial assessment, the device has demonstrated its potential in determining the condition of the outer polymer layer. This therefore could detect an issue with the external layer before any water penetration, allowing for replacement or mechanical repair before the onset of corrosion.

To summarise, the method outlined within the thesis is a multi-parameter corrosion detector which requires no physical hardware changes to switch between measurements.

Appendix A - Arduino Firmware Main Code Listing

```
#include "pins.h"
#include "baseobj.h"
#include "controller.h"
#include "message.h"
#include "driver.h"
#include "laser.h"

int incomingByte = 0;   // for incoming serial data
message msg;

driver x(x_dir, x_step, x_min, x_max, true);
driver y(y_dir, y_step, y_min, y_max, true);
driver z(z_dir, z_step, z_min, z_max, true);
laser  l(laser_pin);

controller con;

void setup() {
	Serial.begin(9600);     // opens serial port, sets data rate to 9600 bps

	while(!x.home() || !y.home() || !z.home()) {
		delayMicroseconds(step_delay);
	};
	unsigned int rx = con.addObject(x);
	unsigned int ry = con.addObject(y);
	unsigned int rz = con.addObject(z);
	unsigned int rl = con.addObject(l);

	con.addTrigger(rx, 'x');
	con.addTrigger(ry, 'y');
	con.addTrigger(rz, 'z');
	con.addTrigger(rl, 'l');

	con.addTimer(rx, step_delay);
	con.addTimer(ry, step_delay);
	con.addTimer(rz, step_delay);
}
void loop() {
	if (Serial.available() > 0) {
		if(msg.addChar(Serial.read())) {
			con.fireTrigger(msg.getTarget(), msg.getMessageLen(),
msg.getMessage());
		}
	}
	con.fireTimer();
}
```

Appendix B - Arduino 'Controller' Sub Code

```
#include "Arduino.h"
#define MAX_OBJS 20

class controller {
	private:
		baseobj*	objects[MAX_OBJS];
		unsigned int	refrence[MAX_OBJS];
		bool		timeenable[MAX_OBJS];
		unsigned long	timestep[MAX_OBJS];
		unsigned long	timerun[MAX_OBJS];
		unsigned long	lasttime;
		unsigned int	count;
	public:
		controller(){
			//lasttime = millis();
			count = 0;
			for(int i =0; i < MAX_OBJS; i++) {timeenable[i] = false;}
		}
		unsigned int addObject(baseobj &instance) {
			objects[count] = &instance;
			return count++;
		}
		void addTrigger(unsigned int index, unsigned int ref) {
			if(index < count)
				refrence[index] = ref;
		}
		void addTimer(unsigned int index, unsigned int delay) {
			if(index < count) {
				timestep[index] = delay;
		      timeenable[index] = true;
		      timerun[index] = micros();
		}
		}
		void fireTrigger(unsigned int ref, unsigned int length, unsigned int
&array) {
			for(int i = 0; i < count; i++) {
				if(refrence[i] == ref)
					objects[i]->message(length, &array);
			}
		}
	   void fireTimer() {
			for(int i = 0; i < count; i++) {
				if(timeenable[i]){
					if(micros() > timerun[i] + timestep[i]) {
						objects[i]->timer();
						timerun[i] = micros();
					}
				}
			}
		}
};
```

Appendix C - Arduino 'Driver' Sub Code

```
#include <Arduino.h>

class driver : public baseobj {
        private:
                unsigned long posit;
                unsigned long target;
                unsigned int  spd;

                int           p_dir;
                int           p_step;
                int           p_max;
                int           p_min;
                bool          direction;

        public:
                driver(int dir, int step, int min, int max, bool invert) {
                        p_dir = dir;
                        p_max = max;
                        p_min = min;
                        p_step = step;
                        direction = invert;

                        pinMode(dir, OUTPUT);
                        pinMode(step, OUTPUT);
                        pinMode(min, INPUT);
                        pinMode(min, INPUT);
                }

                void message(unsigned int length, unsigned int array[]) {
                        switch(length) {
                                case 0:
                                        return;
                                break;
                                case 1:
```

```
                        setTarget(array[0], true);
                    break;
                    case 2:
                        long temp = array[1];
                  temp = (array[0] == '-') ? -temp:temp;
Serial.println(temp);
                        setTarget(temp, false);
                    break;
                }
            }

            void timer() {
                if(digitalRead(p_step)) {
                    digitalWrite(p_step, LOW);
                } else {
                    step();
                }
            }

            void step() {
                if((digitalRead(p_max) && target > posit) || (digitalRead(p_min)
&& target < posit))
                    target = posit;
                if (posit == target)
                    return;
                if(posit > target) {
                    digitalWrite(p_dir, direction);
                    posit--;
                } else {
                    digitalWrite(p_dir, !direction);
                    posit++;
                }
                digitalWrite(p_step, HIGH);
            }
```

```
        boolean home() {
                digitalWrite(p_dir,direction);
                if(digitalRead(p_min)) {
                        posit = 0;
                        return true;
                }
                digitalWrite(p_step, !(digitalRead(p_step)));
                return false;
        }

        void setTarget(long value, bool absolute) {
                if(!absolute)
                        target = posit + value;
                else
                        if(value >= 0)
                                target = value;
        }

        unsigned long getPos() {
                return posit;
        }

        void setSpeed(unsigned int speed) {
                spd = speed;
        }
};
```

Appendix D - Arduino 'Laser' Sub Code

```
#include <Arduino.h>

class laser : public baseobj {
	private:
		int		p_laser;
	public:
		laser(int pin) {
			p_laser = pin;
			pinMode(p_laser, OUTPUT);
		}

		void message(unsigned int length, unsigned int array[]) {
			switch(length) {
				case 0:
					return;
				break;
				case 1:
					set(array[0]);
				break;
			}
		}
		void set(unsigned int value) {
			(value == 1) ? on() : off();
		}

		void on() {
			digitalWrite(p_laser, HIGH);
		}

		void off() {
			digitalWrite(p_laser, LOW);
		}

		void pulse(unsigned int count, unsigned long duration) {
			while(count-- > 0) {
				on();
				delay(duration);
				off();
				delay(duration);
			}
		}
};
```

