The University of Nottingham

Faculty of Engineering

Department of Electrical and Electronic Engineering

MPhil Thesis

Intelligent Real-Time Prediction for Energy and Sensing Applications

AUTHOR	:	Arun Kumar a/l Ramasamy
ID	:	UNIMKL - 025649
SUPERVISOR	:	Dr Mumtaj Begam
DATE	:	2nd August 2021

Declaration

I declare that the work in this thesis was carried out in accordance with the regulations of The University of Nottingham Malaysia. It is original and is the result of my own work, unless otherwise indicated or acknowledged as referenced work. This thesis has not been submitted to any other academic institution or non-academic institution for any other degree or qualification. I, hereby, acknowledge that I have been supplied with the Academic Rules and Regulations for Post Graduate, The University of Nottingham Malaysia, regulating the conduct of my study and research.

Name: Arun Kumar a/l Ramasamy

Signature: 

Abstract

The advancements of science and technology has been rapid since the boom of the fourth industrial revolution that began arguable about a decade ago. As smart hardware and software began to be paired together over high speed transfer of information, the world of innovation witnessed the rise of Big Data and Machine Learning.

These 2 heroes of the 21st centuries have been widely embraced and adopted in various industries, resulting in innovations and outcomes that we never could have perceived otherwise, especially in closing the gaps between probability and predictability i.e. stock market predictions and such.

However, one gigantic industry that has yet to reap on the offerings of Big Data and Machine Learning is the oil and gas industry. As extreme a form of engineering it is, methods and technologies are still primarily mechanically driven, specifically when it comes to safety and preventive measures i.e. in failure prediction efforts. Manual methods using predated technologies are still industry standard for many applications within industry.

This research takes a look at these current methods, and proposes a new way of performing failure prediction analysis using machine learning.

Acknowledgements

I would like to express my heartfelt gratitude to Dr. Rajprasad Kumar, who was my initial supervisor, for being always reachable for support, guidance and encouragement throughout this journey. He had truly been a point of reference for everything I had been able to achieve, and has never declined to a request for advice no matter the time or moment.

I wish the late Prof. Dino Isa had been still alive today to witness this journey having been my second supervisor when I began. Thank you Prof for believing in me, beyond academics, right from the undergraduate days. Prof Dino had been a mentor to me, with teachings and life lessons I will take to the grave.

I would like to thank Dr. Wong Yee Wan and Dr. Mumtaj Begum both for coming along half way to help me complete this work. The both of them filled in during a time of challenge for me as replacement supervisors. I appreciate them both truly. I also thank my family for their continuous support over the last few years.

Finally, I'd like to thank God without whom, none of these would have been possible. His grace had been sufficient for me throughout.

Table of Contents

List of Tables

List of Abbreviations

Abbreviation	Full text
NDT	Non Destructive Testing
NDE	Non Destructive Evaluation
SVM	Support Vector Machines
SVC	Support Vector Classifier
LRUT	Long Range Ultrasonic Transducer
FPGA	Field Programmable Gate Arrays

List of Figures

Chapter 1: Introduction

1.1 Overview

One of the biggest industries in the world over the last century had been the oil and gas industry. It has contributed greatly towards the advancement of mankind through many areas ranging from household applications to transportation, to the global economy in general. However, due to its volatility and high risk in managing the raw material, there have been many great disasters within the industry that has claimed many lives and caused several significant damages to the environment and humanity in general.

1.2 Problem Statement

The rising risk of handling pure oil and gas substance despite its importance in everyday life has generated a great need for the intervention of technology in the space of safety and security. Generally, the primary mode of transporting oil and gas from one location to another is via a pipeline. Since pipelines are man-made, like other man-made products they are subject to deterioration in quality and health, and therefore if not assessed for integrity periodically can result in catastrophic disasters like the ones that has happened in history. However, periodic methods are costly and provide plenty of room for error between inspection schedules.

Therefore, we have to replace periodic methods of Non-Destructive Testing via inline inspection with a continuous monitoring system utilising

permanently placed LRUT transducers on the line combined with a Machine Learning model on a local CPU.

1.3 Research Background

Non-Destructive Testing became the methodology of integrity assessments for oil and gas pipelines with a comprehensive code of inspection developed progressively over time. Technologically, the methodologies evolved from regular manual ultrasonic wall-thickness testing to what is now known as intelligent in-line inspection technologies via pipeline inspection gauges (PIG).

1.4 Aim and Objectives

This research aims to set the foundations in producing an intelligent real time prediction tool for energy and sensing applications for the oil and gas pipelines. Such a tool can potentially help prevent future disasters by a great factor. Although there have been events such as Piper Alpha[1] and Deepwater Horizon[2] that were largely due to human error, research shows that most pipeline failures are caused by corrosion, and faulty welding can go as high as 61% [3].

Therefore the objectives of this research are as follows:

- To setup pipeline in the lab where corrosions and wall defects will be simulated
- To develop microcontroller to process corrosions and defects data

- To evaluate machine learning model hosted on a Raspberry Pi to enable real-time unsupervised failure predictions on the

This research in essence will begin with a pipeline setup in the lab where corrosions and wall defects will be simulated. Data acquired will then be sent to a microcontroller to be processed and then sent to a machine learning model hosted on the Raspberry Pi to enable real-time unsupervised failure predictions on the pipeline.

Therefore by automating Non-Destructive Testing methodologies[4] using machine learning, we are able to push preventive maintenance to whole new level of increased safety and reliability within the oil and gas industry.

1.5 Overall Methodology

This research is divided into three phases as seen in Figure 1. Beginning with the process of generating and transmitting suitable signals into the pipeline via LRUT transducers, to receiving, sampling and optimizing the incoming signals, before sending it to the machine learning model to be trained and tested with.

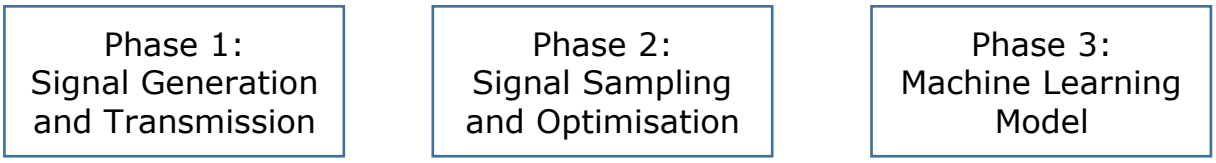

Figure 1: Three phases of the process methodology

It explores and documents in detail the microcontroller and the process flow from the simulated data from the pipeline setup to the Raspberry Pi that hosts the machine learning algorithm as shown in Figure 2.

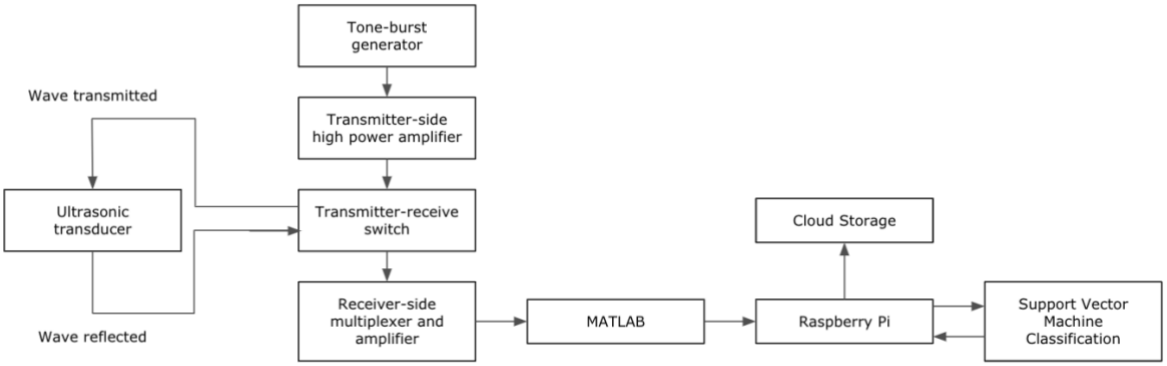

Figure 2: System block diagram

1.6 Summary

Given the current methods of performing pipeline integrity assessments, which is still periodical and can incur high costs, this research will aim to produce an alternative method that provides real time failure predictions via machine learning.

The overall process will include hardware and software integration with sensors mounted on a pipeline setup that collects wall thickness data. This data will then be sent to a microcontroller that samples it for a machine learning model for failure prediction analysis.

Chapter 2: Literature Review

2.1 Introduction

This chapter aims to answer these 2 key questions that form the founding block to this research.

1. How are failure predictions and integrity assessments being conducted at the moment?
2. What are other ways to conduct failure predictions and integrity assessments moving forward?

A deep dive on traditional methods of mechanical engineering to perform Non-Destructive Testing is done in section 2.1, with potential new methods employing machine learning being then explored in section 2.2. These two sections aims to explain the types of technologies available, whereas section 2.3 will discuss the research gaps and how this research work will aim to over the previous drawbacks.

2.2 Non-Destructive Desting (NDT)

Non-Destructive Desting, also popularly known in the industry as NDT is basically, the science of assessing the integrity of anything without having to destroy it to find out its breaking point, evaporating to find the boiling point and such[5]. Non-Destructive Testing, NDT is also known as Non-Destructive Evaluation, NDE[6].

NDT applies to pretty much any industry that involves manufacturing, construction and anything that requires the usage of a man-made item. It is widely used in constructions to inspect and assess the hardness of concrete, steel, bricks, also in the case of manufacturing and fabrication, to inspect the breaking point of vessels and more.

In the energy sector, NDT plays perhaps one of the most pivotal roles in ensuring the stability of the industry. Energy can be very volatile especially when it is produced using fossil fuels, and when something is volatile, it is highly important to ensure its operations are very reliable. The annual cost of corrosion in the USA oil and gas industry is more than USD27 billion which estimates the cost for the whole world to be north of USD60 billion[7].

There is severe importance to make sure a tank that holds acid does not have surprises that leaks out corrosive material under certain temperature or pressure. A boiler in the oil and gas industry should be made sure to not have hollow walls that can burst out poisonous gases during high performance operations. Most commonly, pipelines that carry products such as gas, oil or even sludge through the sea beds or even underneath roads within cities have to be constantly inspected for corrosion to avoid bursts under a certain operating pressure.

In order to achieve all these assessments, there had to be an intelligent technique that can provide a highly accurate prediction of failure simply by inspecting and updating current conditions of a particular system. In other words, one does not need to wait to see the pipe burst to identify it’s

breaking point. NDT does precisely that, and it has enabled plants to sustain while assessment is conducted during operations or in some extreme conditions, during plant shut downs.

2.2.1 Intelligent Pigging

In pipelines, non-destructive testing is conducted via a method known as intelligent pigging. Although the name doesn't sound as scientific as it should for an industry of such a stature, pigging is actually the spoken term for what is actually a very advanced technology that is called the Pipeline Inspection Gauge (PIG). This pig houses a setup of sensors and microcontroller. It is used to collect wall thickness profile of a pipeline through what is known as in-line inspection.[8]

This is a process that requires the PIG to be inserted through a launcher at one end of the pipeline and then received at the receiver at the other end after running through the line under control conditions to gather data as it runs.[9]

Upon retrieving the pig (also referred to as a "tool" in the industry), figure 2, borrowed from an established NDT company called Penspen Integrity describes well the average process of conducting an integrity assessment also known as a failure prediction analysis using an intelligent pig.

The entire process begins with the engagement of a vendor and operator to understand the pipeline health based on the preliminary assessment which is basically a short process of browsing through the data to seek for

severe defects while on site itself, upon retrieval of the tool. This is vital to make sure both parties are completely aware of the situation, requirements and consequences.

In the event of an unsatisfactory quality of data, the inspection run will have to be repeated until sufficient quality is achieved. Poor data quality would make a poor assessment which yields a completely redundant effort. Considering sufficient data quality is obtained under good circumstances, the pipeline health is virtually assessed via software to determine if a level 1 or 2 integrity assessment is needed. Should a level 1 assessment suffice, the vendor affirms the operator that the pipeline health is satisfactory and proceeds to schedule the next inspection date.

If it was found that the pipeline had suffered some compromising defects, a level 2 assessment is then conducted to provide operators with a comprehensive report that helps them decide on new operating conditions or in some cases, even the decision to discharge the line from future use.

This is also usually the case in the event initial virtual assessment shows severe defects on the pipeline. An expert assessment is conducted immediately. The pipeline is then shut down for repairs and other necessary actions. As mentioned earlier, the entire process is summarized in Figure 3.

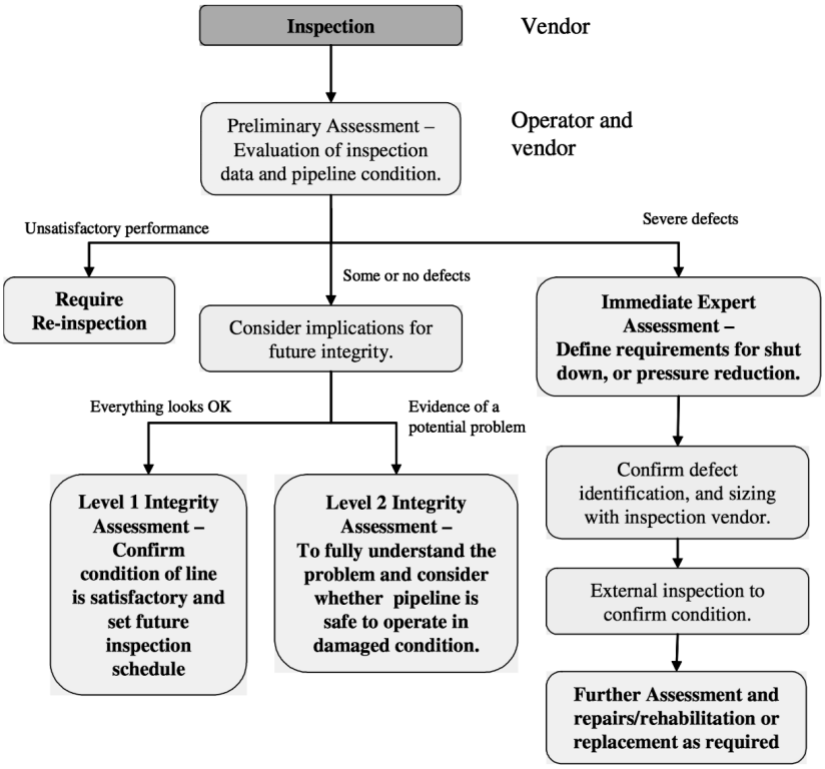

Figure 3: Process of assessment using intelligent pigging[10]

Although this process describes a very detailed method of assessing pipeline integrity, it is backed by a set of standards that act as a reference point in determining pipeline health. There are several standards that are widely used internationally such as the ASME B31G[11], API 579 FFS-1[12] and the DNV RP-F101.[13]

These standards utilise a set of complex mathematical formulas to calculate the failure pressure of a pipeline to determine the remaining strength of the line. It requires the measurement of the defect (defect profile) and several other important parameters such as the flow stress, also known as the failure stress. The flow stress of a pipeline can be approximated by the

ultimate tensile strength of the pipeline. In essence, the flow stress describes the role of the material of which the pipeline is made of.[14]

2.2.1.1 ASME B31G

The ASME B31G is a widely used method introduced by the American National Standards Institute to assess the remaining strength of a pipeline. Currently at its 3rd edition, the method utilises complex mathematical formulas to calculate the failure pressure of a pipeline based on its health that is obtained from the pigging process, as seen in equation 1.

$$P = \frac{\sigma_{flow} \cdot 2 \cdot t}{D} \left[\frac{1 - 0.85 \cdot \frac{d}{t}}{1 - 0.85 \cdot \frac{d}{t} \cdot \frac{1}{M_3}} \right] \quad (1)$$

The first two editions were introduced in 1984 and 1991 respectively and the main difference between all three is the modification in the formula to make calculations less conservative as compared in Figure 4.

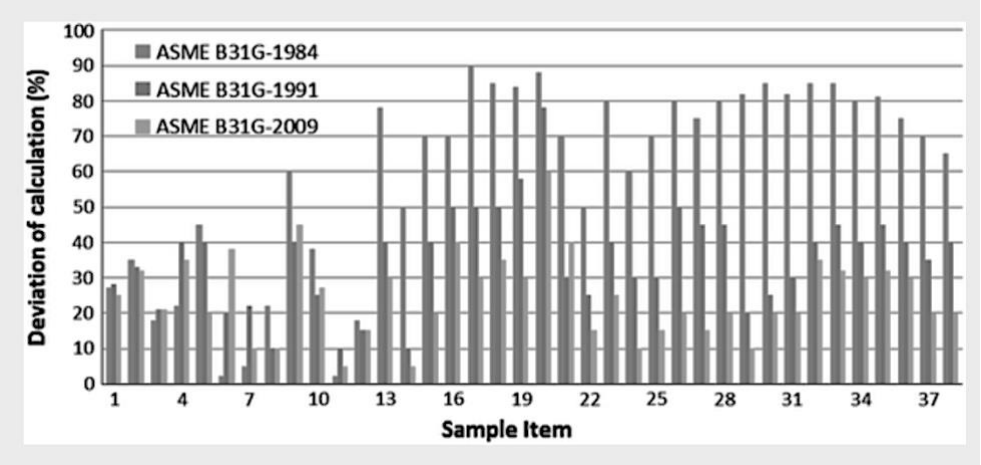

Figure 4: Comparison of results between all three ASME B31G editions[15]

2.2.1.2 RSTRENG

The RSTRENG method is essentially an improved version of the B31G in terms of conservatism. In B31G, the defect profile in the data for calculation is taken through a rectangular or parabolic assumption of the defect size. In other words, it does not take the exact measurements of the defect size, and therefore yields a fairly conservative result.

The RSTRENG method is introduced to make calculations by taking into account the exact defect profile from the data. Therefore, a slightly different calculation method is used as shown in equation 2, and a more accurate result is achieved.

$$P = \frac{\sigma_{\text{flow}} \cdot 2 \cdot t}{D} \left[\frac{1 - 0.85 \cdot \frac{d}{t}}{1 - 0.85 \cdot \frac{d}{t} \cdot \frac{1}{M_3}} \right] \quad (2)$$

It is noticeable that the formula for both the RSTRENG and B31G methods are the same. The difference is actually in the calculation of the flow stress, σ_{flow} in both formulas. Table 1 shows the basic difference between all formulas including B31G 1984, 1991 as well as the RSTRENG effective area. These techniques are not discussed in detail as it is reasonably irrelevant for the progress of this research. SMYS is the specific minimum yield strength of a pipeline.

Criterion	B31G-1984	B31G-1991	B31G-2009	RSTRENG 0.85 dL	RSTRENG effective area
Flow stress	1.1*SMYS	1.1*SMYS	1.1*SMYS	SMYS + 68.95 MPa	SMYS + 68.95 MPa
Defect area	dL or 2/3 dL	dL or 2/3 dL	0.85 dL	0.85 dL	Effective area
Value of L^2/Dt	20	20	50	50	50
Kinds of M	1	1	2	2	2
Kinds of failure stress	2	2	1	1	1

Table 1: Comparison between ASME B31G 1984, 1991, 2009, RSTRENG and RSTRENG effective area for failure pressure calculation in pipelines[16]

2.2.1.3 DNV RP-F101

Det Norske Veritas, DNV together with BG Technology developed the RP-F101 standards based on their Joint Industry Projects (JIP). Equation 3 shows the formula for failure pressure using this method.

$$P = \frac{2t\text{UTS}}{(D-t)} \frac{\left(1-\frac{d}{t}\right)}{\left(1-\frac{d}{tQ}\right)} \quad (3)$$

It is noted that the flow stress from the B31G and RSTRENG had been replaced with ultimate tensile strength (UTS) and the bulging stress magnification factor, M had been replaced with the length correction factor, Q as seen in equation 4.

$$Q = \sqrt{1 + 0.31\left(\frac{l}{\sqrt{Dt}}\right)^2} \quad (4)$$

The difference in formula and calculation methods gives a higher failure pressure through the RP-F101 compared to the B31G method, therefore making the RP-F101 less conservative. A higher calculated failure pressure simply means that the pipeline is more usable since the pressure at which it breaks (fail) is not too low. This accuracy in calculating failure pressure makes a huge difference for operators in terms of increasing productivity and reducing loss due to a conservative calculation.[17]

2.2.1.4 Comparison of Standards

In Table 2 , it is seen that while DNV RP-F101 is seemingly a better standard to use due to the level of accuracy in calculation as well as direct benefit in enabling longer pipeline lifespan and therefore increasing productivity, it is not a standard that is as widely accepted in the global industry as compared to the other two standards.

Criteria	ASME B31G	RSTRENG	DNV RP-F101
Influence	High	High	Moderate
Level of conservatism	High	Moderate	Moderate

Pros	Proven method Reduced risk	Improved method Manageable risk	Increased productivity and pipeline lifespan
Cons	Reduced productivity and pipeline lifespan	Limited scope of usage to specific use cases	Less global adoption as compared

Table 2: Comparison between the three standards discussed from section 2.1.1.1 to 2.1.1.3.

2.2.2 Magnetic Flux Leakage, MLF Pigging

At present, most of the in-line inspection methods using intelligent pigs are based on the magnetic flux leakage (MFL) technology. Basically, an axial MFL tool is used to induce a magnetic field along the pipe using powerful magnets to magnetize the pipe wall to saturation. The magnetic flux that is being carried becomes less in areas of the pipe that has reduced wall thickness, indicating a defect such a corrosion. This corresponds to a leak in magnetics flux in those parts of the pipe and hence the name of the technology. The tool the locates and records the leakage in the pipe and this data is further processed in a software to provide a detailed information on the type and location of the defect.[18]

This information is then used to calculate the failure pressure and remaining strength factor using the standards such as ASME B31G, RSTRENG or DNV RP-F101. While MFL tools can be used with or without a medium in the

pipeline such as any form of liquid or even the product itself, they are considerably heavier and larger than Ultrasonic, UT tools. However, MFL tools as seen in figure 5 are able to record data even at higher speeds compared to UT tools, and are also considered the cheaper option.

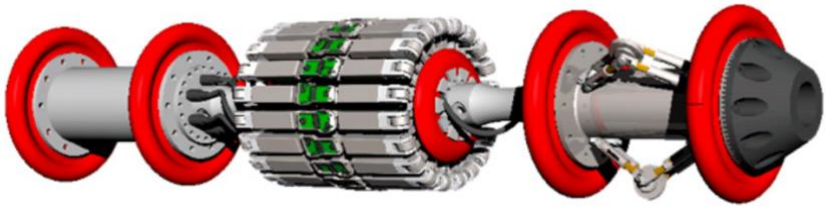

Figure 5: Magnetic Flux Leakage in-line inspection tool (intelligent pig).

2.2.3 Remote Field Testing, RFT Pigging

Remote Field Testing is another technology that is also popular among pipelines that are lined with cement, epoxy or polyethylene. These are pipelines that cannot have MFL or other types of tools running since the pipeline is lined with a different material and will severely interfere the readings. RFT does not require contact with the pipeline while scanning for data and therefore becomes a good option for such lines.