Appendix E - Arduino 'Message' Sub Code

```
class message
{
	private:
		unsigned char  data[20];
		unsigned int    arguments[10];
		unsigned int    length, param;

		void addParam(unsigned int start, unsigned int finish) {
		bool num = false;
		long total = 0;
		for(int i = start; i <= finish; i++) {
				if(data[i] < 48 || data[i] > 57) {
			if(num) {
			 arguments[param++] = total;
			 num = false;
			}
			arguments[param++] = data[i];
			   } else {
			num = true;
				total = (total * 10) + (data[i] - 48);
			   }
			}
		if(num) {
		 arguments[param++] = total;
		 num = false;
		}
		}

	public:
		message() {
			length = 0;
		}

		bool addChar(unsigned char chr) {
```

```
                data[length++] = chr;

                if (chr == ';') {
                        param = 0;
                        unsigned int last = 0;
                        for(unsigned int i = 0; i < length; i++) {
                                if(data[i] == ':') {
                                        addParam(last, i-1);
                                        last = i + 1;
                                }
                        }
                addParam(last,length-2);
                        length = 0;
                        return true;
                }
                return false;
        }

        unsigned int getTarget() {
                return arguments[0];
        }

        unsigned int& getMessage() {
                return arguments[1];
        }

        unsigned int getMessageLen() {
                return param - 1;
        }
};
```

Appendix F - Arduino Pin Declarations

```
#define x_step 3
#define x_dir  2
#define x_min  A5
#define x_max  A4

#define z_step 10
#define z_dir  9
#define z_min  A1
#define z_max  A0

#define y_step 7
#define y_dir  6
#define y_min  A2
#define y_max  A3

#define step_delay  700

#define laser_pin  12
```

Appendix G - Full Matlab Code Listing

```
function varargout = gui2(varargin)
% GUI2 MATLAB code for gui2.fig
%      GUI2, by itself, creates a new GUI2 or raises the existing
%      singleton*.
%
%      H = GUI2 returns the handle to a new GUI2 or the handle to
%      the existing singleton*.
%
%      GUI2('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI2.M with the given input arguments.
%
%      GUI2('Property','Value',...) creates a new GUI2 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before gui2_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to gui2_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".

% See also: GUIDE, GUIDATA, GUIHANDLES
 % Edit the above text to modify the response to help gui2
 % Last Modified by GUIDE v2.5 29-Oct-2013 17:14:04
 % Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @gui2_OpeningFcn, ...
                   'gui_OutputFcn',  @gui2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin[38]);
else
    gui_mainfcn(gui_State, varargin[38]);
end
% End initialization code - DO NOT EDIT
 % --- Executes just before gui2 is made visible.
function gui2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
```

```
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui2 (see VARARGIN)
% Choose default command line output for gui2
handles.output = hObject;

handles.A = [];
handles.B = [];
handles.C = {};
handles.D = [];

handles.xylist = [];

% Update handles structure
tabSetup(handles)
guidata(hObject, handles);

% UIWAIT makes gui2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in cmd_limit_auto.
function cmd_limit_auto_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_limit_auto (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_end('x', 'y');

table = get(handles.table_limit, 'Data');
table{1,1} = 0;
table{2,1} = str2num(handles.stage.query('x'));
table{1,2} = 0;
table{2,2} = str2num(handles.stage.query('y'));
```

```
set(handles.sld_jog_x, 'Max', table{2,1}, 'SliderStep', [(1/(table{2,1} - 1)), 0.01]);
set(handles.sld_jog_y, 'Max', table{2,2}, 'SliderStep', [(1/(table{2,2} - 1)), 0.01]);

set(handles.table_limit, 'Data', table);

handles.stage.block_home('x', 'y');
handles.stage.pleaseSendXandYToTheCenterPlease()
guidata(hObject, handles);

% --- Executes on selection change in menu_serial_list.
function menu_serial_list_Callback(hObject, eventdata, handles)
% hObject    handle to menu_serial_list (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns menu_serial_list contents as
cell array
%        contents{get(hObject,'Value')} returns selected item from menu_serial_list
 % --- Executes during object creation, after setting all properties.
function menu_serial_list_CreateFcn(hObject, eventdata, handles)
% hObject    handle to menu_serial_list (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

serialPorts = instrhwinfo('serial');
set(hObject, 'string', serialPorts.SerialPorts);

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cmd_serial_radar.
function cmd_serial_radar_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_serial_radar (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if(isfield(handles, 'fmcw'))
  delete(handles.fmcw);
end

val = get(handles.menu_serial_list, 'Value');
```

```
list = get(handles.menu_serial_list, 'String');
selected = list{val};

handles.fmcw = serial(selected,'BaudRate', 115200, 'Terminator', 'CR', 'Timeout', 0.1,
'InputBufferSize', 1024);
fopen(handles.fmcw);

fprintf( handles.fmcw,'HARDware:SYSTem RS3400K');
reply = fscanf(handles.fmcw);
fprintf(1,'%s',reply);

fprintf(handles.fmcw,'FREQUENCY:START 24.00e9'); pause(0.1);
reply = fscanf(handles.fmcw);

fprintf(handles.fmcw,'FREQUENCY:STOP 25.5e9'); pause(0.1);
reply = fscanf(handles.fmcw);

fprintf(handles.fmcw,'FREQUENCY:POINTS 1200'); pause(0.1); %1200
reply = fscanf(handles.fmcw);

fprintf(handles.fmcw,'SWEEP:TIME 0.3'); %0.3
reply = fscanf(handles.fmcw);

fprintf(handles.fmcw,'INIT');
reply = fscanf(handles.fmcw);
fprintf(1,'%s',reply);

fprintf(handles.fmcw,'SWEEP:MEASURE ON');
reply = fscanf(handles.fmcw);

fprintf(handles.fmcw, 'SWEEP:NUMBERS 1');
reply = fscanf(handles.fmcw);

fprintf(handles.fmcw, 'TRIGGER:SOURCE IMMEDIATE');
reply = fscanf(handles.fmcw);

guidata(hObject, handles);

% --- Executes on button press in cmd_serial_clear.
function cmd_serial_clear_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_serial_clear (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
delete(instrfind);
```

```
% --- Executes on button press in cmd_serial_stage.
function cmd_serial_stage_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_serial_stage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
if(isfield(handles, 'stage'))
  delete(handles.stage);
end

val = get(handles.menu_serial_list, 'Value');
list = get(handles.menu_serial_list, 'String');
selected = list{val};

handles.stage = Stagecontroller(selected);
guidata(hObject, handles);

% --- Executes when entered data in editable cell(s) in table_limit.
function table_limit_CellEditCallback(hObject, eventdata, handles)
% hObject    handle to table_limit (see GCBO)

% eventdata  structure with the following fields (see UITABLE)
%   Indices: row and column indices of the cell(s) edited
%   PreviousData: previous data for the cell(s) edited
%   EditData: string(s) entered by the user
%   NewData: EditData or its converted form set on the Data property. Empty if Data
was not changed

%   Error: error string when failed to convert EditData to appropriate value for Data
% handles    structure with handles and user data (see GUIDATA)

table = get(hObject, 'Data');
if(sum(sum(cellfun(@(x) all(isnan(x)), table(:,1:end-1)))) == 0)  % check for nans
   handles.xylist = [];
   for x = table{3,1}:table{4,1}:table{5,1}
      for y = table{3,2}:table[93]:table[94]
         handles.xylist = [handles.xylist;x,y,0];
      end
   end
   scatter(handles.axes_limits, handles.xylist(:,1),handles.xylist(:,2), 'fill');
end

set(handles.sld_jog_x, 'Max', table{2,1}, 'SliderStep', [(1/(table{2,1} - 1)), 0.01]);
set(handles.sld_jog_y, 'Max', table{2,2}, 'SliderStep', [(1/(table{2,2} - 1)), 0.01]);
set(handles.sld_jog_z, 'Max', table{3,2}, 'SliderStep', [(1/(table{2,2} - 1)), 0.01]);
```

```
set(handles.axes_limits, 'XLim', [table{1,1} , table{2,1}], 'YLim', [table{1,2},
table{2,2}]);
guidata(hObject, handles);

% --- Executes on button press in cmd_jog_x_home.
function cmd_jog_x_home_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_x_home (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_home('x');

updateLocation(handles, 'x');

% --- Executes on button press in cmd_jog_x_left.
function cmd_jog_x_left_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_x_left (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('x', '-100');

updateLocation(handles, 'x');



% --- Executes on slider movement.
function sld_jog_x_Callback(hObject, eventdata, handles)
% hObject    handle to sld_jog_x (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
value = round(get(hObject, 'Value'));
handles.stage.block_move('x', sprintf('%d', value));

updateLocation(handles, 'x');

% --- Executes during object creation, after setting all properties.
function sld_jog_x_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sld_jog_x (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
```

```
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on button press in cmd_jog_x_right.
function cmd_jog_x_right_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_x_right (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('x', '+100');

updateLocation(handles, 'x');

function txt_jog_x_Callback(hObject, eventdata, handles)
% hObject    handle to txt_jog_x (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 % Hints: get(hObject,'String') returns contents of txt_jog_x as text
%        str2double(get(hObject,'String')) returns contents of txt_jog_x as a double
 % --- Executes during object creation, after setting all properties.
function txt_jog_x_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txt_jog_x (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in cmd_jog_y_home.
function cmd_jog_y_home_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_y_home (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_home('y');

updateLocation(handles, 'y');
```

```
% --- Executes on button press in cmd_jog_y_left.
function cmd_jog_y_left_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_y_left (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('y', '-100');

updateLocation(handles, 'y');

% --- Executes on slider movement.
function sld_jog_y_Callback(hObject, eventdata, handles)
% hObject    handle to sld_jog_y (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
 value = round(get(hObject, 'Value'));