A RFT tool as shown in figure 6 works a little differently from the MFL as in this case it uses an AC electromagnetic field that is generated by the tool and this field is attenuated and delayed in the tool. Sensors on board the tool will then look for reduction in time delays which indicate metal loss in the pipeline. Once again, a software is used to interpret the data into a complete wall thickness profile for further processing and failure pressure calculations.[19]

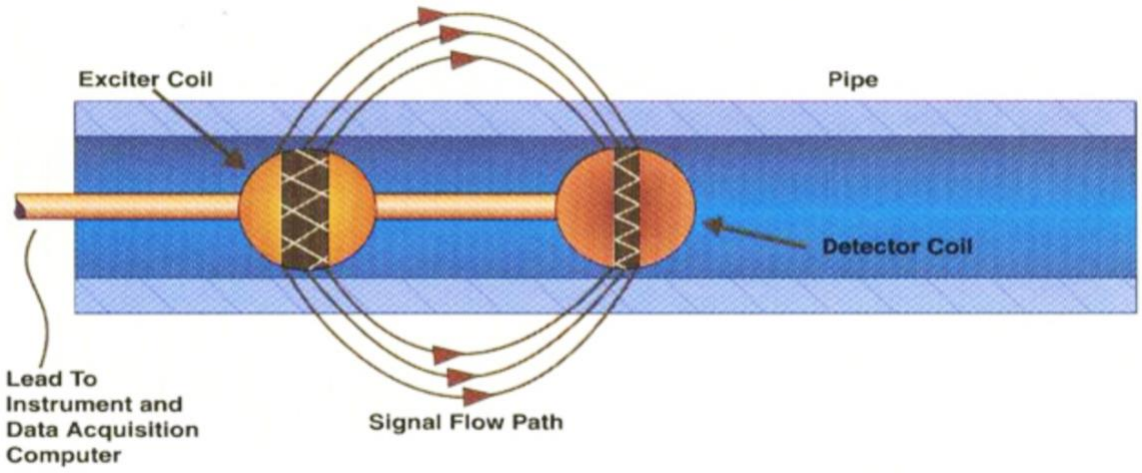

Figure 6: An RFT tool functioning within the pipeline.

2.2.4 Ultrasonic, UT Pigging

Since this research focuses on the use of ultrasonic sensors (transducers) for data collection, it is worth spending more time discussing this particular technology in intelligent pigs. UT tools have transducer on board the tool that fire up ultrasonic signals towards the pipeline. Since UT requires a medium to travel, these tools are usually run along with the product or in any liquid medium as seen in figure 7. The time taken for the signal to return the tool gives an indicator to measure the wall thickness of the pipeline and from thereon, a full pipeline profile can be obtained for analysis as seen in figure 8. UT tools are highly effective as they are very accurate in their readings when compared to MFL tools.[20]

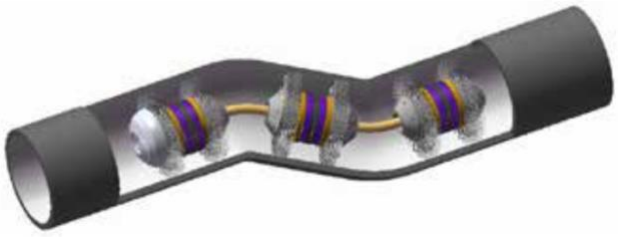

Figure 7: An ultrasonic tool navigating through a pipeline.

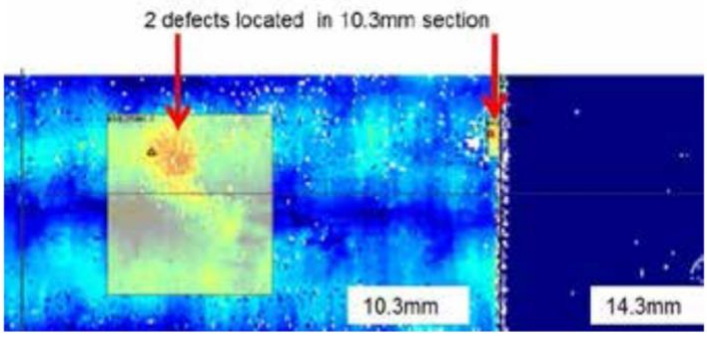

Figure 8: An example of data that has been processed by software for fitness for service assessment.[21]

Ultrasonic tools are the current upcoming technology for intelligent pigging. It's hardware setup enables the tool to be designed to be very versatile and flexible, making it the best and sometimes the only options for pipelines that are complex in design. These types of pipelines are referred to as "unpiggable pipelines" in the industry and are generally referred to lines that were built many decades ago where its design information had been lost. Operators cannot take the risk of having a tool run through it in the fear of having the tool stuck in the line. Unpiggable pipelines also refer to lines with very tight bends where a pig can get stuck, and also pipelines that were built with dual diameters in segments.[22] UT tools can be designed to be small enough to navigate through these unpiggable pipelines and

therefore it is becoming one of the most sought-after technologies over MFL for in-line inspection in recent times. However, all these technologies are more focused on hardware development that constantly improves the data collection process. They are more focused on increasing the resolution of the data which is the first part of the entire assessment process. In an industry such as the oil and gas, time is literally a pot of gold. Loss of time literally translates to a heavy loss of revenue and therefore, mistakes can be very costly. No matter how hi-tech these tools are, the process of data collection through an inspection run is still bound to human errors as well as the potential biasness in determining a defect through visual inspection, which is considered a level 1 assessment. Therefore, this research aims to solve this by implementing artificial intelligence techniques to classify defects at a fraction of a time while providing a real time assessment to operators through an on-site installation of a hardware implementation of the particular artificial intelligence algorithm.

2.2.5 Comparison of Intelligent Pigging techniques

Table 3 summarizes the pros and cons between different intelligent pigging inspection techniques.

Criteria	Magnetic Flux Leakage	Remote Field Testing	Ultrasonic Transducers
Pros	Widely used	Works well for long distance	More accurate

	Works for long distance and large pipelines	Works with non-metal pipelines	Able to inspect small pipelines
Cons	Not suitable for small pipelines Only used for metal pipelines	Bulky Costly	Only used for metal pipelines

Table 3: Comparison between intelligent pigging techniques

2.3 Machine Learning Classifiers

There are several ongoing researches in the study of discovering new techniques with the goal of improving pipeline safety and structural integrity in whole. The goal is to basically keep looking for ways to make lines as safe as possible in the most effective and efficient way. Therefore, in-line inspection should not be the only method such improvements are made on. A diverging perspective is required with other forms of hardware technology coupled with new mathematical techniques.

From what we have known till to-date, all pigging tools have 1 thing in common, which is the acquisition of pipeline wall thickness data. Using this data, mathematical algorithms can be applied to process it and further understand it. As the popular saying goes “Data never lies”, study can be done to understand data in a non-visual perspective.

Perhaps the most relevant mathematical algorithms in this case are essentially machine learning classifiers due to their ability to produce high

accuracy classifications and predictions with minimal to no supervision. There are many different types of machine learning algorithms that exist for classification purposes. This section will briefly cover some of the popular algorithms that are used in general applications worldwide. The choice of classifier of this research will be discussed in the following section 2.3 after a run-through on other non-NDT methods of failure predictions.

2.3.1 Naïve Bayes Classifier

The Naïve Bayes classifier is based on the Bayes theorem by Rev. Thomas Bayes[23] and is basically built on the conditional probability. Conditional probability is the probability that a particular event will take place if another event had already taken place. Therefore, the training process involves the computation to yield the result that carries the highest probability which makes the classification in a given problem. Equation 5 shows the algorithm where *y* is the class variable and x_1 *to* x_n is the dependent feature vector.

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)} \tag{5}$$

There are a few other derivatives of the Naïve Bayes classifier namely the Gaussian Naïve Bayes, Multinomial Bayes and the Bernoulli Bayes classifiers. Each of them hold a slightly different technique meant for different data distribution types. Although very useful and simplified for general classification purposes, the Naïve Bayes classifier falls short in handling data sets of high dimensions.

2.3.2 K-Means Clustering Classifier

K-means is a technique that clusters data based on pre-defined initialisation centroids. The algorithm corrects itself until the most suitable centroids are found and the resulting clusters are considered the labels in the data set. It can be understood that K-means is a technique that works well for unsupervised datasets in which labels need to be identified based on the data. Equation 6 describes K-means where a set of *N* samples *X* is divided into *K* disjoint clusters *C* by the mean μ_I (also known as centroids) of the samples in the cluster.

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_j - \mu_i||^2) \tag{6}$$

The drawback of this technique for this research is that the K-means algorithm will work to create clusters in any given data set. In a pipeline dataset for instance, if there are no damages to the pipe, there will be no standout discrepancies in the dataset, but if we use K-means, the algorithm will still show clusters. These clusters are essentially noise as there were no damages on the pipeline to begin with. This will greatly affect the goal of the research.[24]

2.3.3 Linear Regression

The linear regression algorithm is a very popular one that is widely used in many applications. Equation 7 shows the algorithm where *y* is the dependent variable and *X* is the independent variable.

$$\min_{w} ||Xw - y||_2^2 \tag{7}$$

However, its simple nature although makes it very easily interpretable, makes it relatively incompetent when handling datasets with large number of variables. This is because the linear regression algorithm works best with 2 variables in which 1 is a dependent variable that depends on an independent variable. As this research deals with a non-linear dataset, the linear regression is not suitable for this application. Due to the need of the application in this research, the linear regression algorithm is far too inferior to be used here.[25]

2.3.4 Logistic Regression

The logistic regression algorithm as seen in equation 8 may seem very related to the linear regression algorithm. However, it works through simple logic in which the resulting outcome is always a fixated option rather than a continuous one that a linear regression may yield. For instance, a linear regression can be used to predict how much of sales a particular company can make in the upcoming month based on sales data from preceding months. But in a logistic regression, the possible outcomes would be whether or not sales can be generated in the following month.

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1). \tag{8}$$

This research deals with prediction failures as a continuous outcome in terms of bad a failure could be, instead of simply discovering whether or not a failure can occur. For this reason and the fact that logistic regression does not work well with high dimensional datasets, it is deemed not suitable for the application.[26]

2.3.5 Artificial Neural Networks

Artificial Neural Networks, ANN is probably the most popular algorithm known among the general public when it comes to talks on artificial intelligence. It is also perhaps one of the easier algorithms to be visualised by a lay person with zero exposure on Artificial Intelligence, AI. ANN works like the human brains in which nodes and hidden layers compute to yield the outcome. ANN is also the stepping stone of the now very prolific Deep Learning. Figure 9 shows the working structure of a neural network where the left layer is the input layer that hosts the input features. The middle layer is the hidden later that carries the weight of each input in influencing the end outcome which is at the right, the output layer. ANN works by optimising the weights in the hidden layer until a desired accuracy is obtained.

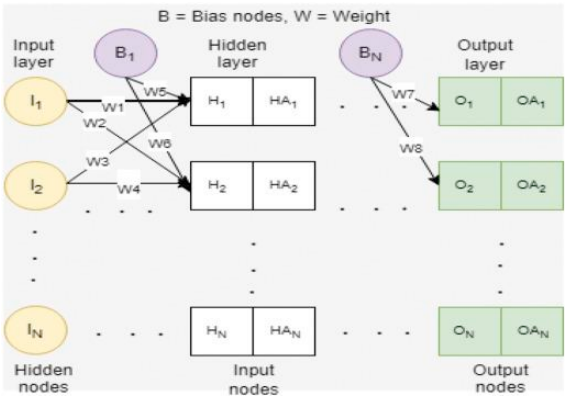

Figure 9: Working structure of a neural network[27]

Although it is possible to use ANN for this research, it is more well suited for applications that require greater work such as image recognition and etc. For this research, the algorithm does not really require the power of an artificial neural network.[28]

2.3.6 K-Nearest Neighbors

Sounding similar to the K-means clustering algorithm, the K-nn algorithm uses a similar approach to classify datasets. It is also good in handling data with large number of noise. However, it is not easily interpretable and has a limited contribution to this research.[29] [30]

2.3.7 Support Vector Machine

Support Vector Machine is a very robust algorithm that works in a very straight forward way by using a hyperplane to draw a line of separation within a data set for classification purposes. In the event of a non-linear data set, it utilises kernel functions to separate data of high dimensions. SVMs are fairly easily tuned and are versatile and adaptable for applications

that may differ slightly within the same use case. In the case of this research, the SVM can be combined with a preliminary clustering technique such as the K-means technique to then effectively classify new data. This will be further explored in the following section 2.3. Figure 10 shows a Support Vector Machine hyperplane separating two data classes.

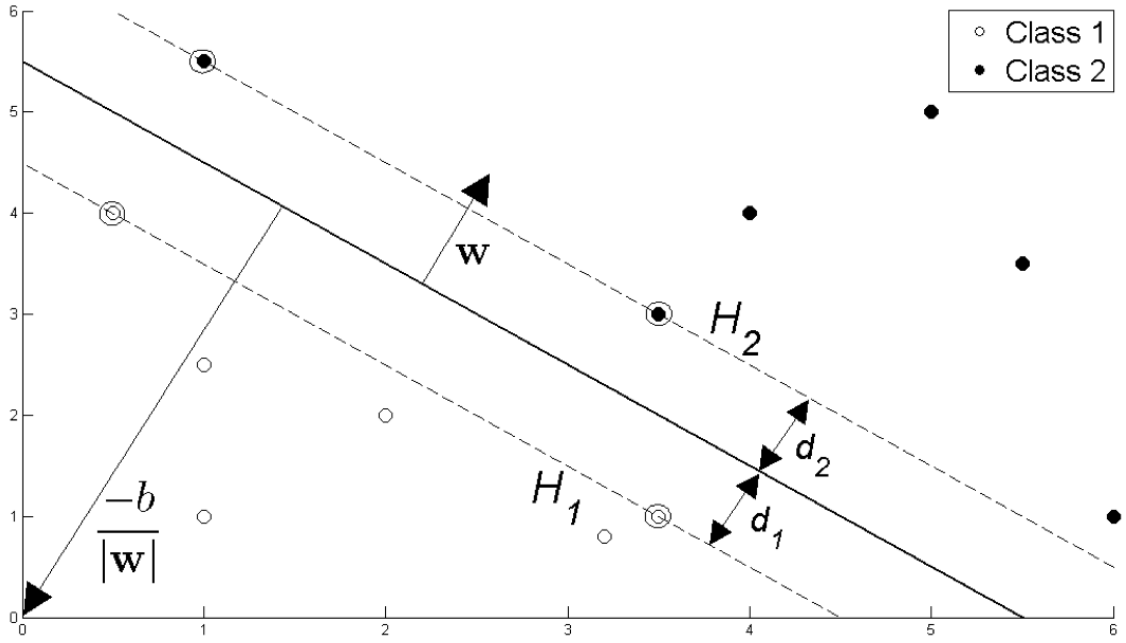

Figure 10: Support Vector Machine hyperplane[31]

Its robustness and ability to handle datasets that nonlinear and high dimensional makes it the most suitable algorithm for this research.[32]

2.3.8 Comparison of Machine Learning Classifiers

Based on the comparison in Table 4, Support Vector Machines is the preferred algorithm for this research due to its ability to handle both nonlinear and high dimensional datasets despite its limited usability in other

applications. Support Vector Machines along with other algorithms is further discussed in section 2.3.

Classifiers	Pros	Cons
Naïve Bayes	Widely used for general classification purposes	Can't handle high dimension datasets
K-Means Clustering	works well for unsupervised datasets	Creates clusters in any given dataset (not needed for pipelines)
Linear Regression	works best with 2 inter-dependent variables	Cant handle nonlinear datasets
Logistic Regression	Gives outcomes based on yes or no	Cant handle high dimension datasets
Artificial Neural Networks	Widely usable	Too powerful for the this research
K-Nearest Neighbors	good in handling data with large number of noise	not easily interpretable
Support Vector Machines	Able to handle nonlinear and high dimensional datasets	Limited use cases

Table 4: Comparison between machine learning algorithms

2.4 Raspberry Pi

The Raspberry Pi as seen in Figure 11 is perhaps one of the most popular mini computers in the market especially among hobbyists. It is a credit-card sized Central Processing Unit (CPU) that can be connected with a monitor, keyboard and mouse just like an ordinary full sized CPU. The Pi is more than just a microcontroller like the STM board or the Arduino. It has a System On Chip (SOC), graphics processing unit (GPU), is easy to use especially on the Python programming language making it especially good machine learning projects.

Figure 11: Raspberry Pi

2.5 Research Gaps and Research Focus

2.5.1 Research Gaps

We can observe that the data that comes from intelligent pigs clearly show the types of defects with all its accompanying attributes. It corresponds to a highly nonlinear dataset with many defects categories. This is considered as a complex classification problem. However, there are mathematical theories that are popular in other applications currently that may well be suited for this task i.e. Support Vector Machines and K-Nearest Neighbors (KNN). Through a supervised learning technique, the data set is first clustered into relevant defects to produce labels which can then be used as the benchmark for future classification by the SVM.[33]

Another research that provides an alternative to intelligent pigging is by utilising inductance coils that are wrapped around a damaged section of the pipeline. The pipe will behave as a magnetic conductor of a transformer eddy-current transducer. The electromagnetic force (EMF) induced is based on the pipe parameters and thickness. In essence, the change in EMF value will be compared to accumulated data of EMF values to show a defect or loss of wall thickness.[34] This method could be considered theoretically plausible but not practical as firstly it does not give the wall thickness profile of the entire pipeline but only the previously identified defected area. This is very limiting and furthermore, this method of assessment does not provide real time assessment privileges which is the aim of this research.

There is also work being done to improve the accuracy of defect detection from in-line inspection data. The Bayesian approach is used for calibrating and estimation actual defect depth for pipelines that are buried underground. A Gaussian mixture model, GMM, Bayesian information criteria, BIC and expectation-maximization, EM techniques were used to as clustering algorithms to further improve defect classification accuracy.[35] However, this method still requires an in-line inspection tool to provide the data in the first place and also does not fulfill the real-time assessment aim.

Another research aims to use an inverse Gaussian process-based model to characterize the growth of defect depts. As it is modeled in a hierarchical Bayesian framework, it evaluates the probability distributions of unknown parameters by intergrating the previous knowledge of the same parameters. Using this information and the hierarchical framework, the growth of a metal-loss defect could be mathematically predicted.[36]

The more relevant techniques that plays the role of a foundation to this current research would be the use of support vector machines to detect defects and predict them. A research done in 2009 serves as the base for this where piezoelectric sensors that are placed along the surface of the pipeline gives raw sensor signal data that will then be processed using the discrete wavelet transform (DWT) technique before being sent to train the SVM for classification purposes. This research was successful as it could give room for growth in real-time defect prediction assessment of a pipeline.[37]

In 2012, the research was further built on with the experiment with the use of long range ultrasonic transducers (LRUT). These sensors are to be permanently positioned on the pipeline to yield of real-time data for processing and decision making. Since in-line inspection was a periodical technique of assessment, there could be possibilities of unforeseen defects occurring on the pipeline in between inspection schedules that could potentially lead to a disaster. This LRUT technique serves to overcome that issue through continuous monitoring systems. The problem was that a continuous monitoring system would need an intelligent software that could make decisions automatically based on the data. Support vector machines was the method used thus far as the machine learning algorithm to serve this purpose. However, accuracy of SVM is very dependent on the choice of kernel and soft margin parameters and this was not an easy task to be automated. To overcome the need of having to optimise the SVM with relevant kernels at a given time, an Euclidian approach was employed and since it is not dependent on choice of kernels, it was able to optimise the SVM as much as it could have been by manually adjusting kernel parameters. This research served as a good benchmark for a real-time defect prediction system.[38]

Then in 2013, the research continued with the use of Kalman filters to filter out noise to make defect classification by the SVM more accurate. This research further optimised the SVM classifier into being able to detect defects at different depths and even differentiating closed spaced defects with a high accuracy.[39]

This method of defect classification and failure prediction became a very strong competitor to in-line inspection except for the hardware implementation struggles that it may have for actual deployment. As much as the software functionality was up and running, the hardware setup to actually install LRUTs onto a pipeline and have an onboard processer that handles the entire task of processing data and classifying defects is still far from completion. Therefore, this research continues to this point where an implementation of SVM on to a hardware setup through Field Programmable Gate Arrays, FPGA simulations and a possible integration of a Raspberry Pi is continuously studied and documented.

There are also other researches being done at present to investigate the implementation of a real-time, unsupervised machine learning technique onto a hardware setup so that it could be a standalone unit that performs all necessary functions. This research is aiming to achieve the same thing.

There was a research to implement neural network models for adaptive control of a robot arm movement. The reason this research is considered as real-time is because what is trying to be achieved is to control the arm movement trajectory during visual guided reaching. Visual data of an object is being fed into the neural network real-time and unknown obstacles or positions are being discovered making it an unsupervised learning. The neural network has to correct the trajectory in real time and enable the arm to achieve its target.[40]

Another research was done in 2013 to train a neural network to be able to detect pedestrians in real-time. As features coming in the visual feed is definitely unsupervised, this considers as a similar initiative to this research as it aims to have a standalone unit. It was noted in this particular research that near real-time speed was achieved using parallel hardware but with improved graphic cards, complete real-time, unsupervised learning could be achieved.[41]

A research using deep learning and reinforcement learning was also done to improve the Atari game playing experience. By using deep learning methods in a real-time gaming environment, their goal was to build better artificial opponents for game players.[42]

Another similar research was done on a popular game StarCraft:Broodwar with the aim of using reinforcement learning algorithms in an unsupervised manner during real-time gameplay to replace previously used non-adaptive, deterministic methods.[43]

A research done in 2016 aims to quantify the difference between the deep learning algorithm and support vector machines in a waste sorting task. The relevance of this work to this research is the employment of the Raspberry Pi to host the classifier. Although training was done beforehand, the classifier is then implemented on the Raspberry Pi where it produced very fast classification.[44]

At some point, research was done to understand deep learning a little better as an advanced algorithm that is grown from neural networks. Through multiple procession layers, the deep learning model has massively improved pretty much everything from speech recognition to classification and regression works.[45]

With deep learning on demand now, open source libraries like Tensorflow along with the Tensor Processing Unit by Google provides a platform for exceptional artificial intelligence work to be carried out. This algorithm could very well be useful for the future development from this current research as it can be very well used to yield exceptional results.[46] [47]

In terms of hardware, there was a research done in 2016 to investigate the potential benefits of the resistive switching memory (RRAM) as a hardware that can host machine learning algorithms. It was found that the use of RRAM contributed in reduction of power consumption with a very decent classification efficiency rate. This could be an area to look into in later phases of development in reference to this research.[48]

Perhaps the closest research to what is being worked on here is the one done in 2016 about a real-time anomaly detection framework for a NoC-based many-core architecture. The research implements a support vector machine algorithm along with the K-Nearest Neighbor (KNN) onto an FPGA board. It was found that the SVM adopted significantly well onto the FPGA. Most importantly, this work shows that support vector machines implemented on an FPGA could yield positive results.[49]

2.5.2 Research Focus

As for this research, the focus would be to utilise LRUT transducers mounted on a pipeline to transmit guided waves along the pipeline, and feed incoming data to a machine learning model hosted on a microcontroller onboard the pipeline to perform defect classification. The novelty of this current research really lies in the capability to perform real-time assessments all the time. LRUT transducers are chosen instead of intelligent pigs based on reasons in Table 5.

Inspection Method	Intelligent Pigging	LRUT
Accessibility	Inline inspection Limited to piggable pipelines	External inspection Can be placed anywhere
Cost	Relatively cheaper, however subjective to technical challenges Requires pipeline to be shutdown	Cheaper if permanently placed on the pipeline, minimal technical challenges Pipeline does not require shutdown
Time	Periodical	Continuous, real time
Range	Unlimited	Unlimited
Accuracy	Dependent on various factor i.e. product flowrate etc.	High accuracy as sensors are not on the move and independent of product flowrate

Table 5: Comparison between Intelligent Pigging and LRUT.