handles.stage.block_move('y', sprintf('%d', value));

updateLocation(handles, 'y');

% --- Executes during object creation, after setting all properties.
function sld_jog_y_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sld_jog_y (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in cmd_jog_y_right.
function cmd_jog_y_right_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_y_right (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('y', '+100');

updateLocation(handles, 'y');

function txt_jog_y_Callback(hObject, eventdata, handles)
% hObject    handle to txt_jog_y (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of txt_jog_y as text
% --- Executes during object creation, after setting all properties.
function txt_jog_y_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txt_jog_y (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in cmd_jog_z_home.
function cmd_jog_z_home_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_z_home (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_home('z');

updateLocation(handles, 'z');

% --- Executes on button press in cmd_jog_z_left.
function cmd_jog_z_left_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_z_left (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('z', '-1000');

updateLocation(handles, 'z');

 % --- Executes on slider movement.
function sld_jog_z_Callback(hObject, eventdata, handles)
% hObject    handle to sld_jog_z (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
```

```
value = round(get(hObject, 'Value'));
handles.stage.block_move('z', sprintf('%d', value));

updateLocation(handles, 'z');

 % --- Executes during object creation, after setting all properties.
function sld_jog_z_CreateFcn(hObject, eventdata, handles)
% hObject    handle to sld_jog_z (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on button press in cmd_jog_z_right.
function cmd_jog_z_right_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_z_right (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('z', '+1000');

updateLocation(handles, 'z');

function txt_jog_z_Callback(hObject, eventdata, handles)
% hObject    handle to txt_jog_z (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% --- Executes during object creation, after setting all properties.
function txt_jog_z_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txt_jog_z (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
   set(hObject,'BackgroundColor','white');
end
```

```
 % --- Executes on button press in cmd_jog_x_end.
function cmd_jog_x_end_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_x_end (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_end('x');

updateLocation(handles, 'x');

 % --- Executes on button press in cmd_jog_y_end.
function cmd_jog_y_end_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_y_end (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_end('y');

updateLocation(handles, 'y');

% --- Executes on button press in cmd_jog_z_end.
function cmd_jog_z_end_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_z_end (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_end('z');

updateLocation(handles, 'z');

% --- Executes on button press in cmd_data_measure.
function cmd_data_measure_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_data_measure (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[raw, fftx, peaks] = measure(handles);
handles.A(:,end+1) = raw;
handles.B(:,end+1) = fftx;
handles.C{end+1} = peaks;

full = zeros(1,size(handles.B,1));
full(handles.C[95](:,1)) = handles.C[95](:,2);
handles.D(:,end+1) = full';
guidata(hObject, handles);
drawgraphs(handles);
```

```
% --- Executes on button press in cmd_data_clear.
function cmd_data_clear_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_data_clear (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.A = [];
handles.B = [];
handles.C = {};

drawgraphs(handles);

% Update handles structure
guidata(hObject, handles);

% --- Executes on button press in cmd_data_export.
function cmd_data_export_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_data_export (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[FileName,PathName] = uiputfile('*.xls', 'Export to File...', 'davesdata');
filename = fullfile(PathName, FileName);

xlswrite(filename, handles.A, 1);
%xlswrite(filename, handles.B, 2);
xlswrite(filename, cellfun(@num2str, num2cell(handles.B), 'UniformOutput', false), 2);
for i = 1:numel(handles.C)
   xlswrite(filename, handles.C{1,i}, 3, nn2an(1,(2*i)-1));
end
xlswrite(filename, handles.D, 4);

% --- Executes on button press in cmd_jog_x_left2.
function cmd_jog_x_left2_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_x_left2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('x', '-400');

updateLocation(handles, 'x');

% --- Executes on button press in cmd_jog_y_left2.
function cmd_jog_y_left2_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to cmd_jog_y_left2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('y', '-400');

updateLocation(handles, 'y');

% --- Executes on button press in cmd_jog_z_left2.
function cmd_jog_z_left2_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_z_left2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('z', '-4000');

updateLocation(handles, 'z');

% --- Executes on button press in cmd_jog_x_right2.
function cmd_jog_x_right2_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_x_right2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('x', '+400');

updateLocation(handles, 'x');

% --- Executes on button press in cmd_jog_y_right2.
function cmd_jog_y_right2_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_y_right2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('y', '+400');

updateLocation(handles, 'y');

% --- Executes on button press in cmd_jog_z_right2.
function cmd_jog_z_right2_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_z_right2 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.block_move('z', '+4000');

updateLocation(handles, 'z');
```

```
% --- Executes on button press in cmd_limits_run.
function cmd_limits_run_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_limits_run (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

pnadata = [];
maxmean = 0;
if(get(handles.chk_pnameasure, 'Value'))
   handles.pna.create_trace(get(handles.txtPNAMeasureName, 'String'),
get(handles.txtPNAMeasureS, 'String'), 1);
end

for i = 1:size(handles.xylist)
   handles.stage.block_move('x', sprintf('%d', handles.xylist(i,1)), 'y', sprintf('%d',
handles.xylist(i,2)));
   pause(1.0);

if(get(handles.chk_pnameasure, 'Value'))
   measure = handles.pna.read_data_db(get(handles.txtPNAMeasureName, 'String'));
  pnadata = [pnadata; measure];
  if (i == 1 || maxmean < mean(measure))
      maxmean = mean(measure);
      set(handles.lblPNAmin, 'String', sprintf('Max Measure\nu:%f\nx:%d\ny:%d',
mean(measure), handles.xylist(i,1), handles.xylist(i,2)));
  end
end
 end

if(get(handles.chk_pnameasure, 'Value'))
     save(sprintf('%s.mat', handles.PNAfilename), 'pnadata');
end

function [raw, fftd, peaks] = measure(handles)
   fprintf(handles.fmcw,'TRIG:ARM');
   reply = fscanf(handles.fmcw);

   pause(0.1);
   fprintf(handles.fmcw,'TRACE:DATA ?') ;

   data = '00';
   raw = 0;

   while not(strcmp('OK', data(1:2)))
```

```
        data = fscanf(handles.fmcw);
        raw = [raw,str2double(data)];
    end

    raw = raw(2:end-1);

    N=length (raw);
    fftd = fft(raw,N);
    fftd = fftd(20:floor(end/2));

    [pks, lks] = findpeaks(abs(fftd));
    peaks = [lks; pks]';

function drawgraphs(handles)
if(isempty(handles.A))
    cla(handles.axes_data_fft);
    cla(handles.axes_data_raw);
else
    plot(handles.axes_data_raw, handles.A)
    fftx = 1:size(handles.B,1);
    plot(handles.axes_data_fft,
fftx,abs(handles.B),fftx,(std(abs(handles.B(:,1))))+1.5*mean(abs(handles.B(:,1))))
    end

function updateLocation(handles, axis)

switch(axis)

    case 'x'
        location = handles.stage.query('x');
        set(handles.txt_jog_x, 'String', location);
        set(handles.sld_jog_x, 'Value', str2num(location));
    case 'y'
        location = handles.stage.query('y');
        set(handles.txt_jog_y, 'String', location);
        set(handles.sld_jog_y, 'Value', str2num(location));
    case 'z'
        location = handles.stage.query('z');
        set(handles.txt_jog_z, 'String', location);
        set(handles.sld_jog_z, 'Value', str2num(location));

end

% --- Executes on button press in cmd_laser_on.
function cmd_laser_on_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_laser_on (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.laser('on');

% --- Executes on button press in cmd_laser_off.
function cmd_laser_off_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_laser_off (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.stage.laser('off');

% --- Executes on button press in chk_laser_measure.
function chk_laser_measure_Callback(hObject, eventdata, handles)
% hObject    handle to chk_laser_measure (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of chk_laser_measure

% --- Executes on button press in cmd_jog_joy.
function cmd_jog_joy_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_jog_joy (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

pad = vrjoystick(1);
set(handles.cmd_jog_joy, 'String', 'Press Space to regain control');
drawnow
buttons = zeros(1,16);
speedcontrol = 0;
lasterror = 0;

while (buttons(16) == 0)
   array = {};
   [axes, buttons, povs] = read(pad)

%    switch(povs(1))
 %       case -1
         if(axes(1) > 0.2)
            if(speedcontrol)
               speed = round(abs(axes(1)) * 10);
            else
```

```
            speed = 10;
        end
        array = [array;'x';sprintf('+%d', speed);sprintf('%d', speed)];
    end
    if(axes(1) < -0.2)
        if(speedcontrol)
            speed = round(abs(axes(1)) * 10);
        else
            speed = 10;
        end
        array = [array;'x';sprintf('-%d', speed);sprintf('%d', speed)];
    end
    if(axes(7) > 0.2)
        if(speedcontrol)
            speed = round(abs(axes(7)) * 10);
        else
            speed = 10;
        end
        array = [array;'y';sprintf('-%d', speed);sprintf('%d', speed)];
    end
    if(axes(7) < -0.2)
        if(speedcontrol)
            speed = round(abs(axes(7)) * 10);
        else
            speed = 10;
        end
        array = [array;'y';sprintf('+%d', speed);sprintf('%d', speed)];
    end

if(buttons(9))
    handles.stage.home('x');
end
if(buttons(10))
    handles.stage.home('y');
end
if(buttons(11))
    handles.stage.hend('x');
end
if(buttons(12))
    handles.stage.hend('y');
end

if(buttons(5) && not(buttons(7)))
    array = [array;'z';'+10';'10'];
elseif(buttons(7) && not(buttons(5)))
    array = [array;'z';'-10';'10'];
```

```
    end

    if(buttons(6) && not(buttons(8)))
        handles.stage.hend('z');
    elseif(buttons(8) && not(buttons(6)))
        handles.stage.home('z');
    end

    if(buttons(4) && not(buttons(2)))
        handles.stage.laser('off');
    elseif(buttons(2) && not(buttons(4)))
        handles.stage.laser('on');
    end

    if(buttons(13))
        [raw, fftx, peaks] = measure(handles);
        handles.A(:,end+1) = raw;
        handles.B(:,end+1) = fftx;
        handles.C{end+1} = peaks;
            full = zeros(1,size(handles.B,1));
    full(handles.C[95](:,1)) = handles.C[95](:,2);
    handles.D(:,end+1) = full';
        guidata(hObject, handles);
        drawgraphs(handles);
        drawnow
    end

%    if(buttons(9))
%        drawnow;
%    end

    if(buttons(3))
        handles.A = [];
        handles.B = [];
        handles.C = {};
        guidata(hObject, handles);
        drawgraphs(handles);
        drawnow
    end

    if(buttons(15))
        speedcontrol = 1;
    end

    if(buttons(14))
        speedcontrol = 0;
```

```
    end

    if(buttons(1) && not(lasterror))
    set(handles.panel_error, 'Visible', 'on');
    set(handles.txt_error_1, 'Visible', 'on');
    set(handles.txt_error_2, 'Visible', 'on');

    drawnow
    elseif(lasterror && not(buttons(1)))
    set(handles.panel_error, 'Visible', 'off');
       set(handles.txt_error_1, 'Visible', 'off');
    set(handles.txt_error_2, 'Visible', 'off');
    drawnow
    end

    lasterror = buttons(1);

    handles.stage.move_speed(array[38]);
end
%handles.stage.home('x', 'y');
updateLocation(handles, 'x');
updateLocation(handles, 'y');
updateLocation(handles, 'z');

set(handles.cmd_jog_joy, 'String', 'Enable Voice Control');

function cr = nn2an(r, c)
% convert number, number format to alpha, number format
%t = [floor(c/27) + 64 floor((c - 1)/26) - 2 + rem(c - 1, 26) + 65];
t = [floor((c - 1)/26) + 64 rem(c - 1, 26) + 65];
if(t(1)<65), t(1) = []; end
cr = [char(t) num2str(r)];

% --- Executes on button press in cmd_limit_load.
function cmd_limit_load_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_limit_load (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
variable = inputdlg('Workspace Variable Name:');

handles.xylist = getappdata(0,variable[38]);
scatter(handles.axes_limits, handles.xylist(:,1),handles.xylist(:,2), 'fill');
guidata(hObject, handles);

function tabConnections_ClickedCallback(hObject, eventdata, handles)
```

```
% hObject    handle to tabConnections (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tabHandler(handles, 0);

function tabStage_ClickedCallback(hObject, eventdata, handles)
% hObject    handle to tabStage (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tabHandler(handles, 1);

function tabMeasure_ClickedCallback(hObject, eventdata, handles)
% hObject    handle to tabMeasure (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tabHandler(handles, 2);

function tabAuto_ClickedCallback(hObject, eventdata, handles)
% hObject    handle to tabAuto (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
tabHandler(handles, 3);

function tabHandler(handles, view)
set(handles.tabConnections, 'State', 'off');
set(handles.tabStage, 'State', 'off');
set(handles.tabMeasure, 'State', 'off');
set(handles.tabAuto, 'State', 'off');

switch(view)
   case 0
      set(handles.panelConnections, 'Visible', 'On');
      set(handles.panelStage, 'Visible', 'Off');
      set(handles.panelMeasure, 'Visible', 'Off');
      set(handles.panelAuto, 'Visible', 'Off');
   case 1
      set(handles.panelConnections, 'Visible', 'Off');
      set(handles.panelStage, 'Visible', 'On');
      set(handles.panelMeasure, 'Visible', 'On');
      set(handles.panelAuto, 'Visible', 'Off');
   case 2
      set(handles.panelConnections, 'Visible', 'Off');
      set(handles.panelStage, 'Visible', 'On');
      set(handles.panelMeasure, 'Visible', 'On');
      set(handles.panelAuto, 'Visible', 'Off');
   case 3
```

```
        set(handles.panelConnections, 'Visible', 'Off');
        set(handles.panelStage, 'Visible', 'Off');
        set(handles.panelMeasure, 'Visible', 'Off');
        set(handles.panelAuto, 'Visible', 'On');
end

function tabSetup(handles)
    pos = get(handles.figure1, 'Position');
    pad = 2;
    w = pos(3)% - 2*pad;
    h = pos(4)% - 2*pad;
    set(handles.panelConnections, 'Position', [pad, pad, w-2*pad, h-2*pad]);
    set(handles.panelStage, 'Position', [pad, (h-12-pad), w-2*pad, 12]);
    set(handles.panelMeasure, 'Position', [pad, pad, w-pad, h-12-2*pad]);
    set(handles.panelAuto, 'Position', [pad, pad, w-2*pad, h-2*pad]);

% --- Executes on button press in chk_pnameasure.
function chk_pnameasure_Callback(hObject, eventdata, handles)
% hObject    handle to chk_pnameasure (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of chk_pnameasure
if(get(hObject,'Value'))
    if(not(isfield(handles,'PNAfilename')))
        set(hObject,'Value','off');
    end
end

 function txtPNAMeasureName_Callback(hObject, eventdata, handles)
% hObject    handle to txtPNAMeasureName (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txtPNAMeasureName as text
%        str2double(get(hObject,'String')) returns contents of txtPNAMeasureName as a
double

 % --- Executes during object creation, after setting all properties.
function txtPNAMeasureName_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txtPNAMeasureName (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txtPNAMeasureS_Callback(hObject, eventdata, handles)
% hObject    handle to txtPNAMeasureS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of txtPNAMeasureS as text
%        str2double(get(hObject,'String')) returns contents of txtPNAMeasureS as a
double

% --- Executes during object creation, after setting all properties.
function txtPNAMeasureS_CreateFcn(hObject, eventdata, handles)
% hObject    handle to txtPNAMeasureS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

 % --- Executes on button press in cmdPNAFile.
function cmdPNAFile_Callback(hObject, eventdata, handles)
% hObject    handle to cmdPNAFile (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[FileName,PathName] = uiputfile('*.mat', 'Export to File...', 'davesdata');
handles.PNAfilename = fullfile(PathName, FileName);
guidata(hObject, handles);

% --- Executes on button press in cmdPNA.
function cmdPNA_Callback(hObject, eventdata, handles)
% hObject    handle to cmdPNA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.pna = PNA;
set(hObject, 'String', 'PNA CONNECTED');
guidata(hObject, handles);

% --- Executes on button press in cmd_save_mat.
```

```
function cmd_save_mat_Callback(hObject, eventdata, handles)
% hObject    handle to cmd_save_mat (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
a = handles.A;
b = handles.B;
c = handles.C;
d = handles.D;
uisave({'a', 'b', 'c', 'd'})
```