For this research, the Support Vector Machine classifier algorithm has been chosen mainly due to its ability to handle high dimensional data. In any given infrastructure within the energy industry, there are one too many factors that contribute towards its disintegration. Depending on the installation sites, these factors can vary, and therefore the classifier must be able to handle a high dimension dataset and be versatile to be tuned according to applications. The Support Vector Machine classifier has a range of kernels to choose from for tuning purposes. Therefore, due to its ability to handle a high dimension dataset and the versatility to be tuned according to applications, Support Vector Machines has been chosen as the classifier algorithm for this research.

The microcontroller chosen for this research is a Raspberry Pi due to its adaptabiity, versatility and ease of use in accommodating low data bandwidth.

2.6 Summary

The fact that machine learning has the potential to redefine pipeline integrity assessments based on the literature review conducted poses an exciting future ahead for the industry. The ability for the model to not just perform unsupervised classification of defects, but to also predict failures to a certain degree of accuracy is truly a game changer.

Chapter 3 will discuss the methodology behind this research, which is essentially divided into three segments that begin with the process of generating and transmitting signals into the pipeline wall, to be used for

the wall thickness analysis. Following to this would be the process of sampling and optimisation of the received signals, before finally sending them to the machine learning model for the purpose of training and testing.

This entire process will involve the integration of hardware and software as the process of generating and sampling the signals require signal generators, digital to analogue converters, amplifiers, and multiplexers to name a few. Therefore, chapter 3 will also discuss the process of setting up this integration, including the tuning of the components.

Chapter 3: Methodology

3.1 Introduction

This research work is aimed at replacing periodic methods of Non-Destructive Testing via inline inspection with a continuous monitoring system utilizing permanently placed LRUT transducers on the line combined with a Machine Learning model on a local CPU. A pipeline to run all tests on was already available as a continuation from Dr. Rajprasad's research.

LRUT transducers are fitted on collar and fixed around the circumference of the pipeline, and a signal generator hardware is setup to supply signals through the pipeline via the sensors. A signal sampling system is also setup to receive the incoming signals and convert them to digital data via MATLAB before sending the data to a machine learning model to be trained, tuned and tested on. As the effort was a combination of hardware and software components, the method of achieving this outcome was divided into three phases as shown in figure 12.

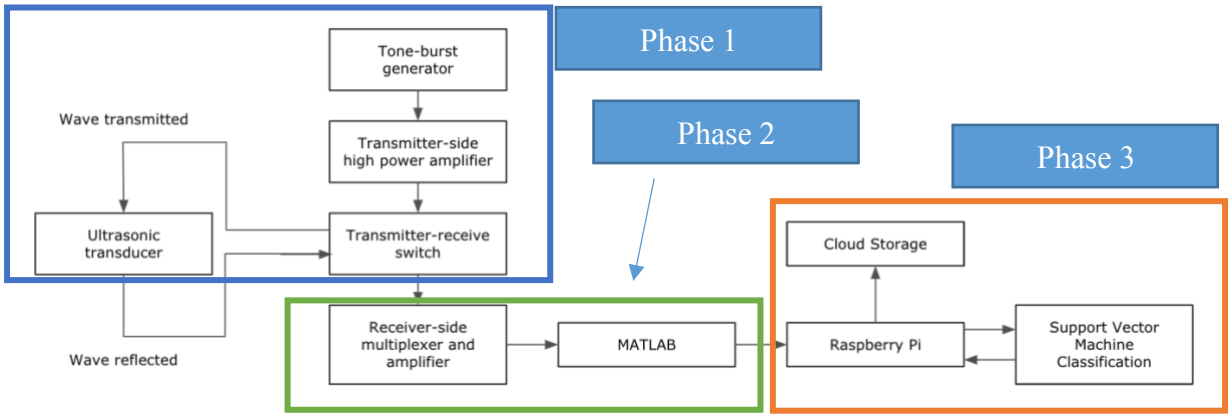

Figure 12: Three phases of research

3.2 Phase 1: Signal Generation and Transmission

The LRUT transducers require analog signals to be transmitted along the pipeline. These transmitted signals bounce back and forth between the two ends of the pipeline and is being received for analysis later on (discussed in section 3.2).

Producing high voltage analog signals in a laboratory requires a tone-burst generator, which is essentially a combination of a microcontroller and a digital-to-analog-controller (DAC), which is essentially an R2R ladder. This combination is such as a microcontroller produces digital signals, so the DAC converts this digital signal to analog signals to be sent to the LRUT transducers. However, the signal created in the microcontroller would have low peak-to-peak voltage i.e. $3V_{pp}$. Therefore, a high-power amplifier is used to amplify the analog signal that's coming from the DAC from $3V_{pp}$ to $300V_{pp}$ before sending to the LRUT transducers.

3.2.1 Tone Burst Generator

As mentioned previously in section 3.1, the tone-burst generator is a combination of a a microcontroller and a DAC. For the purpose of this research, several microcontrollers were considered with two specific models being compared seriously due to time constraints as seen in Table 6.

Microcontroller	Pros	Cons
Arduino Uno	Simpler setup	Simpler use cases

	Complete ecosystem available	
STM32F767ZI	More capable Higher clock speed	Higher cost

Table 6: Comparison between Arduino and STM microcontrollers

As it is crucial to have high processing speed to develop intelligence that is capable of near real-time performance, the STM microcontroller as seen in Figure 13 was chosen as it has a clock speed as low as 0.06μs compared to 4μs of the Arduino.

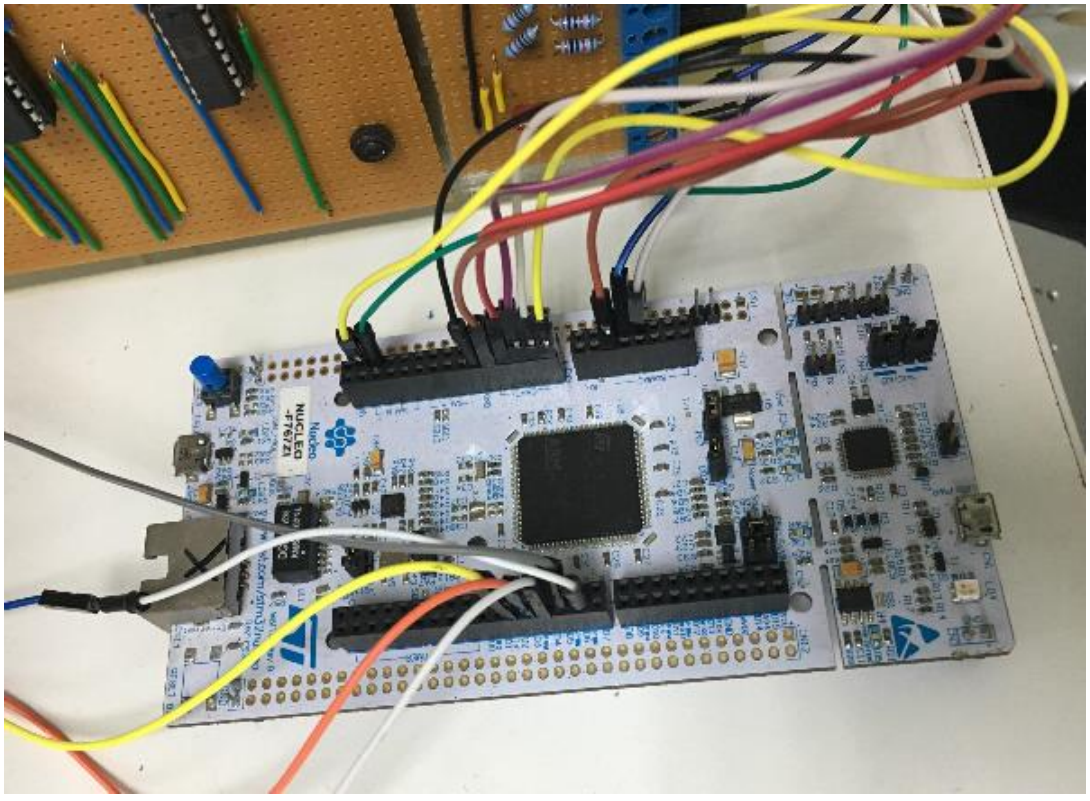

Figure 13: STM32F767ZI microcontroller

The sine waves generated from the STM microcontroller were attenuated by multiplying them by a attenuation factor to produce a wave with an increasing and decreasing amplitude with the peak amplitude in the middle. The 8 bits of discrete points were then ported to the DAC shown in Figure 14.

The DAC is a setup of 8 resistors to form a R2R ladder design that consists of 8 inputs to carry the 8-bit data of the signal that comes from the microcontroller.

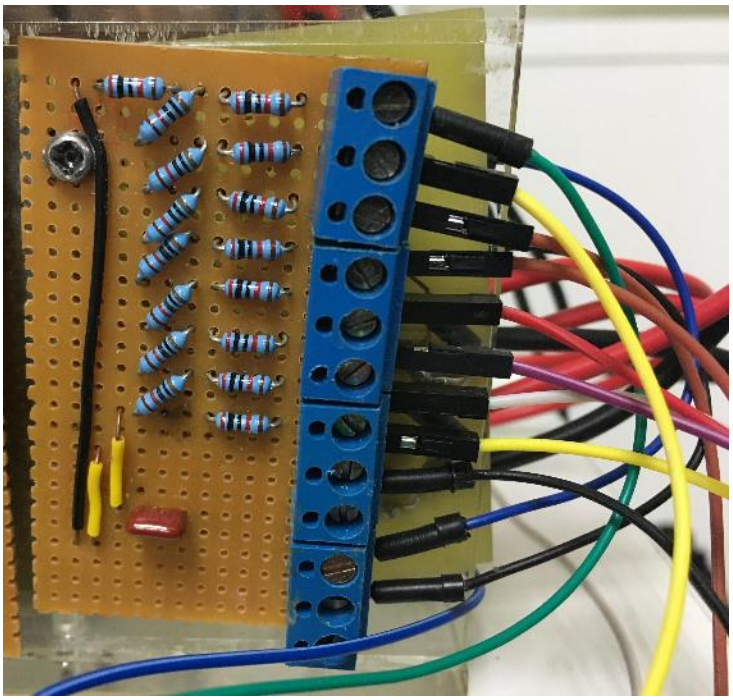

Figure 14: DAC with R2R ladder circuit

This signal that comes from the tone-burst generator is an analog signal with a peak-to-peak voltage of about 3V, frequency of 18.5kHz with 5 as the number of cycles. This signal as explained previously will have to be

amplified before being transmitted along the pipeline through the LRUT trandsucers.

3.2.2 Transmitter-Side High Power Amplifier

Ultrasonic transducers require high voltage to be activated, in this case 250V is required. As the signal generated from the tone-burst generator is not powerful enough, it has to be amplified through a high-power amplifier first before transmitted to the transducers.

The transmitter-side high power is made of a PA240cc operational amplifier which has an operating range of ±175V. It was configured to amplify the signal from the tone-burst generator by a 100 times from the original $3V_{pp}$ to $300V_{pp}$ to activate the ultrasonic transducers. Figure 15 shows the schematic of the transmitter-side high-power amplifier.

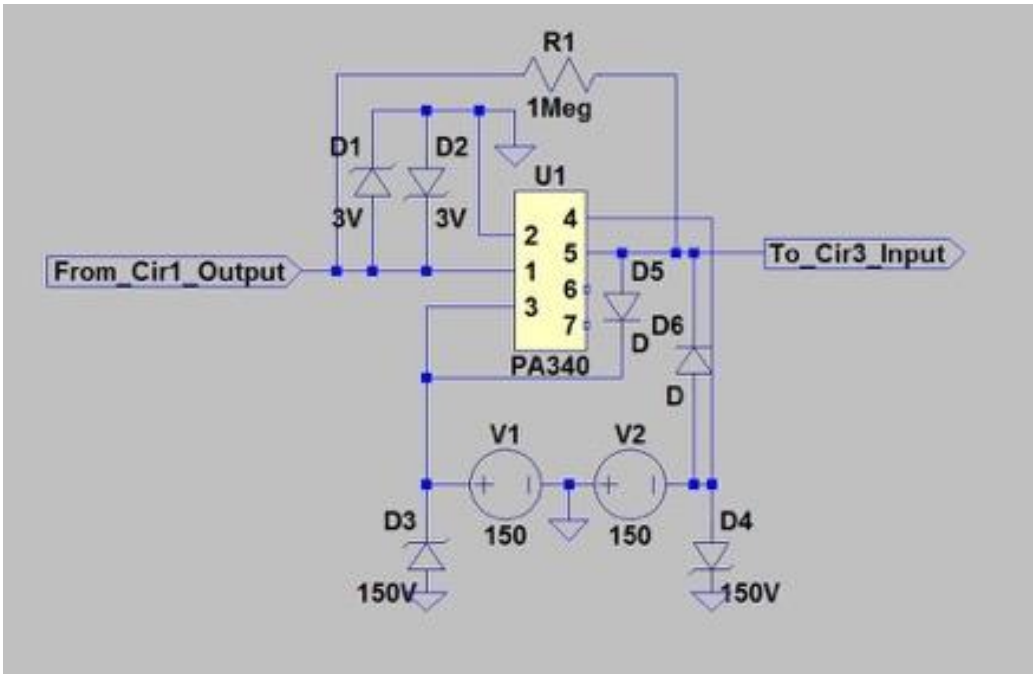

Figure 15: Schematic of transmitter-side high-power amplifier

As high voltages pose a potential hazard in operational circumstances, safety measures were taken which included using Zener diodes D1, D2, D3 and D4 based on the setup shown in figure 15 to protect against overvoltage. Ordinary diodes D5 and D6 were added to prevent the blowing up of input in the event the output became saturated. As seen in figure 16, operational amplifiers generally require a good heat dissipation system to prevent overheating, therefore high conductivity thermal paste of 4K/W was used as a medium between a metal piece that had the operational amplifier mounted on with copper poured onto it, to effectively dissipate heat.

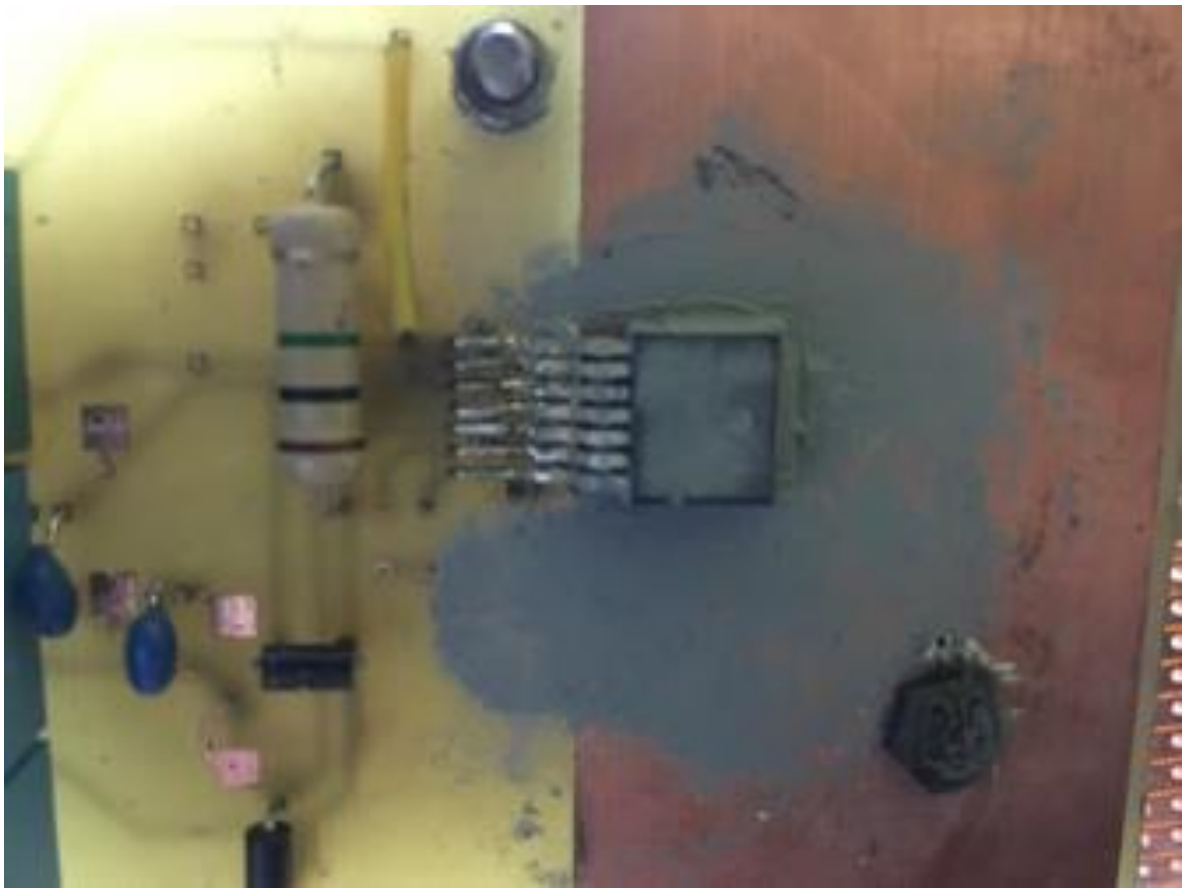

Figure 16: PA240cc Operational Amplifier with copper pour

The signals are now sent to the transmitter-receiver switch to be transmitted along the pipeline via the LRUT transducers.

3.2.3 Transmitter-Receive Switch

Since the same cable will be used for both transmitting and receiving of a signal, there has to be a switch in place to control the movement of flows to avoid both signals getting mixed up. As the transmitted signal carries a very high amplitude of $300V_{pp}$ the red circled part shown in Figure 17 is setup to prevent the signal from damaging the receiving end of the circuit.

Once the reflected signal returns, there is a likely possibility of a backflow of the signal heading back to the transmitting point. Therefore, to prevent this from happening, the green circle part of the circuit functions to block reflected signals from travelling back to the transmitter and eventually losing amplitude along the way.

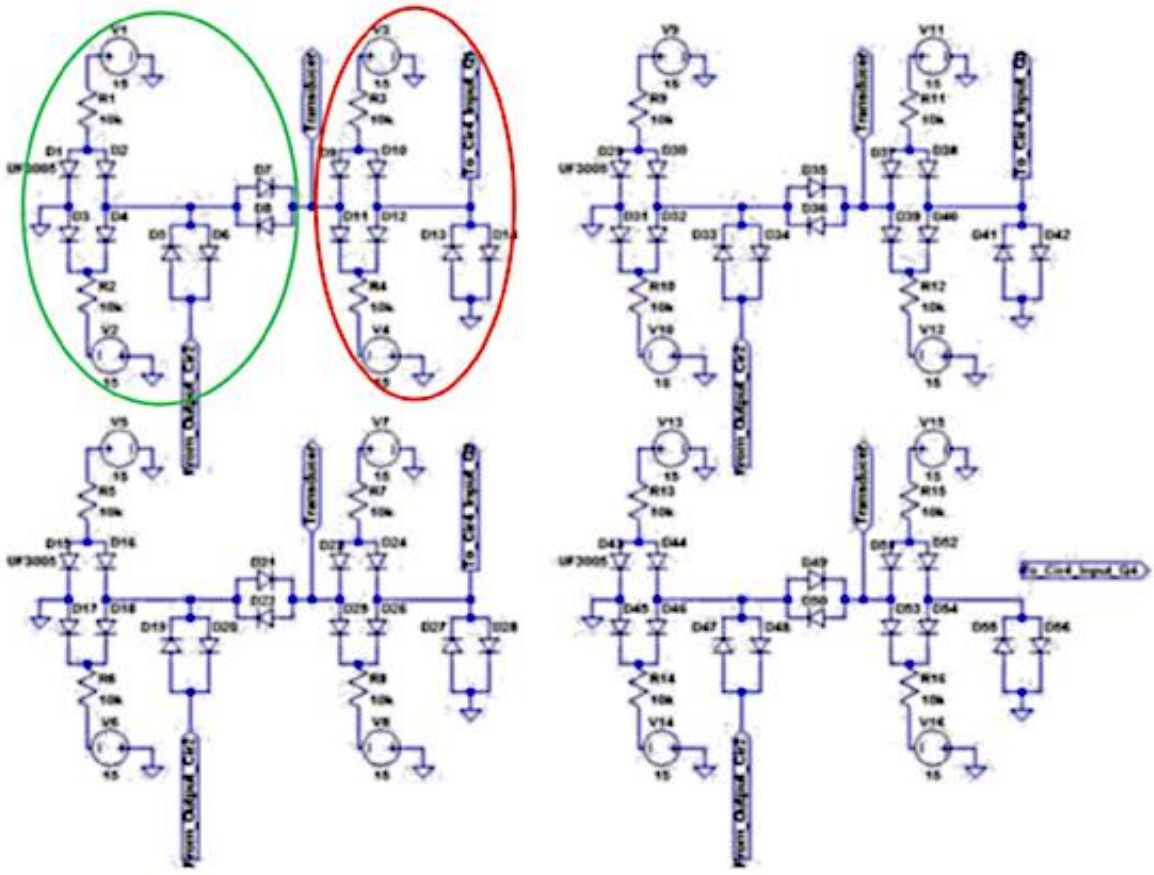

Figure 17: Transmit-receive switch schematic

Figure 18 shows the transmitter-receiver switch circuit with UF3005 ultra-fast switching diodes used due to its really fast switching time which is deemed highly important and useful for this setup.

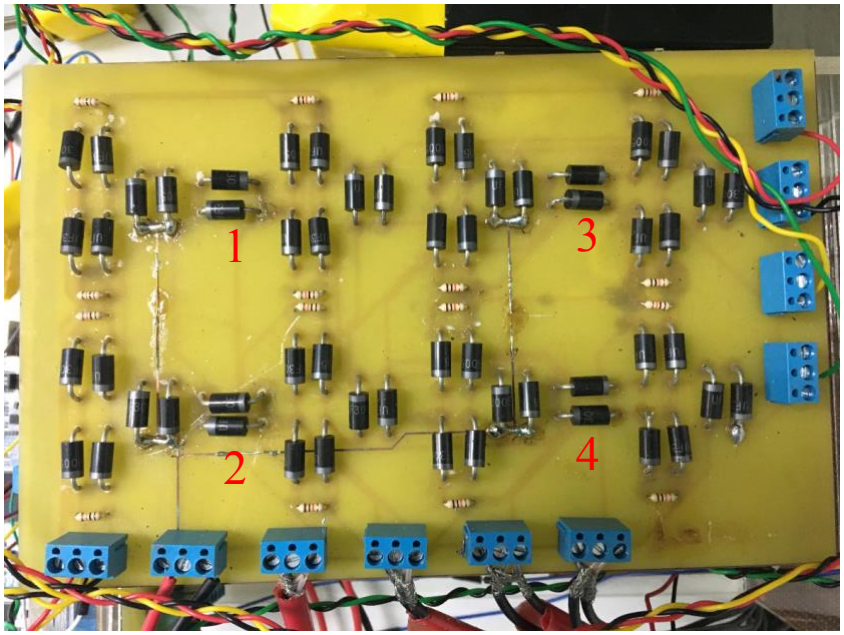

Figure 18: Transmit-receive switch circuit

The circuit is a 4-times repeated arrangement of diodes, with each part responsible for 1 quadrant of the ultrasonic transducers from the collar as seen in Figure 19.

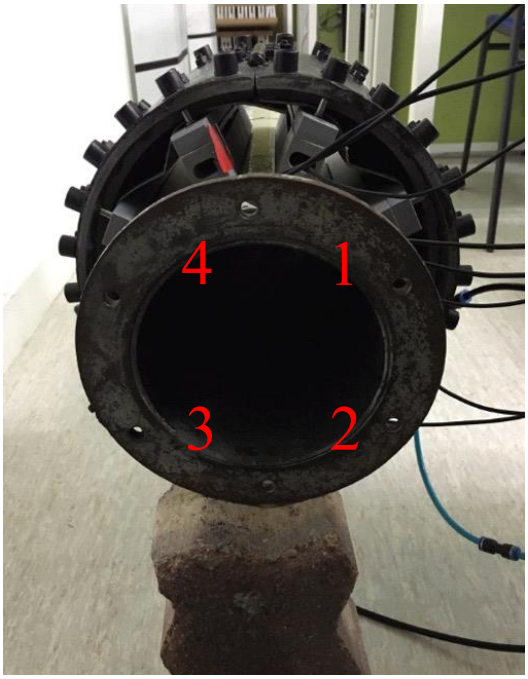

Figure 19: Oil and gas pipeline with quadrant number for transducer collars

3.3 Phase 2: Signal Sampling and Optimisation

The signals that have been transmitted in the pipeline will be received back in the same transmitter-receiver switch as explained in section 3.1.3.

The returning signal waves will now be sent to the multiplexers which functions to split the signal based on the quadrant to enable focused analysis.

3.2.1 Receiver-Side Multiplexer and Amplifier

As mentioned in section 3.2, the multiplexer acts as a data selector. The LRUT transducers positioned around the circumference of the pipe is divided into four sections, or four quadrants to enable focused analysis by quadrant. Therefore, three 8-to-1 multiplexers are used in a combination to accommodate this need. The multiplexers used are the ADH609 monolithic CMOS analog ones as seen in Figure 20, which has 8 input channels and 4 differential channels.

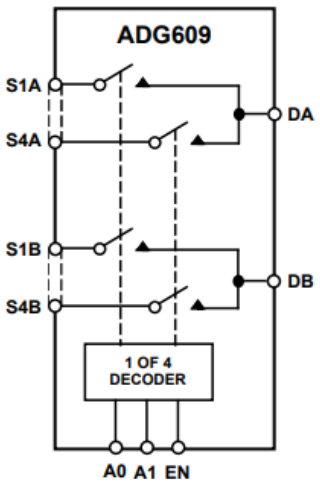

Figure 20: ADG609 functional block diagram

In Figure 20 that shows the functional block diagram of the ADG609 multiplexer, it is noted that there are 8 input channels consisting of S1A-S4A and S1B-S4B with 8 switches assigned to them. The control of switch is enabled when the enable pin is high. Therefore, outputs DA and DB can be determined by adjusting the states of A1 and A0, as shown in Table 7.

Enable (EN)	A1	A0	Switch Closed (Switch)		Output	
			S(x)A	S(x)B	DA	DB
1	0	0	S1A	S1B	S1	S1
1	0	1	S2A	S2B	S2	S2
1	1	0	S3A	S3B	S3	S3
1	1	1	S4A	S4B	S4	S4

** 1 = HIGH, 0 = LOW*

Table 7: Operation of ADG609

Figure 21 shows the circuit for the receiver side multiplexer, and Figure 22 shows the schematic that consists of a multiplexer setup on the left and an amplifier setup on the right with a line separating them.. The right side of the black line in Figure 22 which hosts the amplifier will receive the multiplexed output.

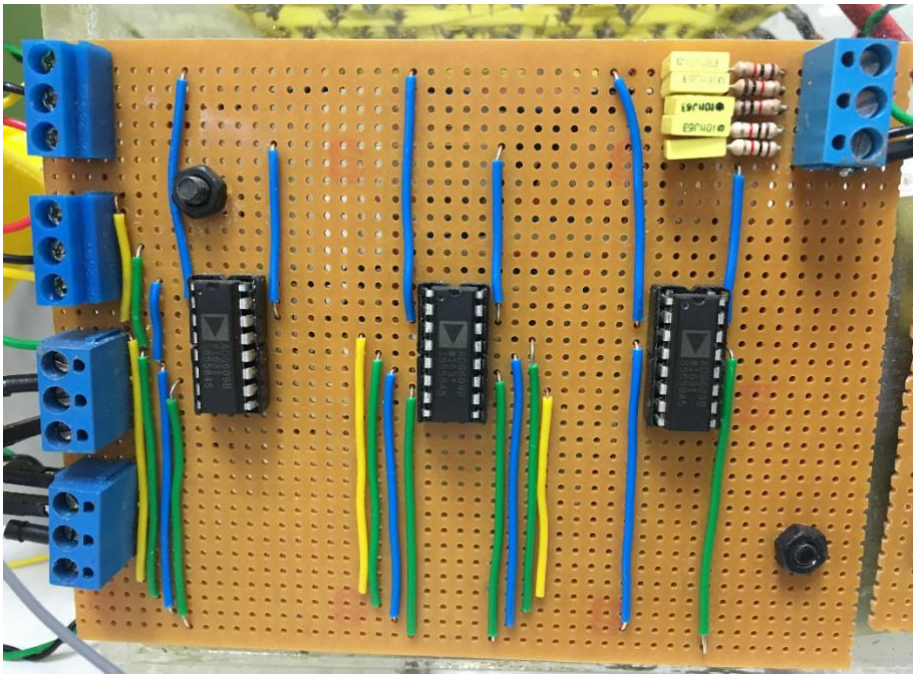

Figure 21: Receiver-side multiplexer circuit

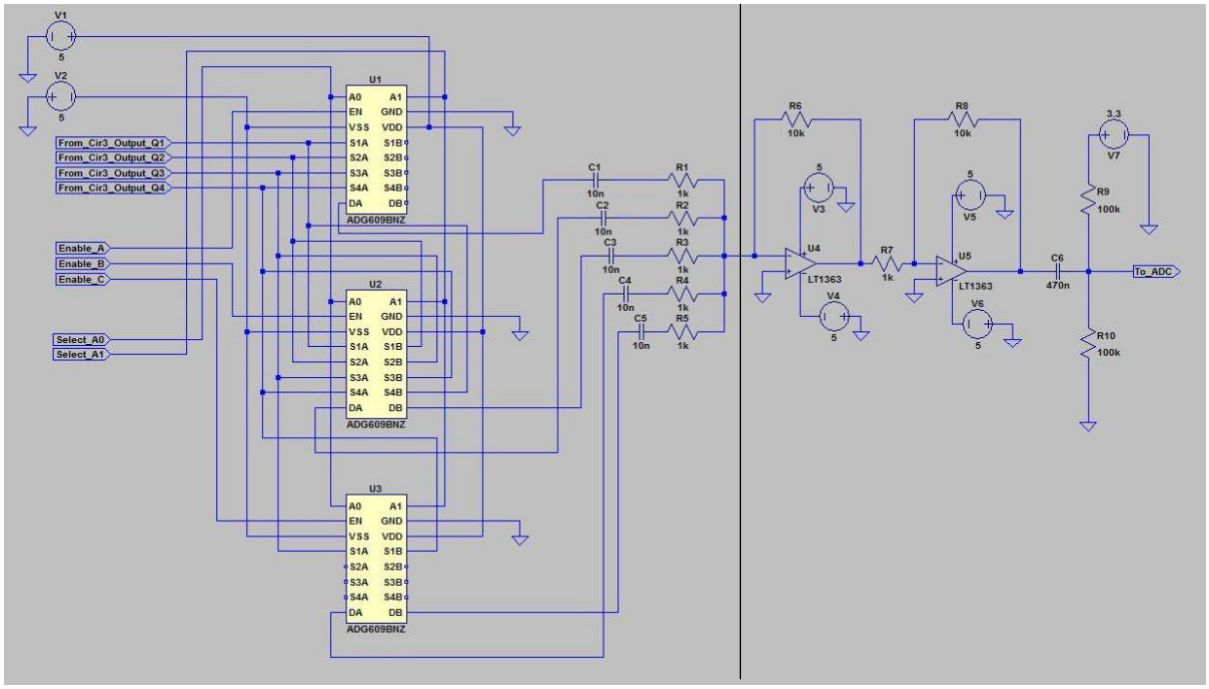

Figure 22: Receiver-side multiplexer and amplifier schematic

Table 8 shows all the connection points that are going into the multiplexer circuits. Enable pins A, B, C, A0 and A1 originate from the digital I/O ports of the STM32F767ZI microcontroller, which enables handlers to control the monitoring angle according to quadrant preferences on the pipeline setup. Input channels S1A to S4B in all three multiplexers receive signals based on the four quadrants.

Multiplexer U1		**Multiplexer U2**		**Multiplexer U3**	
Point	**Input**	**Point**	**Input**	**Point**	**Input**
VDD	5V	VDD	5V	VDD	5V
VCC	-5V	VCC	-5V	VCC	-5V
GND	Ground	GND	Ground	GND	Ground
A0	D2	A0	D2	A0	D2
A1	D3	A1	D3	A1	D3
ENA	D4	ENC	D6	ENB	D5
S1A	Quadrant 1 output	S1A	Quadrant 1 output	S1A	Quadrant 3 output
S2A	Quadrant 2 output	S2A	Quadrant 2 output	S2A	NC
S3A	Quadrant 3 output	S3A	Quadrant 3 output	S3A	NC

S4A	Quadrant 4 output	S4A	Quadrant 4 output	S4A	NC
S1B	NC	S1B	Quadrant 2 output	S1B	Quadrant 4 output
S2B	NC	S2B	Quadrant 3 output	S2B	NC
S3B	NC	S3B	Quadrant 4 output	S3B	NC
S4B	NC	S4B	Quadrant 1 output	S4B	NC

* NC = No Connect

Table 8: The connection of the points of the multiplexers

Table 9 shows the logic that determines the output. For example, if Enable A = 1, Enable B = 0, Enable C = 0, A1 = 1, A0 = 0, then the output is showing the signal from quadrant 3. I if Enable A = 0, Enable B = 1, Enable C = 0, A1 = 1, A0 = 1, then the output is showing the signal from quadrants 4 and 1.

Enable A	Enable B	Enable C	A1	A0	Output (Quadrant)
1	0	0	0	0	1
1	0	0	0	1	2

1	0	0	1	0	3
1	0	0	1	1	4
0	1	0	0	0	1 + 2
0	1	0	0	1	2 + 3
0	1	0	1	0	3 + 4
0	1	0	1	1	4 + 1
0	1	1	0	0	1 + 2 + 3 + 4

Table 9: The states of Enables, A1 and A0 with respective output

The selected output according to quadrant now has to be amplified as incoming signals are generally very weak after having loss energy during the transmission on the pipeline. As a recap, the signal was too weak to be transmitted in the beginning and therefore needed a high-power amplifier to increase its peak-to-peak voltage. Now, returning signals are also weak due to energy loss and will need to be amplified again using an operational amplifier for analysis.

2 LT1363 operational amplifiers (schematic shown in Figure 23) are used in the circuit due to its high speed of 70MHz and high slew of 1000μs/V, enabling it to perform well in DC conditions regardless of the capacitive loads. Therefore it is a good choice when it comes to setups that include lots of cable wires and buffers.

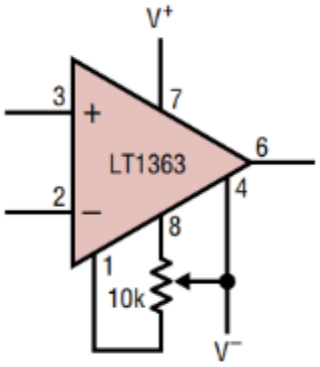

Figure 23: Schematic for LT1363 operational amplifier

The LT1363 operational amplifiers in Figure 24 is arranged in a cascading effect format with each having a feedback resistor of 10kΩ and input resistance of 1kΩ, resulting in a gain of 10 times. The cascading effect of both the op-amps with a gain of 10 will result in a total gain of 100 resulting in the multiplexed output being amplified by a factor of 100. The green box in Figure 24 shows the operational amplifiers on the circuit, and the red box highlights a PVN012A MOSFET switch that takes in a 16V DC voltage to be supplied to the operational amplifiers.

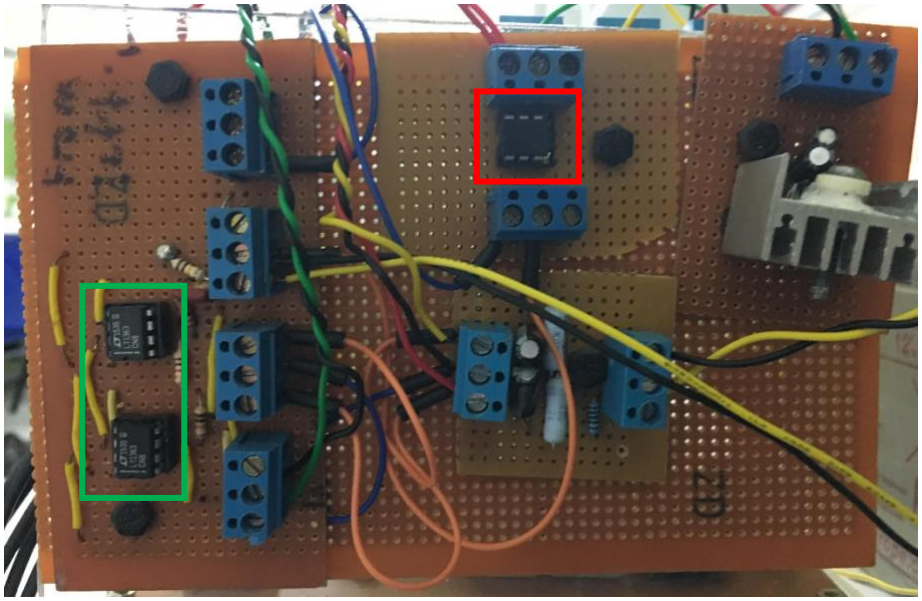

Figure 24: Receiver-side amplifier circuit

The resulting signals through this setup are signals that are amplified enough to be processed and analysed. This process is repeated with simulated laboratory defects being implicated on the pipeline setup to observe changes in signal patters.

3.3.2 Signal Sampling System

Figure 25 shows the flow that begins from the amplified signal being acquired by a Data Acquisition tool before sampled to be ready to be used as a machine learning dataset. It is the process of removing the signal offset by using a 0.1μF capacitor. By setting the sampling frequency at 250kHz and sampling period to 0.2s, there will 2 desired signals on MATLAB scanning via NI USD-6212. This process is further discussed in this sub-section and the following 3.2.3.

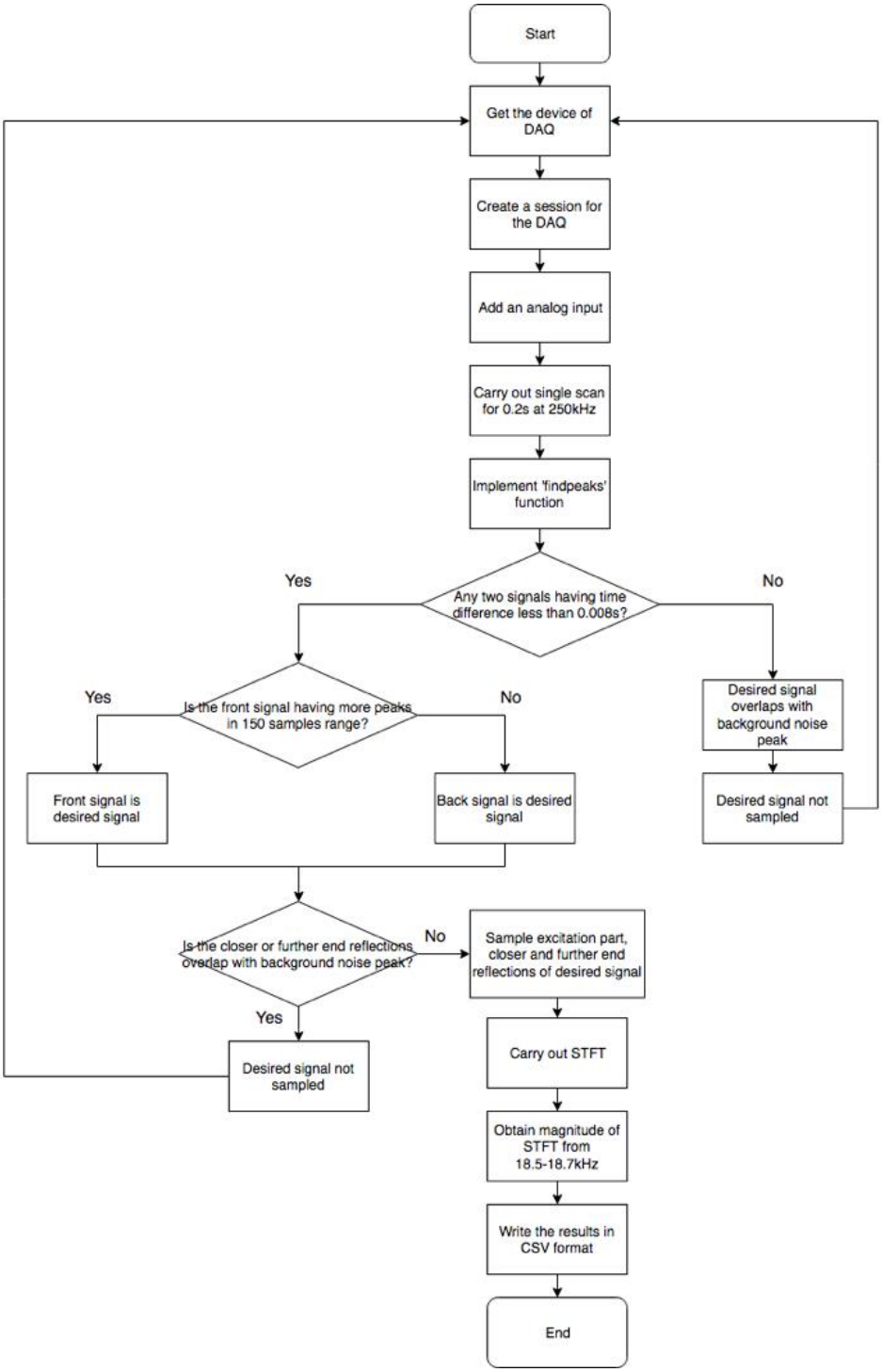

Figure 25: The flowchart of the sampling of data from the monitoring system

As mentioned, a Data Acquisition tool (DAQ) by National Instruments (NI) was used to acquire output signal to be sent to the MATLAB platform for data sampling.

The USB-6212 DAQ model from NI was used for this setup since MATLAB had its library available in its Data Acquisition Toolbox for it. At a sample rate of 250kHz, using 1 analog input done through a piezoelectric sensor to sample the output signal, a single scan from the analog input is done and stored inside the 'data' variables. The scan duration was set at 0.2s.

Fourier transform was attempted to be used to sample the desired signal out from the output signal as through the frequency domain the intensity of the tone-burst signal at the 18.5kHz could be observed. It was discovered however that the Short Time Fourier Transform (STFT) that was as applied showed high intensity even for the background noise of the signal on the frequency domain. This made it fairly difficult to differentiate between the desired output signal and the background noise. In addition to that, due to the evenly distributed frequency of background noise, filtering techniques appeared to be quite redundant as well.

Although, it was also noted that the background noise and the desired output signal had several distinctive properties. The noise was coming from the ultra-fast switching diodes and they had a pattern of spiking once every 80ms. Since MATLAB had a 'findpeaks' function, it was possible to find the desired output signal in between two consecutive peaks of background noise. It was achieved by turning MATLAB to find peaks with a minimum

height of 2V every 5ms which are basically non-zero peaks sitting in between noise peaks.

Once a desired output signal peak was discovered, the time difference between the one of the noise peak and the desired output signal peak is recorded and this info together with the peak value would be recorded as 'location' and 'peak' data. The next step would be to determine if the desired peak was a closer to the front noise peak or the back noise peak, or even if it was situation in the middle of both noise peaks. The location of the back noise peak will be stored as 'marked_location_back' while the location of the front noise peak will be stored as 'marked_location_front'.

In the first sample the time difference was below 0.0042s and therefore suggests that the signal was a most overlapped with the background noise. However, the other conditions where the desired signal could be sampled is recorded as 'desired_signal_sampled=1'.

3.3.3 Optimising Sampling Signal

Due to certain conditions such as the short scanning time period which is 0.2s only, it is quite difficult to get constant waveforms all the time as the two desired output signals will appear at any point of the waveform inconsistently and this could end up as the sampling signal being the background noise peak.

To prevent this from happening, the theoretical working principle of the DAQ is revisited and reminded that it is essentially an Analog-to-Digital

Converted (ADC) and therefore analog signals are stored as data points consisting of 'data' and 'time' variables. By understanding this and knowing the number of samples (50000) that is recorded on MATLAB after a scan is done, it is possible to calculate the number of samples by multiplying the sampling rate and the sampling time together i.e.:

$$Number\ of\ samples = (250k * 0.2) + 1$$

$$= 50001\ samples$$

Equation 9: Number of samples

Using this simple mathematical technique, it is now possible to locate the waveform in terms of sample points. For instance, if the point in the waveform that is needed to be looked at is at the time 0.07234s, then the number of sample for that point would be:

$$Number\ of\ sample = (250k * 0.07234) + 1$$

$$= 18086\ sample$$

Equation 10: Number of samples

This means that the waveform point in terms of digital data is located at the 18086^{th} data point.

Figure 26 shows a zoomed in image of the desired signal out of which only a tiny section will be sampled for analysis. Point 3 and 4 shows the excitation part of the signal and by using the mathematics explained earlier

the number of samples between these points is calculated to be 76. The first signal after the excitation signal is the desired signal to be sampled.

Calculated numbers may differ slightly from the numbers shown on MATLAB and therefore calculated numbers are rounded to the nearest multiple of 5, where in this case 76 becomes 75.

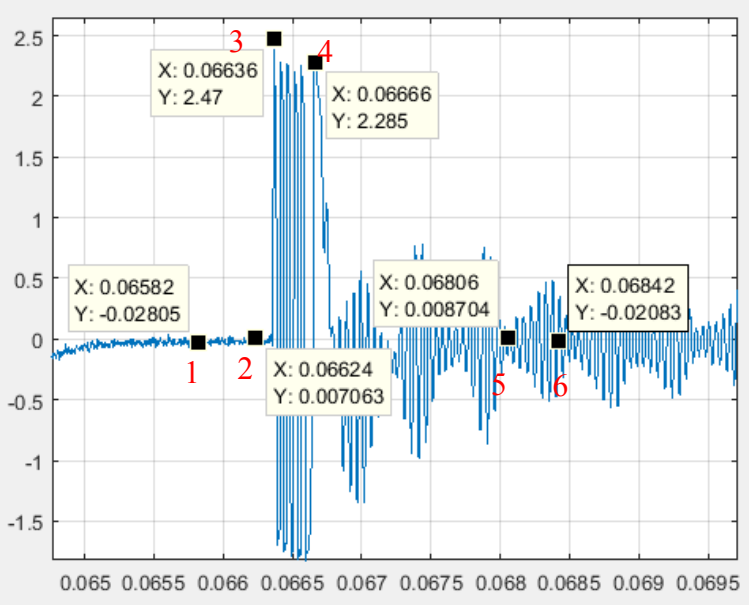

Figure 26: Zoomed in and labelled version of desired signal

The sample number for the first peak is calculated and stored as 'no_marked_front' and for the second peak as 'no_marked_back'. Using the 'findpeaks' function on MATLAB again, front and back peaks with a minimum amplititude of 2V will be stored as 'check_front_peak', 'check_front_location', 'check_back_peak' and 'check_back_location'. By comparing both the signals, the desired signal is taken as the one with the more number of peaks in the category. Desired signals cannot be sampled

when background noise overlaps with the reflected part of the signal as shown in figure 27.