References

[1] A. Nobel. (2016). *About Resicoat - AkzoNobel Powder Coatings*. Available: https://www.akzonobel.com/powder/aboutus/about_akzonobel_powder_coatings/about_resicoat/

[2] (2016). *Dudley firm invests £3m to tackle corrosion worldwide*. Available: http://www.kidderminstershuttle.co.uk/news/business_daily/10685005.Dudley_firm_invests___3m_to_tackle_corrosion_worldwide/

[3] G. H. Koch, M. P. Brongers, N. G. Thompson, Y. P. Virmani, and J. H. Payer, "Corrosion cost and preventive strategies in the United States," 2002.

[4] G. H. Koch, M. P. Brongers, N. G. Thompson, Y. P. Virmani, and J. Payer, "Cost of corrosion study unveiled," *Supplement to Materials Performance,* vol. 2, 2002.

[5] M. Lettich, "Is There A Cure for Corrosion under Insulation," *Insulation Outlook Magazine,* 2005.

[6] A. A. Mokhtar, N. Saari, and M. C. Ismail, "Assessment of Insulated Piping System Inspection Using Logistic Regression," in *Engineering Asset Management-Systems, Professional Practices and Certification*, ed: Springer, 2015, pp. 265-277.

[7] D. American Petroleum Institute. Manufacturing and M. Department, *Piping Inspection Code: Inspection, Repair, Alteration, and Rerating of In-service Piping Systems: API 570*: American Petroleum Institute, 1993.

[8] (2015, 25/08/15). *API 570 | API Standards Store*. Available: http://www.techstreet.com/api/products/1654366#jumps

[9] (2016). *Insul-8Â® HT - Spray Applied Polyurethane Foam Insulation for High Temperature Pipelines*. Available: http://www.brederoshaw.com/solutions/onshore/insul-8_ht.html

[10] (2016). *CUI, or how small solutions can solve big problems*. Available: http://www.temati.com/news/69-cui-or-how-small-solutions-can-solve-big-problems

[11] (2016). *Chem1 Electrochemistry: Electrochemical corrosion*. Available: http://www.chem1.com/acad/webtext/elchem/ec7.html

[12] (2016). *Understanding Cathodic Protection System | ogd*. Available: http://www.raychemrpg.com/ogd/?page_id=586

[13] R. W. Revie, *Corrosion and corrosion control*: John Wiley & Sons, 2008.

[14] (2016). *Corrosion College White Paper*. Available: http://www.corrosioncollege.com/white-paper.cfm

[15] (2016). *SSINA: Stainless Steel: Corrosion*. Available: http://www.ssina.com/corrosion/galvanic.html

[16] D. Spencer, M. Edwards, M. Wenman, C. Tsitsios, G. Scatigno, and P. Chard-Tuckey, "The initiation and propagation of chloride-induced transgranular stress-corrosion cracking (TGSCC) of 304L austenitic stainless steel under atmospheric conditions," *Corrosion Science,* vol. 88, pp. 76-88, 2014.

[17] T. Prosek, A. Le Gac, D. Thierry, S. Le Manchet, C. Lojewski, A. Fanica*, et al.*, "Low-temperature stress corrosion cracking of austenitic and duplex stainless steels under chloride deposits," *Corrosion,* vol. 70, pp. 1052-1063, 2014.

[18] "Stress Corrosion Cracking (SCC) - Integrated Global," ed, 2016.