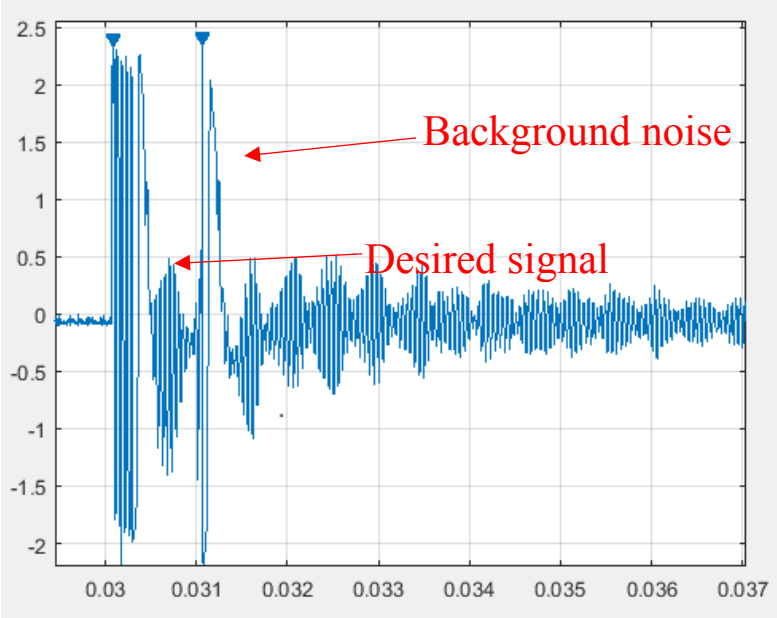

Figure 27: Waveform of background noise overlapping with the reflection part of desired signal

The absence of filtering for this process of signal sampling means that background noise cannot be eliminated. Therefore, the sampling process is repeated with another round to further optimise the results.

By applying the calculations for the number of samples again between points 4 and 6, the number of samples calculated turned to be 441, and then rounded down to 440. The desired signal is considered not sampled successfully if the difference in number of samples between the desired signal's peak and the background noise peak is less than 440 ('desired_signal_sample'=0). In this case the sampling process is repeated again by the while loop in the code until a 'desired_signal_sample'=1 result

is obtained, which means there is no overlap between the background noise and either end of the reflected signal. Once no overlap is ensured, 110 samples before the marked peak and 440 samples after the marked peak are sampled.

Now that the desired signal has been successfully detected and sampled, the process is repeated with the zoomed-in version of the sampled desired signal. This is because the actual parts of the signal that are needed are the excitation signal, front end reflection and back end reflection signals. Therefore, the first peak of the excitation signal is first detected and the number of samples in between that part and the back end of the signal is 328 samples or equivalent to roughly 1.3ms. This will be the number of samples from the front to the end of the signal that will be sampled.

Finally, the desired signal that is sampled is processed with Short Time Fourier Transform, STFT where the quantities *s*, *f* and *t* are the magnitude, frequency and time of the STFT results. The *s* values carry both real and imaginary values. Since machine learning requires real numbers to work with, the absolute value of *s* is obtained and used. The STFT results range from 0-120kHz but the frequency required for processing is only in the range of 18.5kHz corresponding with the frequency of the signal from the tone-burst generator.

Eventually data between the frequency of 18.5-18.7kHz is sampled and stored in a Comma Separated Value (CSV) file. This format of storage is preferred compared to other types as it is robust and simple with data

separated only by commas making it very easy to be read on an integrated development (IDE).

3.4 Phase 3: Machine Learning Model

Up till the section 3.2, the process of acquiring sampling data and storing it in a CSV format was discussed. This section explore the method of processing the data to be ready to use for the Support Vector Classification (SVC) model which will be hosted on the Raspberry Pi 3 as mentioned previously. SVC will be used to classify different classes of defects, range from no defects, to 1.0mm dent, 1.5mm dent and two defects.

The Raspberry Pi 3 was used with the PuTTY protocol to transfer data to it. 10 sets of data is connected for each class of defect which equals to a 10 CSV files. The training-testing ratio is divided 50-50 where out of the 10 sets of data, 5 will be used for training and the other 5 will be used to testing. 2 files are created in the Raspberry Pi; training.dat to store training data and testing.dat to store testing data.

Figure 28 shows the data in the CSV format. It is noted that the first number of each line represents the label or class of defect, and the numbers following to that in each line represents the features of the defect.

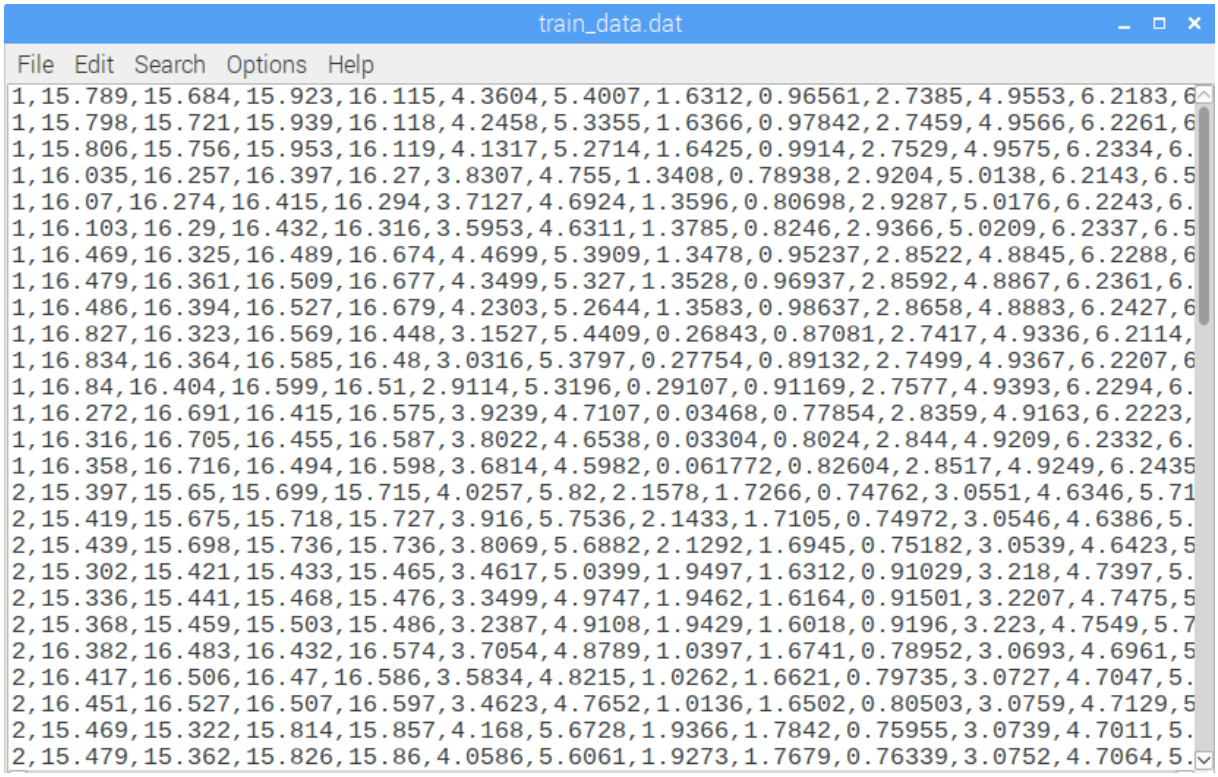

Figure 28: Monitoring system output data in CSV format

From this point, the data will need to be processed before training can happen. This is to make sure irregular values can be removed so that the training model could be optimised as much as possible with a good accuracy.

As such, scaling is the technique used to pre-process the dataset ahead of training. Scaling essentially scales the value of the data based on a mean value, in this case the end result will be within the range of +5 and -5.

The SVC will need to be tuned continuously to obtain the best accuracy of defect classification, based on the following parameters shown in Table 10.

Parameter	Definition	Remark
C	regularization parameter, C, of the error term	Essentially determines the range of support vectors used to determine the hyperplane. The higher the value of C, the higher the amount of error in the SVC will be penalized.
Kernel	specifies the kernel type to be used in the algorithm. It can be 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed', or a callable. The default value is 'rbf'	Determines the revisualization of the dataset.
Degree	the degree of the polynomial kernel function ('poly') and is ignored by all other kernels. The default value is 3	Only used if Kernel is selected as 'poly'. Not used in this research.
Gamma	the kernel coefficient for 'rbf', 'poly', and 'sigmoid'. If gamma is 'auto', then 1/n_features will be used instead	Determines the number of support vectors within the range in C that are used to determine the hyperplane. Higher gamma value takes into account only points that are close to the hyperplane whereas lower gamma value considers points that are further away than the hyperplane too.

Table 10: Support Vector Classifier, SVC parameters

For the purpose of this research, only the parameters C, Kernel and Gamma are used as the kernel types being used are Linear and RBF.

Figure 29 shows how the flow of machine learning process on Raspberry Pi will be like.

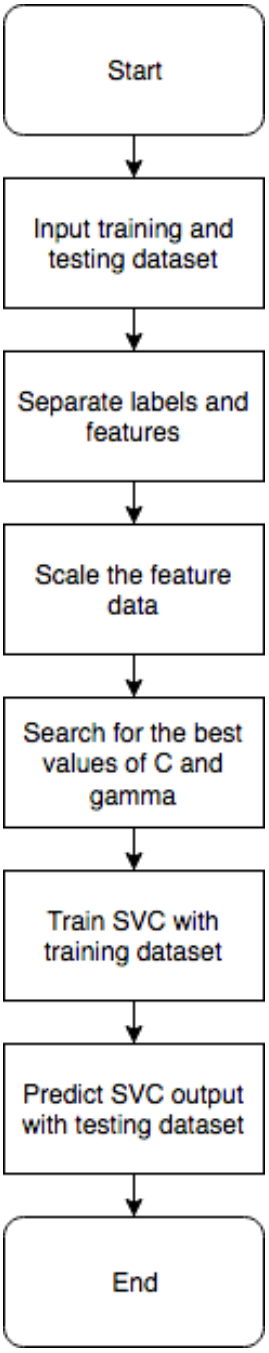

Figure 29: The flowchart of the machine learning model on Raspberry Pi

3.5 Summary

This chapter explains this research work by breaking down the entire process into three phases:

1. Signal Generation and Transmission
2. Signal Sampling and Optimisation
3. Machine Learning Model

A combination of a microcontroller and a DAC was used as the tone-burst generator to produce high voltage analog signals for back and forth transmission along the pipeline. Since ultrasonic transducers require high voltage to be activated, a high-power amplifier was used to amplify the signal before transmitting it to the transducers. The incoming signals are put through a multiplexer circuit to split them into quadrants. Since the incoming signals are relatively weaker than when transmitted, they have to be amplified again before sent to a DAQ for sampling, and subsequently to the machine learning model for training and testing.

Chapter 4.0 will discuss the process of trial and errors, and improvements made in the hardware setup, as well the comparison of results for different parameters tuned on the machine learning model.

Chapter 4: Results and Discussion

4.1 Introduction

The results of the hardware implementation and the machine learning implementation are visited here. It is important to make a note that the way the pipeline was setup in the laboratory as shown in Figure 30 was just as important as everything else in ensuring the quality of the results.

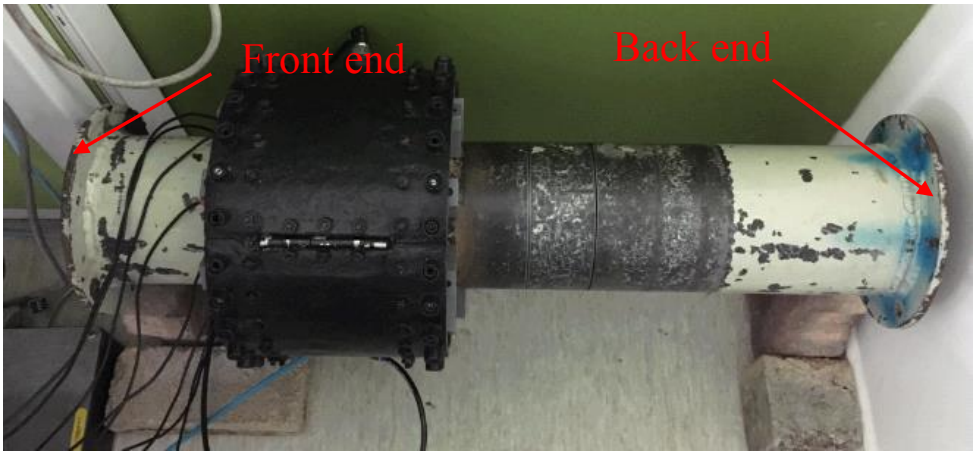

Figure 30: Ultrasonic transducers and pipe setup

It is also very important to make sure the output signal that is generated can be modified according to needs and circumstances. This is due to the operating conditions being different than in a natural scenario of long pipelines in real environments. For instance, if the time delay between two tone-burst signals is too short, then the first generated signal might not have sufficient time to achieve full form before reaching the output at the oscilloscope. Similarly, if the number of cycles and frequency is set to be high the two generated signals might overlap with one another whereas a low number of cycle and low frequency may create a situation where the

signals are too spread out and are unable to be captured as well. To avoid the above mishaps, the codes that have been uploaded into the STM microcontroller can be modified according to the number of cycles and frequency of the burst signal to produce a desired share of output signal.

In figure 31, an example of a good tone-burst signal is shown with 8 number of cycles and a frequency of 18.5kHz. The labels on the figure 29 show the excitation signal which is the residual signal from the transmitted tone-burst signal. This excitation signal also plays the role of a reference to the reflected signal in the pipe. Next is the front end and back end reflection signals with the signal in between them being the portion that will be observed and analysed to assess the condition of the pipe. The remaining signals that come after are discarded as they are just the multiple reflections from both end.

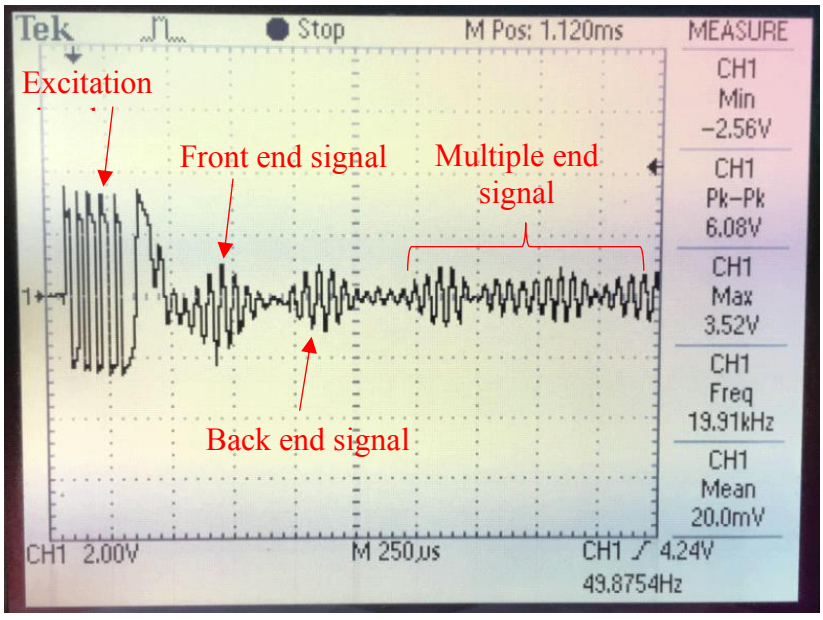

Figure 31: Output signal of tone-burst signal with number of cycles = 8, frequency = 18.5kHz

Figure 32 shows an example of a poor placement of the transducer collars at the middle portion of the pipe. The collars have to be placed at either end of the pipe. The reason for this to avoid output signals to be superimposed onto one another making it difficult to process and analyse as shown in figure 33.

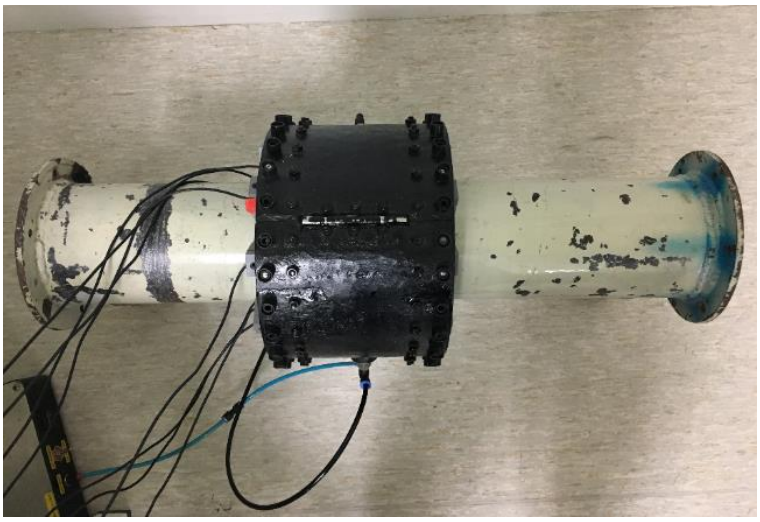

Figure 32: Ultrasonic transducers placed at the middle of the pipe

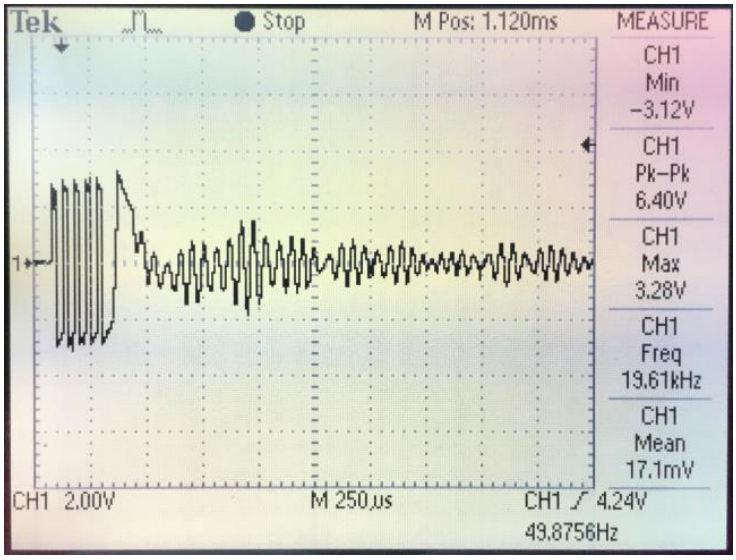

Figure 33: Output signal of monitoring system when the ultrasonic transducers are at the middle of the pipe

4.2 Oil and Gas Monitoring System

The ultrasonic transducers had to be tuned with the right frequency and number of cycles of the tone burst signal to obtain the right output. This is a trial and error process. The number of cycles was set at 5 and the frequency of the tone-burst signal was adjusted from 18.5kHz to 20kHz and eventually slightly more to see how the output responded accordingly. Figure 34 shows an example of a signal over a longer time scale. However, for the purpose of analysis, the only the desired portion of the signal will be focused on to be analysed. This is because the spikes seen after the desired signal are just background noise as a result of signal waves bouncing back and forth the two ends of the pipeline.

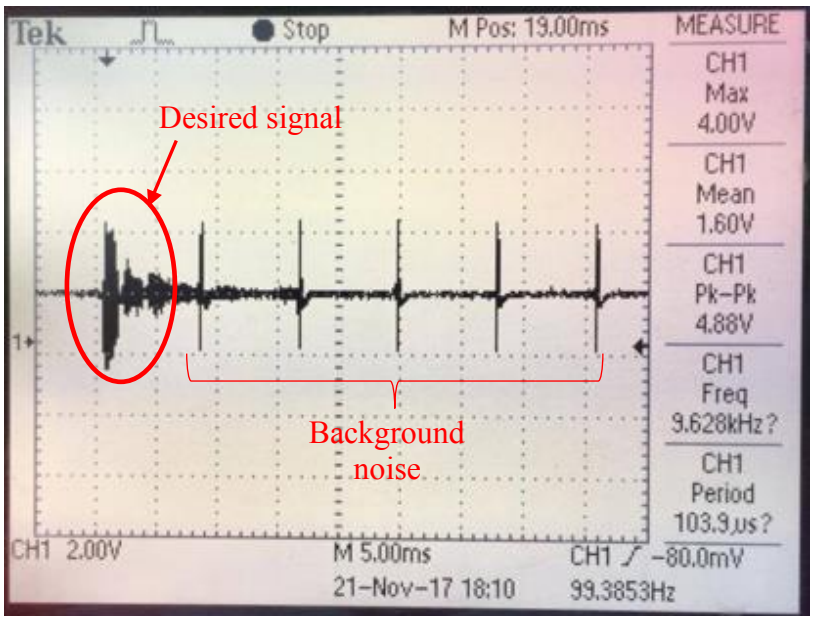

Figure 34: The output of oil and gas monitoring system at longer time scale

In Figures 35 to 38, it can be seen that the shape begins to differ when the tone burst signal frequency is adjusted from 20kHz to 26kHz. It was observed that the shape of the excitation signal, the front and back end reflection signals are still evident up to a tone-burst signal frequency of 26kHz.