[19] (2016). *STAGES OF RUST AND CORROSION*. Available: http://inspectionprocedures.tpub.com/TB-1-1730-229-30-3/TB-1-1730-229-30-30011.htm

[20] (2016). *Corrosion Stages/Scale | Rust Bullet Australasia*. Available: http://www.rustbullet.com.au/technical/how-it-works/corrosion-stagesscale/

[21] (2016). *External Corrosion Coating - Coating - Offshore and Onshore Line Pipe - Tenaris*. Available: http://www.tenaris.com/en/products/offshorelinepipe/coating/externalanticorrosion.aspx

[22] (2016). *SPC - Specialty Polymer Coatings | Pipeline | Product Usage Selection Matrix*. Available: http://www.spc-net.com/pipeline/products/use.php#.VsS-aPKLSUl

[23] A. Nobel, "AkzoNobel working with Siemens to supply coatings for German power plant," 2016.

[24] R. Williams, "Williams Form Engineering Corp. - What's new at Williams," 2016.

[25] "ChrōmX 9000 Series - MMFX," 2016.

[26] C. Baumeister and L. Kilian, "Understanding the Decline in the Price of Oil since June 2014," 2015.

[27] I. V. Ltd. (2015, 12 January 2015). *CUI - Corrosion Under Insulation Thermal Solutions*. Available: http://www.infraredvision.co.uk/cui-corrosion-under-insulation/

[28] C. Balaras and A. Argiriou, "Infrared thermography for building diagnostics," *Energy and buildings,* vol. 34, pp. 171-183, 2002.

[29] N. Laaidi, S. Belattar, and A. Elbaloutti, "Pipeline corrosion, modeling and analysis," *Journal of Nondestructive Evaluation,* vol. 30, pp. 158-163, 2011.

[30] J.-L. Shih, K.-T. Wu, C.-K. Jen, C.-H. Chiu, J.-C. Tzeng, and J.-W. Liaw, "Applications of Flexible Ultrasonic Transducer Array for Defect Detection at 150 °C," *Sensors,* vol. 13, p. 975, 2013.

[31] M. Siqueira, C. Gatts, R. Da Silva, and J. Rebello, "The use of ultrasonic guided waves and wavelets analysis in pipe inspection," *Ultrasonics,* vol. 41, pp. 785-797, 2004.

[32] W. Cheng, "Pulsed eddy current testing of carbon steel pipes' wall-thinning through insulation and cladding," *Journal of Nondestructive evaluation,* vol. 31, pp. 215-224, 2012.

[33] S. Winnik, *Corrosion under insulation (CUI) guidelines*: Elsevier, 2014.

[34] M. Twomey, "Inspection techniques for detecting corrosion under insulation," *Materials evaluation,* vol. 55, 1997.

[35] R. E. Jones, F. Simonetti, M. Lowe, I. Bradley, D. O. Thompson, and D. E. Chimenti, "Use of Microwaves for the Detection of Corrosion Under Insulation," Imperial College London, 2012.

[36] D. Fleisch, *A Student's Guide to Maxwell's Equations*: Cambridge University Press, 2008.

[37] P. G. Huray, *Maxwell's Equations*: Wiley, 2010.

[38] S. Drabowitch, A. Papiernik, H. D. Griffiths, J. Encinas, and B. L. Smith, *Modern Antennas*: Boston, MA : Springer US, 2005.
2nd edition., 2005.

[39] A. W. Love, *Electromagnetic horn antennas*: IEEE, Press, 1976.

[40] C. A. Balanis, *ANTENNA THEORY: ANALYSIS AND DESIGN, 3RD ED (With CD)*: Wiley India Pvt. Limited, 2009.

[41] (2016). *Isotropic Radiation*. Available: http://electriciantraining.tpub.com/14182/css/14182_179.htm

[42] D. I. C. Wolff, "Radar Basics - Isotropic Radiator," 2016.

[43] (2016). *Some Common Antenna Radiation Patterns | RAYmaps*. Available: http://www.raymaps.com/index.php/some-common-antenna-radiation-patterns/

[44] 2016.

[45] P. Bevelacqua, "Directivity - Antenna-Theory.com," 2016.

[46] (2016). *Waveguide Tutorial | Microwave Feeder | Radio-Electronics.com*. Available: http://www.radio-electronics.com/info/antennas/waveguide/waveguide-basics-tutorial.php

[47] S. F. Mahmoud, *Electromagnetic Waveguides: Theory and Applications*: Peregrinus, 1991.

[48] T. U. Editor, "Microwaves101 | Waveguide Mathematics," 2016.

[49] R. Sullivan, *Radar foundations for imaging and advanced concepts*: The Institution of Engineering and Technology, 2004.

[50] C. Cook, *Radar signals: An introduction to theory and application*: Elsevier, 2012.

[51] S. Misra, "Small and Short-Range Radar Systems (Gregory L. Charvat; 2014)[Book Review]," *Wireless Communications, IEEE,* vol. 22, pp. 11-11, 2015.

[52] G. M. Brooker, "Understanding millimetre wave fmcw radars," in *1st international Conference on Sensing Technology*, 2005, pp. 152-157.

[53] "FMCW Radar App Note," ed, 2016, pp. 43-44.

[54] (2016). *Elements of Electrical Engineering: Magnetic Material*. Available: http://elementsofelectricalengineering.blogspot.co.uk/2014/10/magnetic-material.html

[55] "RS3400K/00," 2016.

[56] (2016). *Arduino - SPI*. Available: https://www.arduino.cc/en/Reference/SPI

[57] "CO1000A/00," 2016.

[58] (2016). *MATLAB GUI - MATLAB - MathWorks United Kingdom*. Available: http://uk.mathworks.com/discovery/matlab-gui.html?s_tid=gn_loc_drop

[59] Y.-J. Yu, K. Hearon, T. S. Wilson, and D. J. Maitland, "The effect of moisture absorption on the physical properties of polyurethane shape memory polymer foams," *Smart materials & structures,* vol. 20, p. 085010, 2011.

[60] www.goodfellow.com, "Polymethylmethacrylate - online catalogue source - supplier of research materials in small quantities - Goodfellow," 2016.

[61] www.goodfellow.com, "Polyethylene - High density - online catalogue source - supplier of research materials in small quantities - Goodfellow," 2016.

[62] C. C. Ku and R. Liepins, *Electrical properties of polymers*: MacMillan Publishing Company., 1993.

[63] K. P. Menard, *Dynamic mechanical analysis: a practical introduction*: CRC press, 2008.

[64] D. S. Jones, "Dynamic mechanical analysis of polymeric systems of pharmaceutical and biomedical significance," *International journal of pharmaceutics,* vol. 179, pp. 167-178, 1999.

[65] G. Höhne, W. F. Hemminger, and H.-J. Flammersheim, *Differential scanning calorimetry*: Springer Science & Business Media, 2013.

[66] P. J. Haines, *Thermal methods of analysis: principles, applications and problems*: Springer Science & Business Media, 2012.

[67] R. Berger, "Predictive maintenance–Is the timing right for predictive maintenance in the manufacturing sector?| Publications| Media| Roland Berger," 2015.

[68] M. J. Dupuy, D. E. Wesely, and C. S. Jenkins, "Airline fleet maintenance: Trade-off analysis of alternate aircraft maintenance approaches," in *Systems and Information Engineering Design Symposium (SIEDS), 2011 IEEE*, 2011, pp. 29-34.

[69] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing,* vol. 20, pp. 1483-1510, 2006.

[70] D. Herd and D. Flynn, "A Review of Condition Monitoring and the Progression to Prognostic Health Management," in *Wind Energy Network Magazine*, October / November ed, 2012, pp. 66-67.

[71] D. S. Herd and D. Flynn, "Prognostic Health Management techniques for intelligent condition monitoring of offshore renewable generation assets," in *Renewable Power Generation Conference (RPG 2013), 2nd IET*, 2013, pp. 1-4.

[72] Z. Xu, D. Koltsov, A. Richardson, L. Le, and M. Begbie, "Design and simulation of a multi-function MEMS sensor for health and usage monitoring," in *Prognostics and Health Management Conference, 2010. PHM '10.*, 2010, pp. 1-7.

[73] S. Mathew, D. Das, R. Rossenberger, and M. Pecht, "Failure mechanisms based prognostics," in *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, 2008, pp. 1-6.

[74] A. Alaeddini and I. Dogan, "Using Bayesian networks for root cause analysis in statistical process control," *Expert Systems with Applications,* vol. 38, pp. 11230-11243, 2011.

[75] K. R. McNaught and A. Zagorecki, "Using dynamic Bayesian networks for prognostic modelling to inform maintenance decision making," in *Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on*, 2009, pp. 1155-1159.

[76] L. Pan, N. Ning, A. Chen, and Y. Tong, "Fault Diagnosis of the Blast Furnace Based on the Bayesian Network Model," in *Electrical and Control Engineering (ICECE), 2010 International Conference on*, 2010, pp. 990-993.

[77] J. W. Hines, D. J. Wrest, and R. E. Uhrig, "Use of Autoassociative Neural Networks for Signal Validation," 1998.

[78] B. R. Upadhyaya and E. Eryurek, "Application of neural networks for sensor validation and plant monitoring," *Journal Name: Nuclear Technology; (United States); Journal Volume: 97:2,* pp. Medium: X; Size: Pages: 170-176, 1992.

[79] X. Xu, J. W. Hines, and R. E. Uhrig, "Sensor Validation and Fault Detection Using Neural Networks," 1999.

[80] C. Chaochao, Z. Bin, and G. Vachtsevanos, "Prediction of Machine Health Condition Using Neuro-Fuzzy and Bayesian Algorithms," *Instrumentation and Measurement, IEEE Transactions on,* vol. 61, pp. 297-306, 2012.

[81] F. Jiajie, K. C. Yung, and M. Pecht, "Physics-of-Failure-Based Prognostics and Health Management for High-Power White Light-Emitting Diode Lighting," *Device and Materials Reliability, IEEE Transactions on,* vol. 11, pp. 407-416, 2011.

[82] M. Pecht and J. Gu, "Physics-of-failure-based prognostics for electronic products," *Transactions of the Institute of Measurement and Control,* vol. 31, pp. 309-322, 2009.

[83] M. G. Pecht, *Prognostics and Health Management of Electronics*: John Wiley & Sons, 2008.

[84] M. Pecht and R. Jaai, "A prognostics and health management roadmap for information and electronics-rich systems," *Microelectronics Reliability,* vol. 50, pp. 317-323, 3// 2010.

[85] (2016). *Modulus of Elasticity or Young's Modulus - and Tensile Modulus for some common Materials*. Available: http://www.engineeringtoolbox.com/young-modulus-d_417.html

[86] L. L. C. Engineers Edge, "Yield Strength - Strength (Mechanics) of Materials - Engineers Edge," 2016.

[87] R. Lakes, "Meaning of Poisson's ratio," ed.

[88] D. Binder, "Looking Back to the Future: The Curmudgeon's Guide to the Future of Environmental Law," *Akron L. Rev.,* vol. 46, p. 993, 2013.

[89] "ERW Standard and Line Pipe Grades - Continental Steel & Tube Company," 2016.

[90] L. Vollmer, "Hydrogen Sulphide Corrosion Cracking of Steel★," *Corrosion,* vol. 8, pp. 326-332, 1952.

[91] J. Smith and J. Miller, "Nature of sulphides and their corrosive effect on ferrous metals: a review," *British Corrosion Journal,* vol. 10, pp. 136-143, 1975.

[92] M. Desmulliez, S. K. Pavuluri, D. Flynn, and D. Herd, "Microwave cavity sensor," ed: Google Patents, 2013.

[93] Y. Z. Rosunally, S. Stoyanov, C. Bailey, P. Mason, S. Campbell, G. Monger*, et al.*, "Fusion Approach for Prognostics Framework of Heritage Structure," *Reliability, IEEE Transactions on,* vol. 60, pp. 3-13, 2011.

[94] A. C. M. Fong and B. Fong, "Prognostic health management for environmental monitoring," in *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, 2010, pp. 1279-1283.

[95] V. A. E. Barrios, J. R. R. Méndez, N. V. P. Aguilar, G. A. Espinosa, and J. L. D. Rodríguez, "FTIR–An Essential Characterization Technique for Polymeric Materials," *Infrared Spectroscopy-Materials Science, Engineering and Technology, Prof. Theophanides Theophile (Ed.), ISBN,* pp. 978-953, 2012.