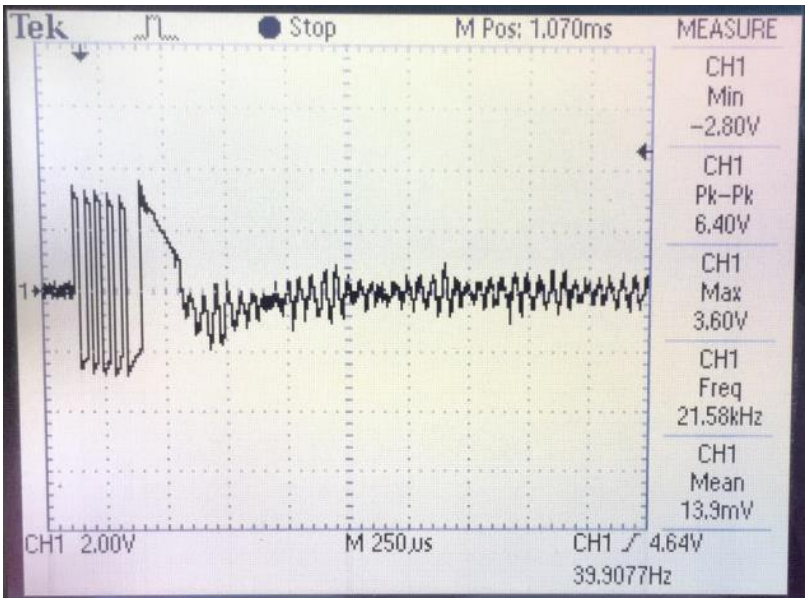

Figure 35: Monitoring system output with tone burst signal frequency of 20kHz

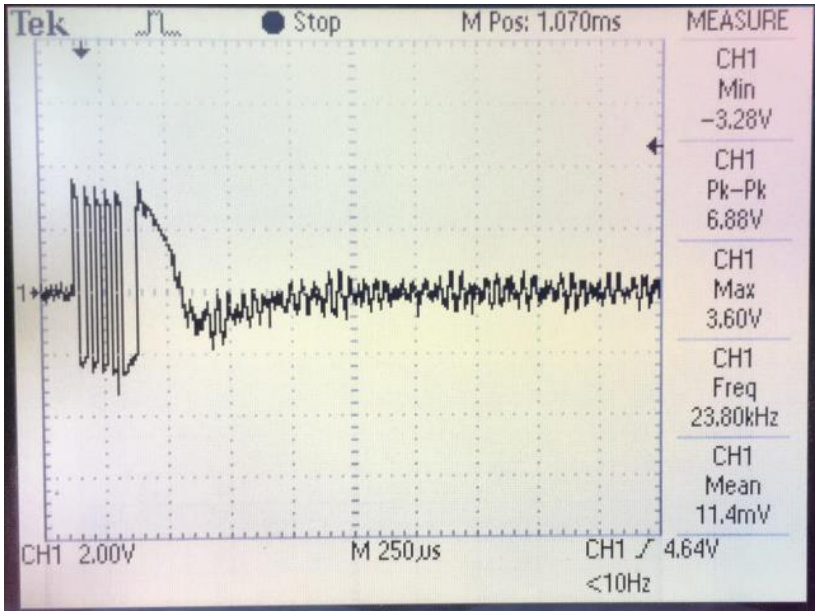

Figure 36: Monitoring system output with tone burst signal frequency of 22kHz

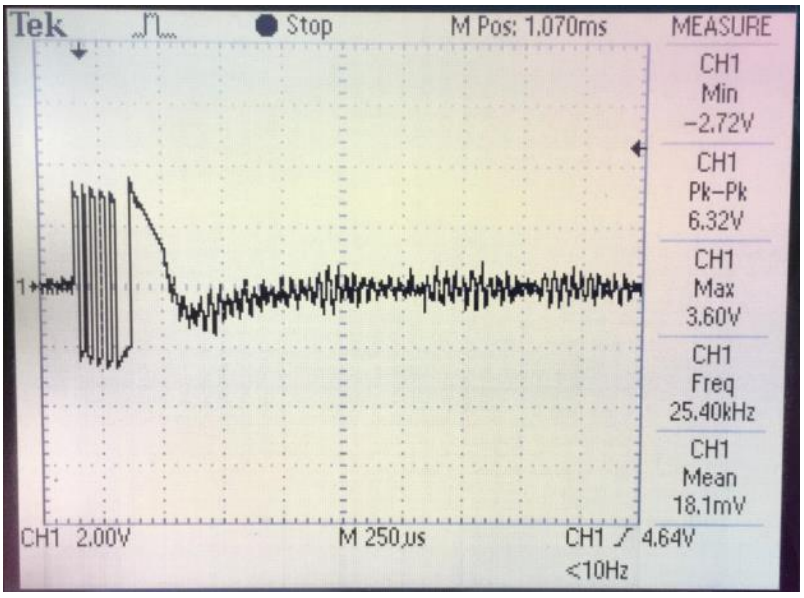

Figure 37: Monitoring system output with tone burst signal frequency of 24kHz

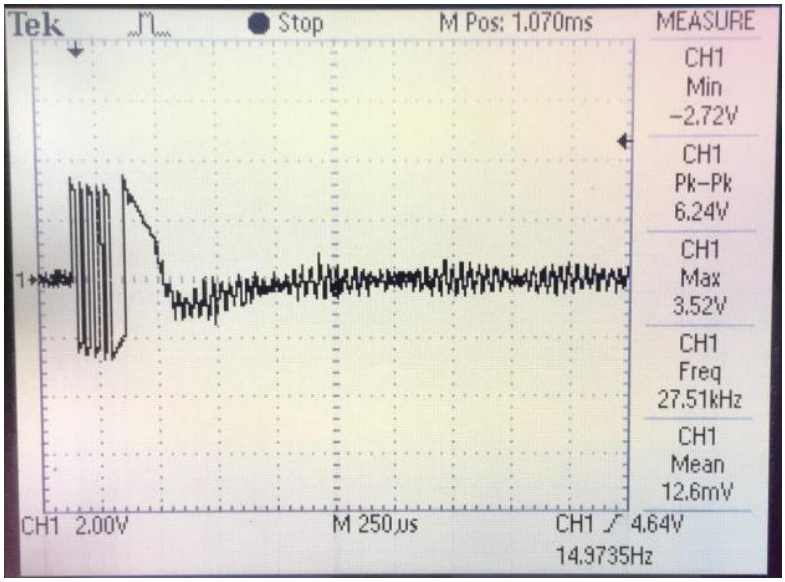

Figure 38: Monitoring system output with tone burst signal frequency of 26kHz

However, in Figures 39 to 41 from tone-burst signal frequency of 28kHz to 32kHz the output begins to lose its shape and peaks become very difficult to be observed. This due to the limitations of the PA240cc operational amplifier that has a limited bandwidth hence unable to amplify signals at higher frequencies.

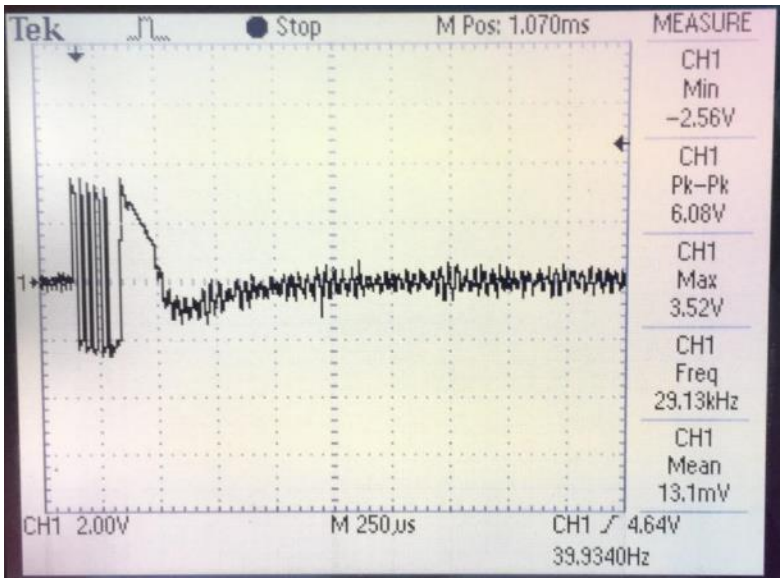

Figure 39: Monitoring system output with tone burst signal frequency of 28kHz

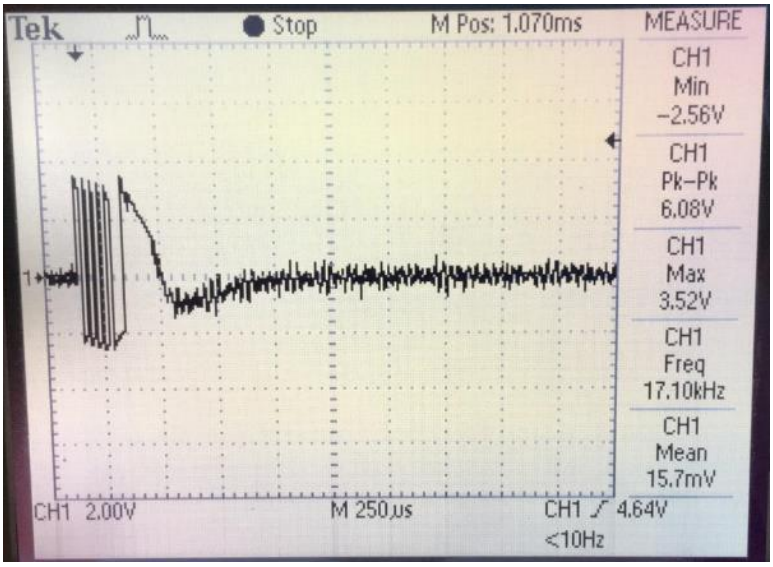

Figure 40: Monitoring system output with tone burst signal frequency of 30kHz

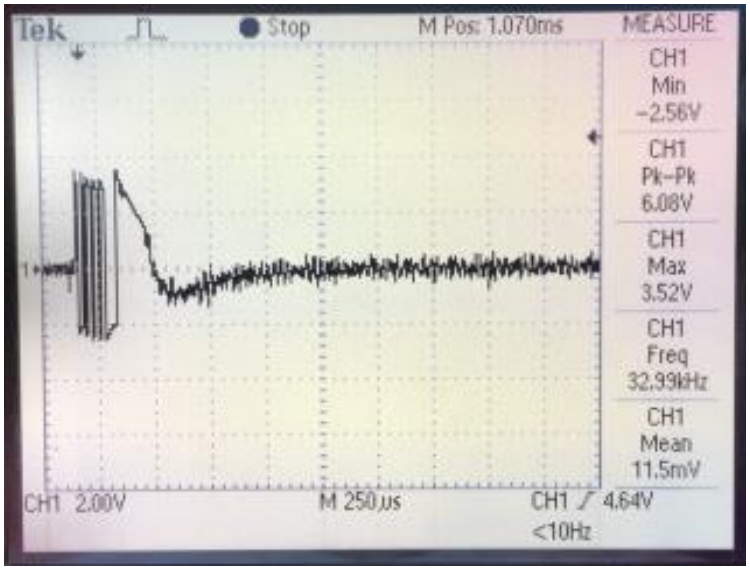

Figure 41: Monitoring system output with tone burst signal frequency of 32kHz

Similarly, at lower tone-burst signal frequencies such as 15kHz shown in figure 42, results are hard to be read because although amplitude gets better at lower frequencies, the exact locations of the front and back end reflection signals are difficult to be determined. The spread-out signals have a tendency to overlap and distort its form before even achieving full form of a signal in the reflection process in the pipe.

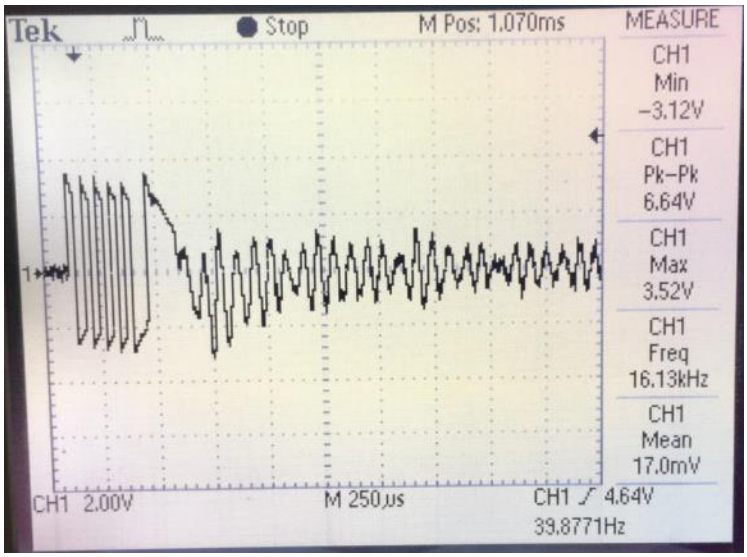

Figure 42: Monitoring system output with tone burst signal frequency of 15kHz

Discrepancies in results are also observable when the number of cycles is adjusted. The figures 43 and 44 below show that. It is noted that while the peak shapes are still visible, it is no longer as clear as the original setting of number of cycles for the tone-burst signal. This is because the signals didn’t have enough time to shape up before being overlapped with other reflection signals.

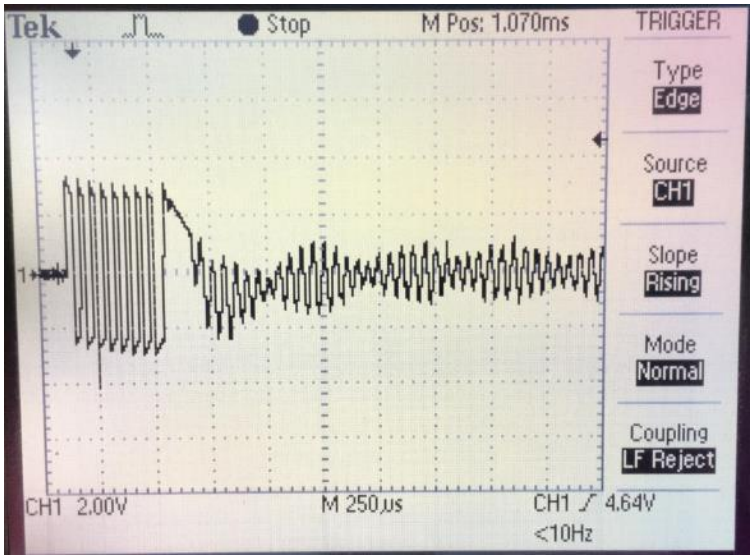

Figure 43: Monitoring system output with 8 cycles of tone burst signal

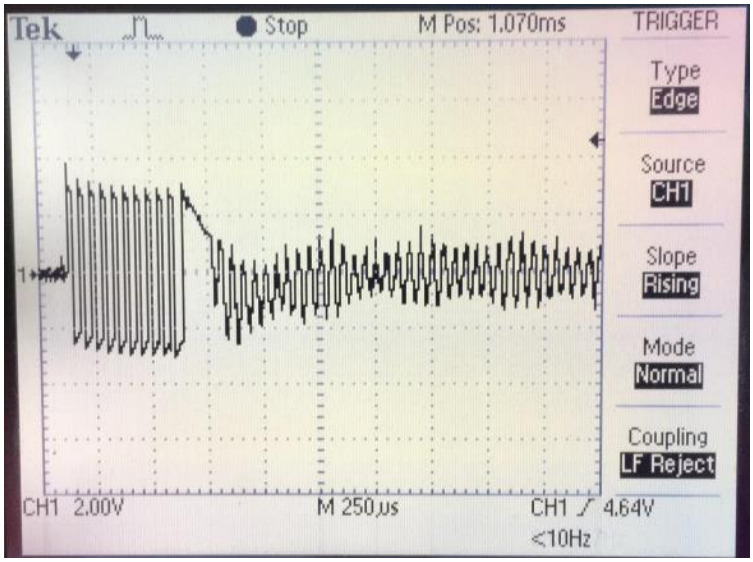

Figure 44: Monitoring system output with 10 cycles of tone burst signal

Similarly, at lower number of cycles, energy is loss before reflection could happen and this affects the signal as well making it difficult to read and differentiate as seen in figure 45.

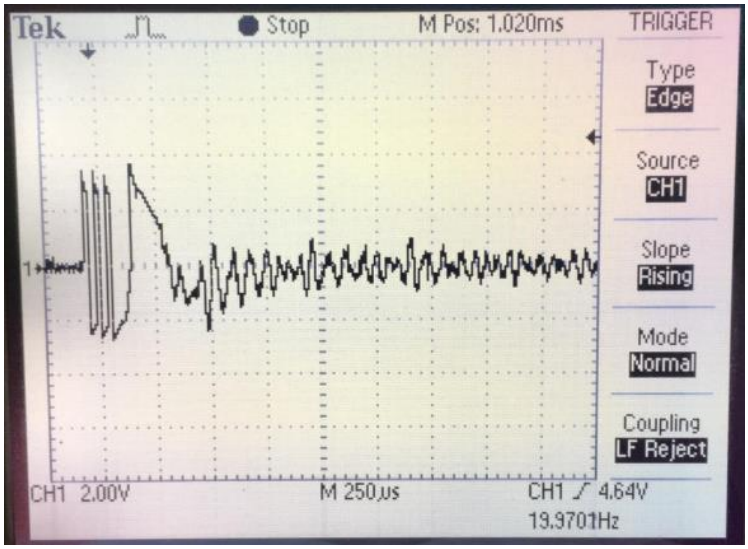

Figure 45: Monitoring system output with 3 cycles of tone burst signal

Therefore, it was understood that high frequencies and number of cycles are more suited for longer pipelines but not for shorter ones, whereas frequencies and number of cycles that are too low are not suitable to be used due to energy loss causing negative effects to the data quality.

4.3 Signal Sampling System

Figure 46 shows the signal after it was scanned on MATLAB. The desired part of the signal to be analysed are the parts circled in red. Every other subsequent peak after that are noise coming from weaker signals continuing to bounce back and forth within the pipe. In this sample, two peaks mean two signals had been transmitted.

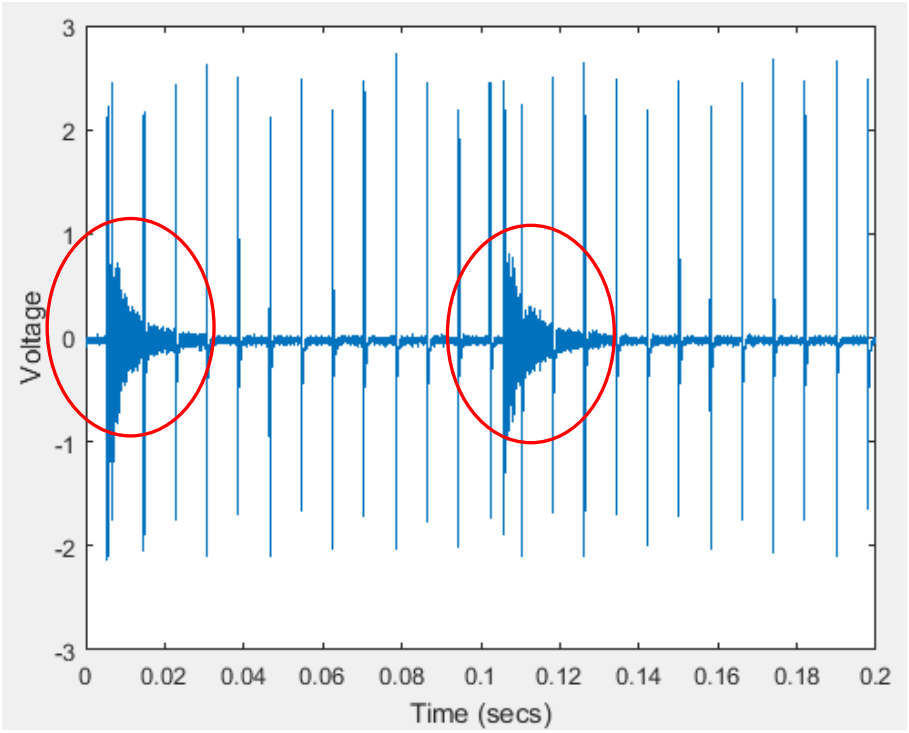

Figure 46: Single scan by using USB-6212 on MATLAB

STFT was performed on this signal and Figure 47 shows the intensity of the peaks in the frequency vs time domain, regardless of the background noise or desired signal. The circled part of the signal being brighter in color which shows that the signal had higher intensity or magnitude. The other yellow stripes are basically the background noise from the STFT results. Figure 48 shows a sampled desired signal.

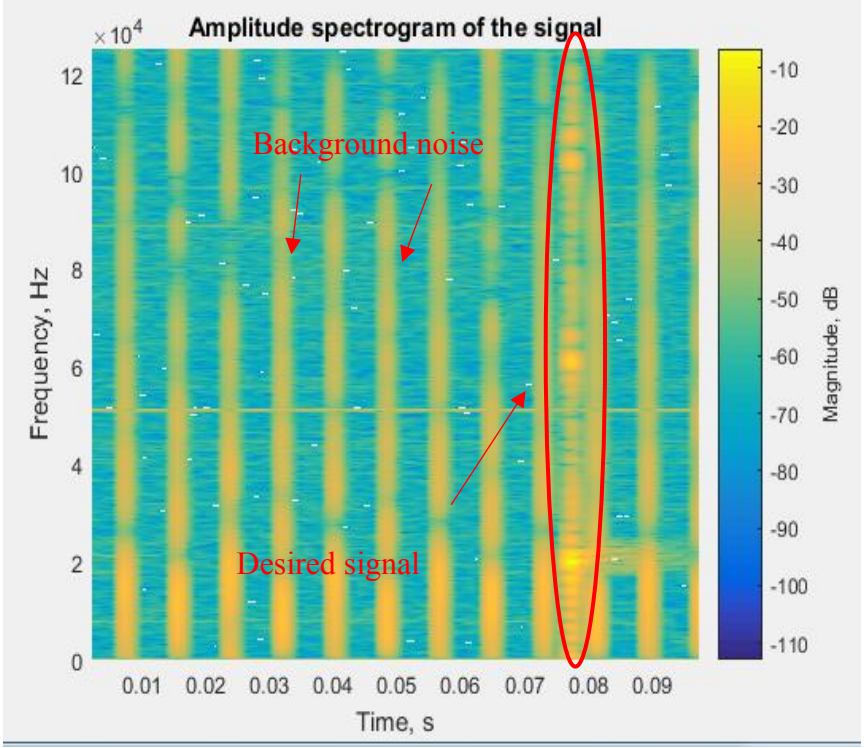

Figure 47: Spectrogram of half of the scan

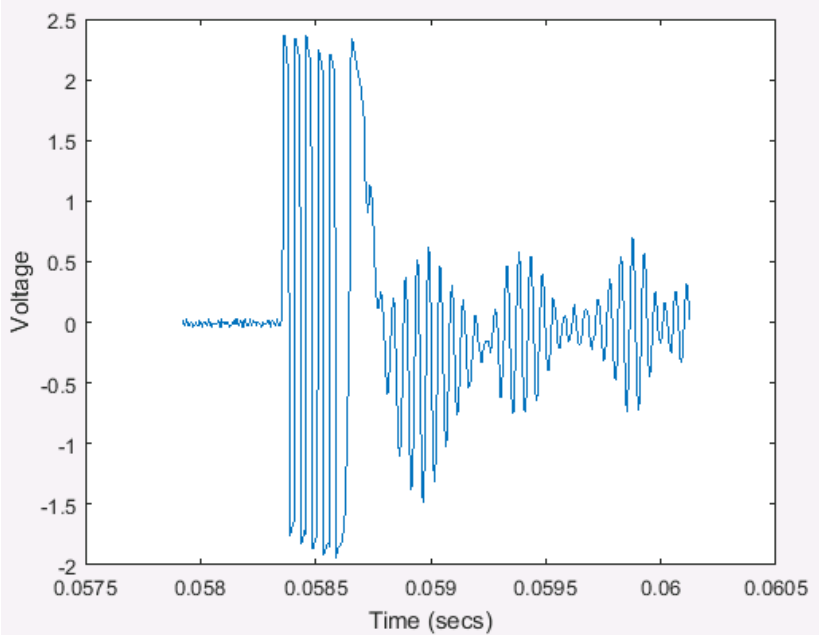

Figure 48: Sampled desired signal

The absence of defects translates to an almost flat wave between the front and back end reflection signals as seen in Figure 49 and Figure 50 as a zoomed in version of Figure 49.

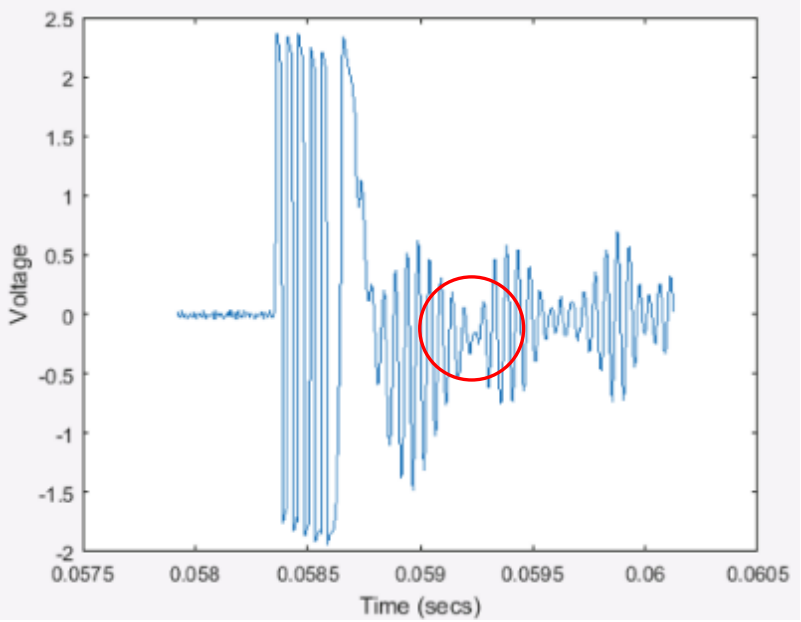

Figure 49: Desired signal when there are no defects

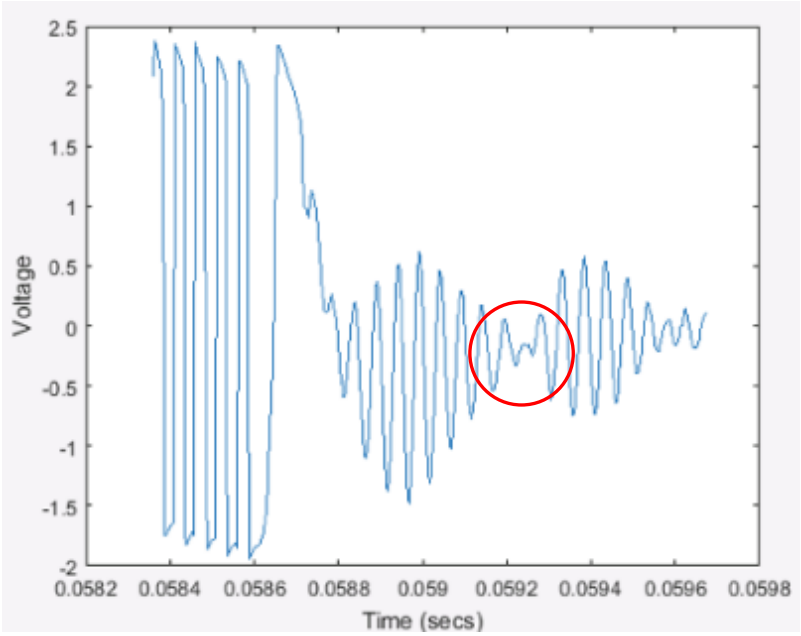

Figure 50: Zoomed of Figure 60

When defects are caused on the pipeline as seen in Figure 51, a small spike can be seen in the circled area of Figure 52 and in Figure 53 which is the zoomed in version of Figure 52. This spike means that the defect on the wall of the pipeline had caused a small portion of the signal to be bounced back to the receiver, instead of bouncing back after completing the journey to the end of the pipeline.

Figure 51: Oil and gas pipe with 1.0mm or 1.5mm defects

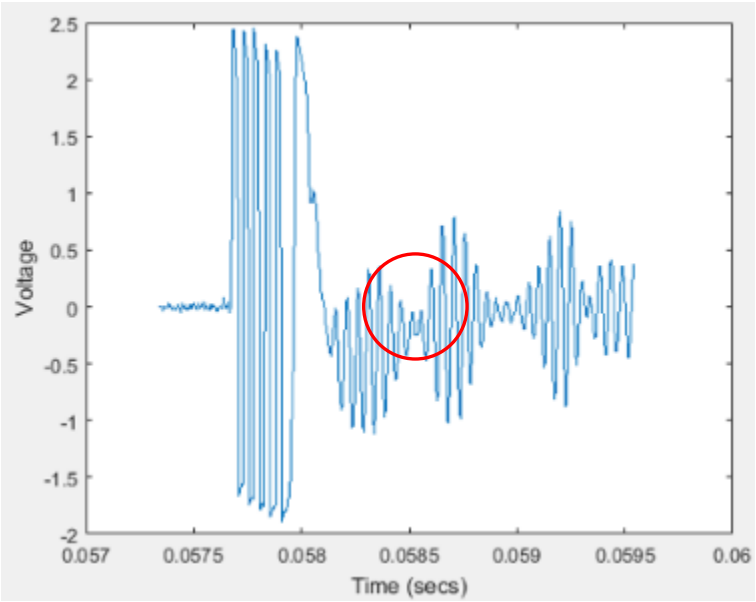

Figure 52: Desired signal when defect is 1.00mm

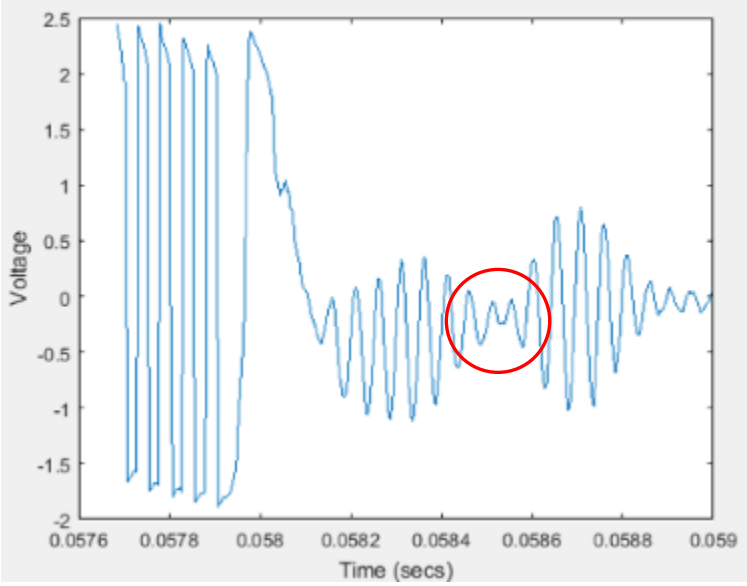

Figure 53: Zoomed in version of Figure 63

At a defect of 1.50mm, the spike becomes more obvious as seen in Figures 55 and 56. This is because as the dept of the cut increases, wall thickness on the point of the defect is reducing, hence causing an 'artificial wall' to cause more parts of the signal to be bounced back to the receiver as seen in Figure 54.

1.50mm defect creates an 'artificial wall'

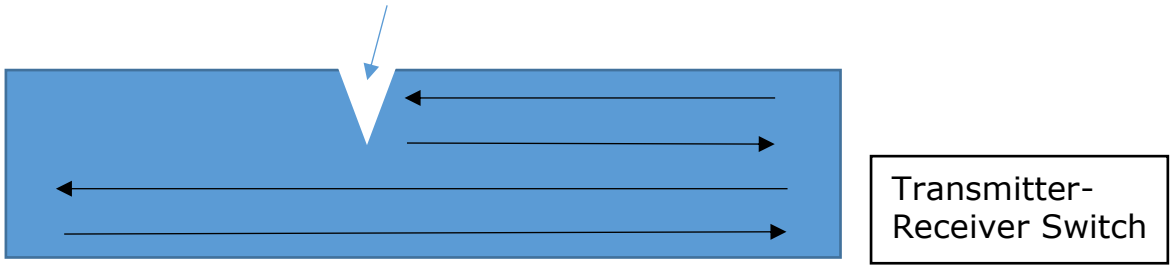

Figure 54: Artificial wall bouncing signals back to receiver

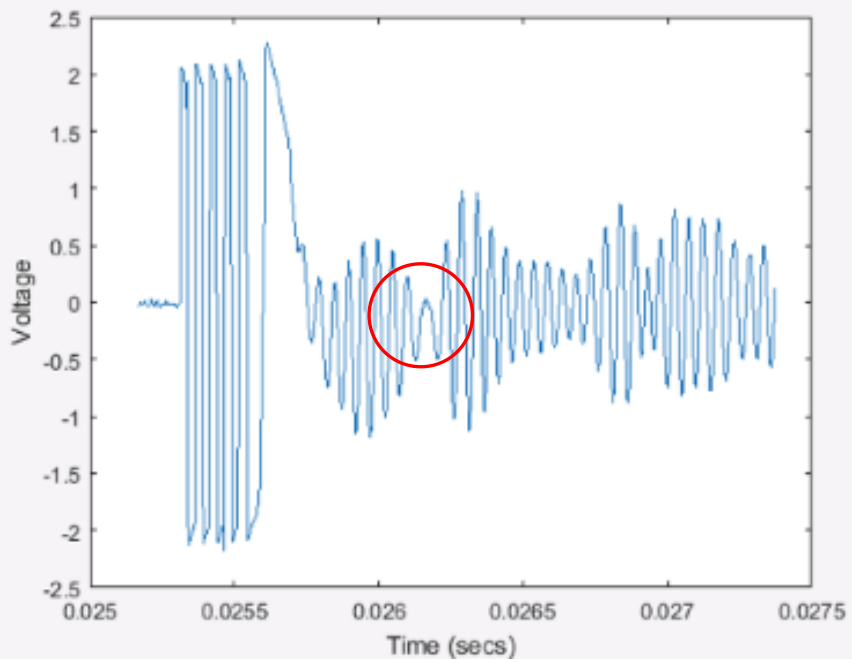

Figure 55: Desired signal when defect is 1.50mm

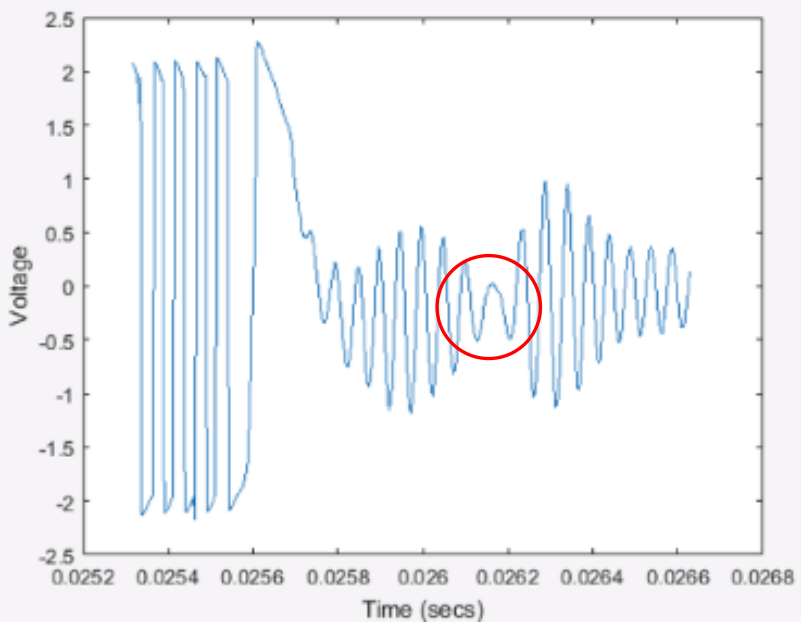

Figure 56: Zoomed in version of Figure 65

Similarly, with two 1.5mm defects, the spikes start to change more and can be seen a bit more clearly in Figure 57 and 58.

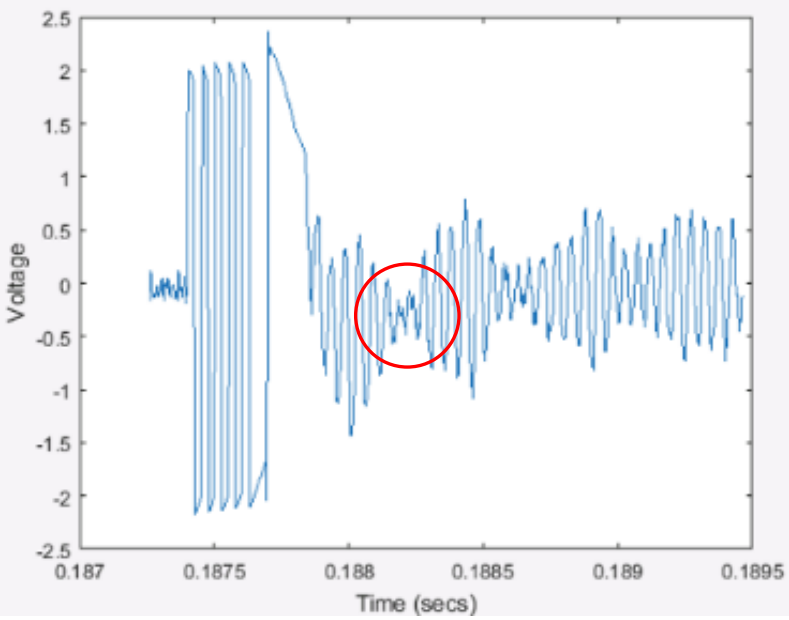

Figure 57: Desired signal when there are two 1.5mm defects

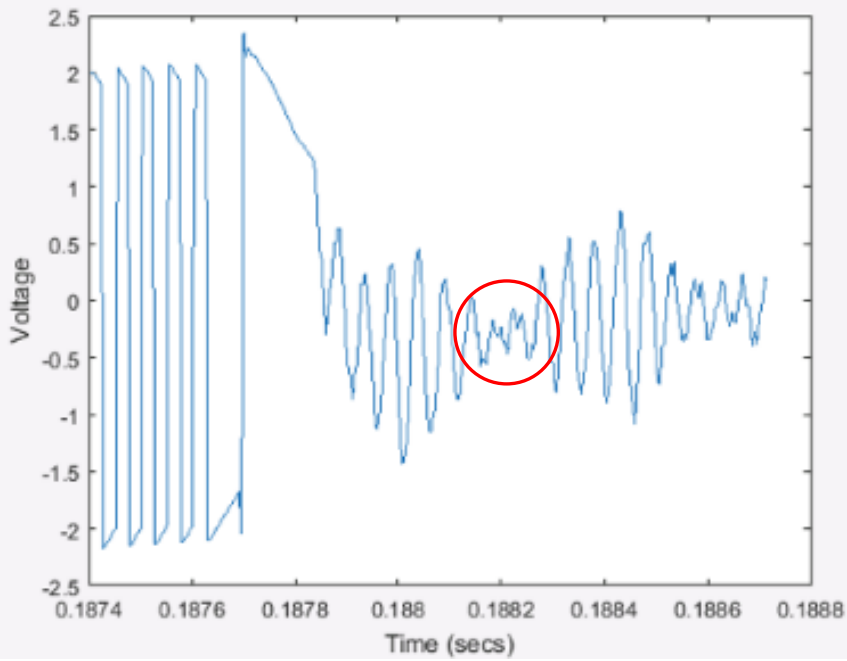

Figure 58: Zoomed in version of Figure 67

4.4 Machine Learning

Date of three sample sizes were used to train and test the model. For each case, two different kernel functions and the C parameter were manipulated to observe the disparity in accuracies.

C is a parameter that aims to balance between error and best hyperplane fit. For instance, if C is high, error percentage might be low but the hyperplane may not be at the best separation point, vice versa.

The gamma parameter was set at 0.01 with only the C parameter and Kernel function changed accordingly.

Table 11 and Figure 59 shows the accuracy results of the SVC for a sample size of 36. It is seen that the accuracy was not satisfactory and seemed to be statistically insignificant to call out any conclusions.

Kernel	**C**	**Gamma**	**Model Accuracy**
RBF	0.1	0.01	67.192%
RBF	0.7	0.01	65.051%
RBF	1.0	0.01	66.465%
Linear	0.1	-	72.323%

Linear	0.7	-	72.202%
Linear	1.0	-	69.535%

Table 11: Model accuracies based on signal sample size: 36

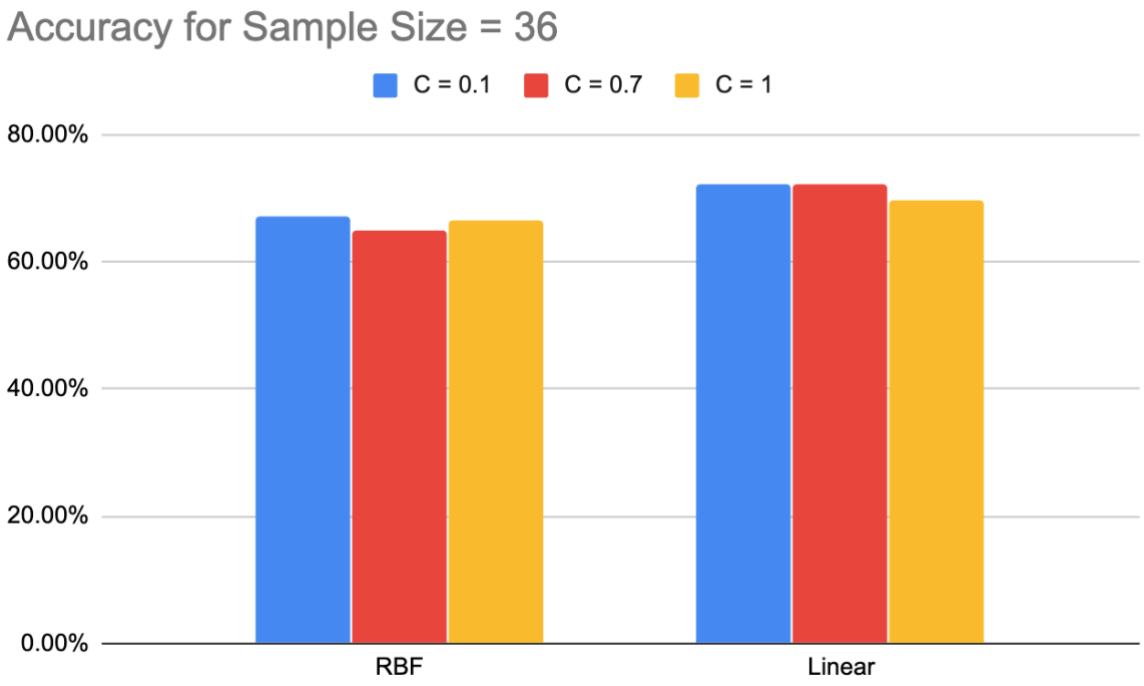

Figure 59: Model accuracies based on signal sample size: 36

However when the sample size was increased to 56, accuracies began to improve as seen in Table 12 and Figure 60.

Kernel	**C**	**Gamma**	**Model Accuracy**
RBF	0.1	0.01	95.556%

RBF	0.7	0.01	99.879%
RBF	1.0	0.01	99.879%
Linear	0.1	-	99.919%
Linear	0.7	-	99.879%
Linear	1.0	-	99.879%

Table 12: Model accuracies based on signal sample size: 56

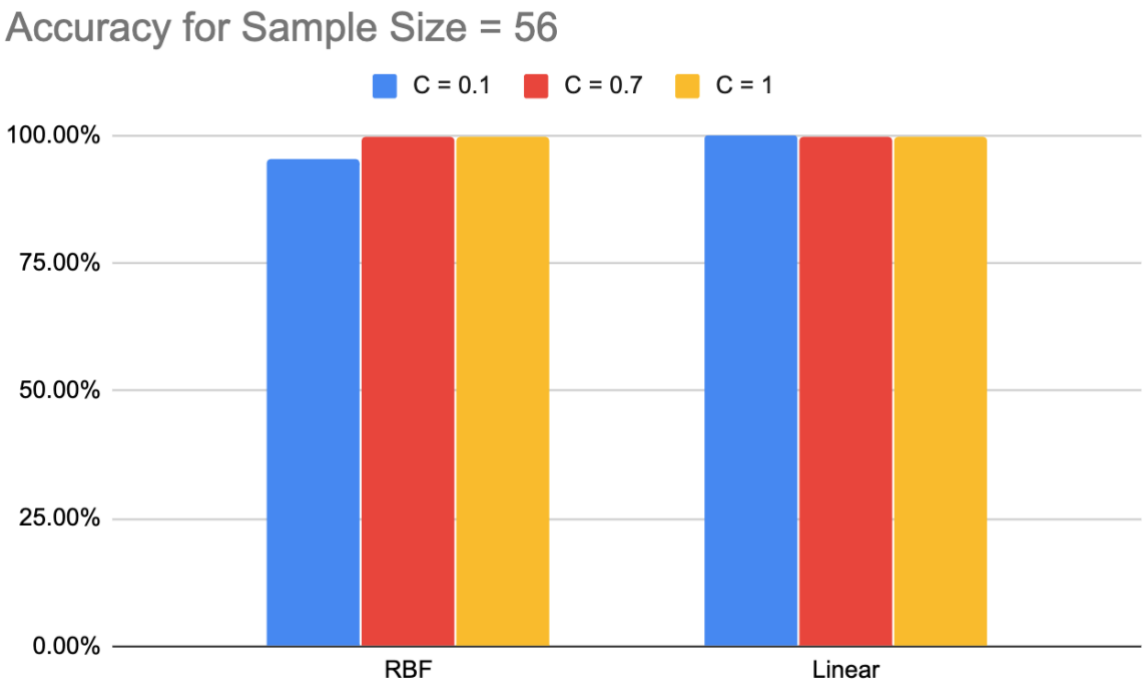

Figure 60: Model accuracies based on signal sample size: 56

The same happens in Table 13 and Figure 61 when the sample size increases to 66. In fact the accuracy is able to reach 100% with a linear kernel as choice.

Kernel	**C**	**Gamma**	**Model Accuracy**
RBF	0.1	0.01	99.838%
RBF	0.7	0.01	99.879%
RBF	1.0	0.01	99.879%
Linear	0.1	-	100.0%
Linear	0.7	-	100.0%
Linear	1.0	-	100.0%

Table 13: Model accuracies based on signal sample size: 66

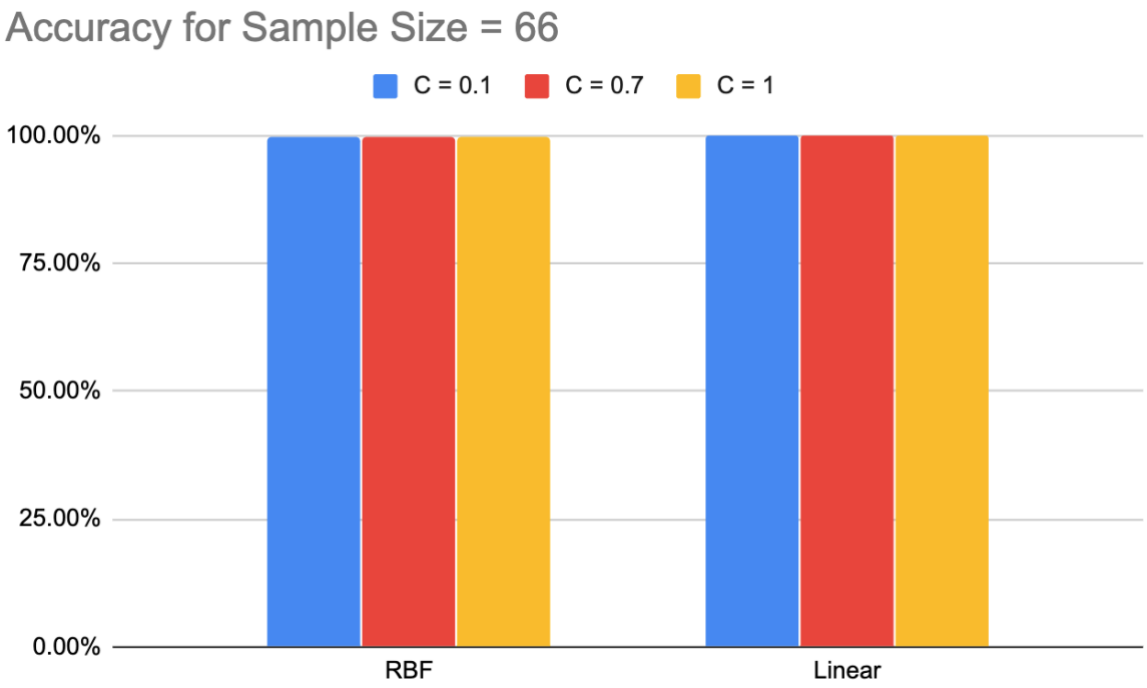

Figure 61: Model accuracies based on signal sample size: 36

4.5 Summary

Essentially it was observed that the accuracy of the model increased with the sample size. The changes of kernel type, C and gamma parameters had very little impact to results due to the following reasons:

- Since C determines the range of support vectors being taken into account in determining the hyperplane, and gamma determines how many of those support vectors in that range is actually used in forming the hyperplane, both parameters had very little impact as the data points were not vastly scattered and therefore the range and number of support vectors did not change much with a change in C and gamma.

- A larger sample size simply means more data was available to train the model, yielding in a pattern of improving accuracy as the sample size increases.

Also to be noted is the fact that this pipeline setup and defects was simulated in a controlled environment. This means that in reality, achieving 100% accuracy would be very much harder.

Chapter 5: Conclusion and Future Works

5.1 Introduction

This research work has successfully produced a proof of concept of an intelligent failure prediction system by implementing a hardware setup that generates a signal and samples the desired output signal to the machine learning model hosted on a Raspberry Pi.

5.2 Conclusion

A microcontroller was used as a tone-burst generator to generate a digital signal, which was then converted to an analog signal using a digital-to-analog converter which was essentially a R2R ladder circuit.

This signal is then further amplified with a high-power amplifier to have enough energy to flow through the pipeline. The amplified signal is fed to the LRUT transducer via a transmitter-receiver switch to be transmitted along the pipeline.

The signal that completes one cycle in the pipeline (bouncing back from the other end) is than retrieved with the same transmitter-receiver switch as a weak analog signal and then fed to a multiplexer to be split according to quadrants for specific analysis. A couple of operational amplifiers were used to amplify the weak signals to be sampled on MATLAB. Sampled signal is then split into training and testing datasets to train the machine learning

model, which was the Support Vector Classifier (SVC) that was hosted on a Raspberry Pi.

The SVC is tuned with its parameters and different sample sizes of the signal to obtain satisfactory accuracy levels.

The accuracy of the classification has been achieved to be **100%** at this point, which proves that an Intelligent Real-Time Prediction for Energy and Sensing Applications system is possible.

5.3 Future Works

Future research building on this should focus on the automation of the entire system that includes making the analysis real time without human intervention. This will require a 3rd major component to be added into the ecosystem that exist currently. An FPGA can be explored to handle the data pre-processing or perhaps any other functionalities before the machine learning model comes into play in the Raspberry Pi.

The goal is to have a continuous data flow from the tone-burst generator through the rest of the hardware setup to the machine learning model to classify and predict defects real-time without the need of a human intervention. To achieve this, more work needs to be done on identifying the missing elements that can enable such a seamless workflow for the purpose of this research.

6.0 Reference

[1] Broadribb, M.P. (2015). What have we really learned? Twenty five years after Piper Alpha. *Process Safety Progress, 34*.

[2] Lichtveld, M.Y., Sherchan, S.P., Gam, K.B., Kwok, R.K., Mundorf, C., Shankar, A., & Soares, L. (2016). The Deepwater Horizon Oil Spill Through the Lens of Human Health and the Ecosystem. *Current Environmental Health Reports, 3*, 370-378.

[3] Dai, L., Wang, D., Wang, T., Feng, Q., & Yang, X. (2017). Analysis and Comparison of Long-Distance Pipeline Failures. *Adv. Artif. Neural Syst., 2017*, 3174636:1-3174636:7.

[4] Kujawinski, E.B., Reddy, C.M., Rodgers, R.P., Thrash, J.C., Valentine, D.L., & White, H.K. (2020). The first decade of scientific insights from the Deepwater Horizon oil release. *Nature Reviews Earth & Environment, 1*, 237 - 250.

[5] Kroworz, A., & Katunin, A. (2018). Non-Destructive Testing of Structures Using Optical and Other Methods: A Review. *Structural Durability and Health Monitoring, 12,* 1-17

[6] Blitz, J., & Simpson, G.K. (1996). *Ultrasonic methods of non-destructive testing*. London, England: Chapman & Hall

[7] Shull, P.J. (2002). Nondestructive Evaluation: Theory, Techniques, and Applications.

[8] Papavinasam, S. (2013). Corrosion Control in the Oil and Gas Industry.

[9] Vanaei, H.R., Eslami, A., & Egbewande, A. (2017). A review on pipeline corrosion, in-line inspection (ILI), and corrosion growth rate models. *International Journal of Pressure Vessels and Piping, 149*, 43-54.

[10] Nagaraj, J.S. (2013). Smart Pigging in High Pressure Gas Pipeline Practical Problems and Solutions: A Case Study. *ASME 2013 India Oil and Gas Pipeline Conference, V001T02A005*

[11] Roland Palmer-Jones, R., Hopkins, P., Eyre, D. (2014). *UNDERSTANDING THE RESULTS OF AN INTELLIGENT PIG INSPECTION.* Penspen Integrity, Newcastle upon Tyne, UK. Retrieved from https://www.penspen.com/wp-content/uploads/2014/09/intelligent-pig-results.pdf

[12] Amandi, K., Diemuodeke, E., & Briggs, T.A. (2019). Model for remaining strength estimation of a corroded pipeline with interacting defects for oil and gas operations. *Cogent Engineering, 6*.

[13] Osage, D.A. (2015). Fatigue Assessment for In-Service Components – A New Part for API 579-1/ASME FFS-1 Fitness-For-Service. *Procedia Engineering, 133*, 320-347.

[14] Basso, A.V., Filho, J.E., & Shang, H.Y. (2015). ASSESSMENT OF DNV-RP-F101 METHOD IN ESTIMATING THE FAILURE PRESSURE IN CORRODED PIPELINES. *23rd ABCM International Congress of Mechanical Engineering*

[15] Hopkins, P., & Cosham, A. (2003). The Assessment Of Corrosion In Pipelines: Guidance In The Pipeline Defect Assessment Manual (PDAM). *Pipeline Pigging and Integrity Management Conference*

[16] Zhu, X. (2021). A comparative study of burst failure models for assessing remaining strength of corroded pipelines. *Journal of Pipeline Science and Engineering, 1*, 36-50

[17] Lee, G., Pouraria, H., Seo, J.K., & Paik, J.K. (2015). Burst strength behaviour of an aging subsea gas pipeline elbow in different external and internal corrosion-damaged positions. *International Journal of Naval Architecture and Ocean Engineering, 7*, 435 - 451.

[18] Ma, B., Shuai, J., Wang, J., & Han, K. (2011). Analysis on the Latest Assessment Criteria of ASME B31G-2009 for the Remaining Strength of Corroded Pipelines. *Journal of Failure Analysis and Prevention, 11*, 666-671.

[19] Shi, Y., Zhang, C., Li, R., Cai, M., & Jia, G. (2015). Theory and Application of Magnetic Flux Leakage Pipeline Detection. *Sensors (Basel, Switzerland), 15*, 31036 - 31055.

[20] Kumpati, R., Skarka, W., & Ontipuli, S.K. (2021). Current Trends in Integration of Nondestructive Testing Methods for Engineered Materials Testing. *Sensors (Basel, Switzerland), 21*.

[21] Al-Hamand, M. (2012). Modern ILI of Subsea Pipelines and Risers: Capabilities for Challenging Projects. *Middle East Nondestructive Conference and Exhibition (6th MENDT)*. Retrieved from https://www.questintegrity.com/assets/PDFs/Technical-Papers-2012/Modern-ILI-of-Subsea-Pipelines-and-Risers-Capabilities-for-Challenging-Projects.pdf

[22] *Offshore Pipeline Inspection Invista™ In-Line Inspection Technology*. (2015). Challenge Convention, Quest Integrity. Retrieved from: www.questintegrity.com/assets/PDFs/Case-studies-new-brand/offshore-pipeline-inspection-using-invista-LTR-Rev.05-15-web.pdf. PDF file

[23] Revelle, D.J. (2011). INSPECTION METHODOLOGIES AND TRADEOFFS FOR INSPECTION OF UNPIGGABLE PIPELINES. *Pigging Products and Services Association*

[24] Chen, S., Webb, G.I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowl. Based Syst., 192*, 105361.

[25] Rodriguez, M.Z., Comin, C.H., Casanova, D., Bruno, O.M., Amancio, D.R., Rodrigues, F.A., & Costa, L.D. (2019). Clustering algorithms: A comparative approach. *PLoS ONE, 14*.

[26] Schneider, A., Hommel, G., & Blettner, M. (2010). Linear regression analysis: part 14 of a series on evaluation of scientific publications. *Deutsches Arzteblatt international, 107 44*, 776-82.

[27] Abiodun, O.I., Jantan, A.B., Omolara, A.E., Dada, K.V., Mohamed, N., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. *Heliyon, 4*.

[28] Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica, 24*, 12 - 18.

[29] Beck, M.W. (2018). NeuralNetTools: Visualization and Analysis Tools for Neural Networks. *Journal of statistical software, 85 11*, 1-20.

[31] Fletcher, T. (2009). Support Vector Machines Explained. *Tutorial paper,* 1-19.

[32] Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine, 4 11*, 218 .

[33] Huang, S., Cai, N., Pacheco, P.P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. *Cancer genomics & proteomics, 15 1*, 41-51 .

[34] Amaya-Gómez, R., Sánchez-Silva, M., & Muñoz, F. (2016). Pattern recognition techniques implementation on data from In-Line Inspection (ILI). *Journal of Loss Prevention in The Process Industries, 44*, 735-747.

[35] *SSH Protocol – Secure Remote Login And File Transfer*. (2018). Ssh.com. https://www.ssh.com/academy/ssh

[36] Anatoliy, B., Galina, B., Natalia, G., Sergey, M., & Mikhail, G. (2017). Main Pipelines Corrosion Monitoring Device.

[37] Wang, H., Yajima, A., Liang, R.Y., & Castaneda, H. (2015). A Bayesian model framework for calibrating ultrasonic in-line inspection data and estimating actual external corrosion depth in buried pipeline utilizing a clustering technique. *Structural Safety, 54*, 19-31.

[38] Zhang, S., Zhou, W., & Qin, H. (2013). Inverse Gaussian process-based corrosion growth model for energy pipelines considering the sizing error in inspection data. *Corrosion Science, 73*, 309-320.

[39] Isa, D., & Rajkumar, R.K. (2009). PIPELINE DEFECT PREDICTION USING SUPPORT VECTOR MACHINES. *Applied Artificial Intelligence, 23*, 758 - 771.

[40] Lee, L.H., Rajkumar, R.K., Lo, L.H., Wan, C.H., & Isa, D. (2013). Oil and gas pipeline failure prediction system using long range ultrasonic transducers and Euclidean-Support Vector Machines classification approach. *Expert Syst. Appl., 40*, 1925-1934.

[41] Hassan, M., Rajkumar, R.K., Isa, D., & Arelhi, R. (2013). Pipeline Defect Classification by Using Non-Destructive Testing and Improved Support Vector Machine Classification. *International Journal of Engineering and Innovative Technology (IJEIT), 2*, 85-93

[42] Gaudiano, P., & Grossberg, S. (1991). Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories. *Neural Networks, 4*, 147-183.

[43] Sermanet, P., Kavukcuoglu, K., Chintala, S., & LeCun, Y. (2013). Pedestrian Detection with Unsupervised Multi-stage Feature Learning. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 3626-3633.

[44] Guo, X., Singh, S., Lee, H., Lewis, R.L., & Wang, X. (2014). Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning. *NIPS*.

[45] Wender, S., & Watson, I.D. (2012). Applying reinforcement learning to small scale combat in the real-time strategy game StarCraft:Broodwar. *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 402-408.

[46] Sakr, G.E., Mokbel, M., Darwich, A., Khneisser, M.N., & Hadi, A. (2016). Comparing deep learning and support vector machines for autonomous waste sorting. *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, 207-212.

[47] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[48] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P.A., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zhang, X. (2016). TensorFlow: A system for large-scale machine learning. *ArXiv, abs/1605.08695*.

[49] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I.J., Harp, A., Irving, G., Isard, M., Jia, Y., Józefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D.G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P.A., Vanhoucke, V., Vasudevan, V., Viégas, F.B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *ArXiv, abs/1603.04467*.

[50] Ambrogio, S., Balatti, S., Milo, V., Carboni, R., Wang, Z., Calderoni, A., Ramaswamy, N., & Ielmini, D. (2016). Novel RRAM-enabled 1T1R synapse capable of low-power STDP via burst-mode communication and real-time unsupervised machine learning. *2016 IEEE Symposium on VLSI Technology*, 1-2.

[51] Kulkarni, A.M., Pino, Y., French, M.C., & Mohsenin, T. (2016). Real-Time Anomaly Detection Framework for Many-Core Router through Machine-Learning Techniques. *ACM J. Emerg. Technol. Comput. Syst., 13*, 10:1-10:22.

[52] Brown, S., Vranesic, Z. (2008). *Fundamentals of Digital Logic with VHDL Design*. Retrieved from

https://theswissbay.ch/pdf/Books/Computer%20science/Fundamentals%20of%20digital%20logic%20with%20VHDL%20design%20%283rd%20edition%29%20-%20Stephen%20Brown%2C%20Zvonko%20Vranesic.pdf

[53] Maxfield, C. (2004). *The design warrior's guide to FPGAs: devices, tools and flows*. Elsevier.

[54] Woods, R.F., McAllister, J., Turner, R.H., Yi, Y., & Lightbody, G. (2017). *FPGA-based Implementation of Signal Processing Systems.* West Sussex, United Kingdom: Wiley

[55] Takashi J. Ozaki, T.J. (April 2014). *Machine learning for package users with R (3): Support Vector Machine*. Retrieved from www.tjo-en.hatenablog.com/entry/2015/04/20/190000

[56] Chang, C., & Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol., 2*, 27:1-27:27.

[57] Chang, C., & Lin, C. (2011). *LIBSVM*. Retrieved from www.github.com/cjlin1/libsvm

[58] Parikh, K.S., & Shah, T.P. (2016). Support Vector Machine – A Large Margin Classifier to Diagnose Skin Illnesses. *Procedia Technology, 23*, 369-375.

[59] Raschka, S., Patterson, J., & Nolet, C.J. (2020). Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence. *ArXiv, abs/2002.04803*.

[60] Harris, C.R., Millman, K.J., Walt, S.V., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M.H., Brett, M., Haldane, A., R'io, J.F., Wiebe, M., Peterson, P., G'erard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., & Oliphant, T.E. (2020). Array programming with NumPy. *Nature, 585*, 357 - 362.

[61] Kumar, V.H. (2018). Python Libraries, Development Frameworks and Algorithms for Machine Learning Applications. *International journal of engineering research and technology, 7*.

[62] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, I., Feng, Y., Moore, E.W., Vanderplas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I.D., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Vijaykumar Bardelli Rothberg Hilboll Kloeckner Sco, A.A., Vijaykumar, A., Bardelli, A.P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A.M., Lee, A., Rokem, A.S., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E.A., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J., Webber, J.T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., de Miranda Cardoso, J.V., Reimer, J., Harrington, J.E., Rodríguez, J.L., Nunez-Iglesias, J., Kuczynski, J., Tritz, K.L., Thoma, M.D., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith,

N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., & Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods, 17*, 261 - 272.

[63] Pimentel, J.F., Murta, L.G., Braganholo, V., & Freire, J. (2021). Understanding and improving the quality and reproducibility of Jupyter notebooks. *Empirical Software Engineering, 26*.

[64] Lozev, M.G., Smith, R.W., & Grimmett, B.B. (2005). Evaluation of Methods for Detecting and Monitoring of Corrosion Damage in Risers. *Journal of Pressure Vessel Technology-transactions of The Asme, 127*, 244-254.

[65] Restrepo, C.E., Simonoff, J.S., & Zimmerman, R. (2009). Causes, cost consequences, and risk implications of accidents in US hazardous liquid pipeline infrastructure. *Int. J. Crit. Infrastructure Prot., 2*, 38-50.

[66] Craig, J., Gerali, F., Macaulay, F., & Sorkhabi, R.B. (2018). The history of the European oil and gas industry (1600s–2000s). *Special Publications, 465*, 1 - 24.

[67] Geck, C.E. (2017). The World Factbook. *The Charleston Advisor, 19*, 58-60.

[68] Ramirez-Martinez, A., Rodríguez-Olivares, N.A., Torres-Torres, S., Ronquillo-Lomelí, G., & Soto-Cajiga, J.A. (2019). Design and Validation of an Articulated Sensor Carrier to Improve the Automatic Pipeline Inspection. *Sensors (Basel, Switzerland), 19*.

[69] Ayadi, A., Ghorbel, O., BenSalah, M.S., & Abid, M.R. (2019). A framework of monitoring water pipeline techniques based on sensors technologies. *Journal of King Saud University - Computer and Information Sciences*.

[70] Sarker, I.H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *Sn Computer Science, 2*.

[71] Shalev-Shwartz, S., Ben-David, S. (2014). *Understanding Machine Learning - From Theory to Algorithms*. New York, USA: Cambridge University Press

[72] Peekema, R.M. (2013). Causes of Natural Gas Pipeline Explosive Ruptures. *Journal of Pipeline Systems Engineering and Practice, 4*, 74-80.

[73] Bhaskaran, P.E., Chennippan, M., & Subramaniam, T. (2020). Future prediction & estimation of faults occurrences in oil pipelines by using data clustering with time series forecasting. *Journal of Loss Prevention in The Process Industries, 66*, 104203.

[74] Zhang, Y., & Weng, W.G. (2020). Bayesian network model for buried gas pipeline failure analysis caused by corrosion and external interference. *Reliab. Eng. Syst. Saf., 203*, 107089.

[75] Upadhyay, V., & Pandey, G.N. (2014). Preventive Maintenance for Over-headed Pipelines with Automated Ultrasonic Thickness Monitoring.

[76] Enani, J. (2016). Corrosion control in oil and gas pipelines. *International Journal of Scientific & Engineering Research, 7*, 1161-1164

[77] Bastian, B.T., N, J., Ranjith, S.K., & Jiji, C.V. (2019). Visual inspection and characterization of external corrosion in pipelines using deep neural network. *NDT & E International*.

[78] Zaman, D., Tiwari, M.K., Gupta, A.K., & Sen, D. (2020). A review of leakage detection strategies for pressurised pipeline in steady-state. *Engineering Failure Analysis, 109*, 104264.

[79] Mousavi, S.M., & Moghaddam, A.S. (2020). Failure pressure estimation error for corroded pipeline using various revisions of ASME B31G. *Engineering Failure Analysis, 109*, 104284.

[80] Salim, B.A., Balland, P., Ahmed, M., & Ahmed, B.A. (2018). Study of the reliability of corroded pipeline by the ASME B31G method. *Modelling, Measurement and Control B*.

[81] Berrekia, H., Benzerga, D., & Haddi, A. (2019). Behavior and damage of a pipe in the presence of a corrosion defect depth of 10% of its thickness and highlighting the weaknesses of the ASME / B31G method. *Frattura ed Integrità Strutturale*.

[82] Soares, E., Bruère, V. M., Afonso, S. M., Willmersdorf, R. B., Lyra, P. R., & Bouchonneau, N. (2019). Structural integrity analysis of pipelines with interacting corrosion defects by multiphysics modeling. *Engineering Failure Analysis*, *97*, 91-102.

7.0 Appendix

Signal sampling code section

```
clear, clc, close all

desired_signal_sampled = 0;
while desired_signal_sampled == 0 % To get the device of the
daq devices = daq.getDevices;

% Create a session to use the daq

s = daq.createSession('ni');

% Add analog input 0

addAnalogInputChannel(s,'Dev1', 0, 'Voltage');

% Sampling rate (max = 400000)

s.Rate = 250000;

fs = s.Rate;

data = s.inputSingleScan;

s.DurationInSeconds = 0.2;

[data,time] = s.startForeground;

%Find the peaks of background noise and desired signal

[peak,location] =

findpeaks(data,time,'MinPeakDistance',0.0005,'MinPeakHeight',2
);

i = 1;

period = location(i+1)-location(i);
```

```
%Detect the peak of desired signal

a = 0;

overlap = 1;

while a == 0

    if(round((location(i+1)-location(i)),3))<0.008

        if(round((location(i+1)-location(i)),4))<0.0042

            marked_location_back = location(i+1);

            marked_location_front = location(i);

            break;

        elseif(round((location(i+1)-location(i)),4))>0.0042

            marked_location_back = location(i+1);

            marked_location_front = location(i);

            desired_signal_sampled = 1;

break;

        elseif(round((location(i+1)-location(i)),4))==0.0042

            marked_location_back = location(i+1);

            marked_location_front = location(i);

            desired_signal_sampled = 1;

break; end


    elseif i == 24

       i = 0;

end

i = i + 1;

end
```

```
% need to consider whether the desired signal is the first
signal of the scan

no_marked_front = (marked_location_front * s.Rate) + 1;

no_marked_back = (marked_location_back * s.Rate) + 1;

[check_back_peak,check_back_location] =

 findpeaks(data((no_marked_back-75):(no_marked_back

+75)),time((no_marked_back-75):(no_marked_back

+75)),'MinPeakDistance',0.0002,'MinPeakHeight',2);

[check_front_peak,check_front_location] =

 findpeaks(data((no_marked_front-75):(no_marked_front

+75)),time((no_marked_front-75):(no_marked_front

+75)),'MinPeakDistance',0.0002,'MinPeakHeight',2);

[m1,n1] = size(check_front_peak);

[m2,n2] = size(check_back_peak);

if m1 > m2

    no_of_data = no_marked_front;

else

    no_of_data = no_marked_back;

end

% Check if there is any background signal overlap with the
reflected

 parts after the excitation part

    if(round(location(i+2)*s.Rate) - round(no_of_data)) < 440

        desired_signal_sampled = 0;

else

       desired_signal_sampled = 1;
```

```
       break;
end end
% Calculate the sequence (nth) of the desired_location
desired_front_temp = round(no_of_data - 110);
desired_end_temp = round(no_of_data + 440);
desired_time_temp = time(desired_front_temp:desired_end_temp);
desired_data_temp = data(desired_front_temp:desired_end_temp);
[m3,n3] = size(desired_data_temp);
%To detect the first peak of the excitation signal
x = 1;
while x < m3
    if desired_data_temp(x) > 1.8
       break;
end
x = x + 1;
end
desired_time = desired_time_temp(x:x+328);
desired_data = desired_data_temp(x:x+328);
%Carry out STFT
f = 3000;
window = 25;
[s,f,t] = spectrogram(desired_data,window,[],f,fs);
magnitude = abs(s);
final_data = magnitude(223,:);
csvwrite('desired_signal.dat',final_data);
```

Support Machine Vector Code

```
import pandas as pd

import numpy as np

from sklearn import svm

import matplotlib.pyplot as plt

# Read training data

train_dataframe = pd.read_csv('Pipe_56_train1.1.csv')

print (train_dataframe)

# Prepare data and separating labels from features

train_labels = train_dataframe.Class

labels = list(set(train_labels))

train_labels = np.array([labels.index(x) for x in train_labels])

train_features = train_dataframe.iloc[:,1:]

train_features = np.array(train_features)
```

```
# Training classifier with parameters

classifier = svm.SVC(C=0.1, cache_size=200, class_weight=None,
coef0=0.0,
        degree=1, gamma=0.01, kernel='rbf', max_iter=-1,
probability=True,
        random_state=None, shrinking=True, tol=0.001, verbose=False)

classifier.fit(train_features, train_labels)

# Read test data

test_dataframe = pd.read_csv('Pipe_56_test1.1.csv')

# Prepare data and separate labels from features

test_labels = test_dataframe.Class

labels = list(set(test_labels))

test_labels = np.array([labels.index(x) for x in test_labels])

test_features = test_dataframe.iloc[:,1:]

test_features = np.array(test_features)

# Run test data on trained classifier

results = classifier.predict(test_features)

num_correct = (results == test_labels).sum()
```

```
recall = num_correct / len(test_labels)



# Print results

print ()

print ("Test Labels")

print (test_labels)

print ()

print ("Results")

print (results)

print()

print ("Model Accuracy (%):", recall*100, "%")

print()
```