A Method for Defect Detection and Characterisation through Magnetic Flux Leakage Signals Using 3D Magnetoresistive Sensors

JESUS DAVID BERNAL-MORALES

A Method for Defect Detection and Characterisation through Magnetic Flux Leakage Signals Using 3D Magnetoresistive Sensors

Jesus David Bernal-Morales

A Thesis presented for the degree of
Master in Science by Research

Department of Engineering
Durham University
United Kingdom

June, 2020

Dedicated to

My family, my friends, and all those people who are doing everything they can to help others.

A Method for Defect Detection and Characterisation through Magnetic Flux Leakage Signals Using 3D Magnetoresistive Sensors

Jesus David Bernal-Morales

Submitted for the degree of Master in Science by Research
June, 2020

Abstract

Health monitoring of large pipeline networks is of great importance for any country or industry. The malfunctioning of these networks may produce economic losses as well as environmental crisis. Non-Destructive Evaluation (NDE) techniques are implemented to ensure proper monitoring of pipeline networks without interfering with their operation. The Magnetic Flux Leakage (MFL) method utilises magnetic fields to detect cracks and corrosion defects on the surface of a pipe, MFL signals are then processed to characterise the detected defects. Recent research has focused on using monopolar Hall-Effect sensors to collect MFL data in order to detect regular-shaped defects. This thesis proposes a method that uses image-processing techniques for defect detection and size estimation. An experimental setup is designed in order to collect MFL signals using 3D GMR sensors and reconstruct 2D images. The approach is then replicated in a simulated environment and is tested with irregular-shaped defects in order to evaluate its accuracy with non-standard defects and attempt depth estimation by designing a novel mathematical depth function that uses the average strength of MFL signals.

Declaration

The work in this thesis is based on research carried out at the Sustainable Infrastructure Research Challenge, the Department of Engineering at Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

Firstly, I would like to show my gratitude to the Mexican Council of Science and Technology (CONACyT), without their financial support I would not have been able to come to Durham University to pursue a postgraduate degree. Secondly, my gratitude goes for my supervisor, Dr Qing Wang, who has been very supportive and attentive during the progression of my research, and also for the Department of Engineering and the technical staff who allowed and helped me use the facilities needed to conduct my investigation. My friends Josue, Mariana, Dalia, Martha and Ivan also played an important role on this path, without their constant help and camaraderie obtaining this degree would have been more difficult. Finally but most importantly, my infinite gratitude goes to my parents and siblings, who have been my pillar of strength for my entire life and who have been by my side helping and supporting me in any imaginable way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Health monitoring of structures is a commonly addressed challenge in engineering in order to develop techniques that can accurately predict potential risks and help in the decision making of carrying out pertinent maintenance. Pipeline networks are an example of large structures with a high level of importance since they can be found in a variety of locations such as in industries, hospitals, and houses in a smaller scale, as well as the larger pipeline networks of an entire city or country. Furthermore, most of hydrocarbon transportation is made via pipelines due to the reliability and viability of this method in terms of the economy of a country. According to a report published in 2019, the United Kingdom has a vast pipeline network that covers over 38000 km, which mainly consists of gathering, transmission, and distribution lines of oil, gas, and refined products [1]. Moreover, the United Kingdom's oil and gas revenues produced 316 million pounds for the 2016-17 fiscal year [2], and recent reports indicate that these revenues increased to 1.1 billion pounds for the 2017-18 fiscal year [3, 4]. Consequently, failure in pipeline systems can be costly not only through direct losses (repair or replacement costs), but also through indirect losses such as public safety and environmental impact, e.g. the Buncefield Incident [5]. The size and economic importance of this infrastructure makes imperative the need for proper health monitoring techniques in order to ensure its integrity and optimal operation.

The development of such monitoring techniques represents a challenge for engineering researchers and is the motivation of this thesis. Furthermore, this task

becomes more complex since the majority of oil and gas pipelines are either underground or under water, thus special tools and techniques must be applied for their inspection without affecting the operation of the system. This thesis investigates Non-Destructive Evaluation (NDE) or Non-Destructive Testing (NDT) techniques and concentrates on the Magnetic Flux Leakage (MFL) method in order to develop a methodology for the detection and size estimation of metal loss defects on the pipe surface. These defects are produced by corrosion on the pipe wall, and accurate detection and quantification is needed in order to avoid leakage of the products being transported (e.g. oil, gas, petrol).

In order to develop a methodology to detect and estimate the size of metal loss defects, simulated and experimental work was conducted. The proposed methodology implements image processing techniques such as edge detectors and morphological operations to carry out defect detection and characterisation of irregular-shaped defects. The three orthogonal components of MFL signals are investigated in order to reconstruct 2D images. Initially, these signals are generated using the Finite Element Method (FEM) in a simulated environment. This approach is then tested with an experimental setup in which MFL signals are collected using an array of 3D Giant Magnetoresistive (GMR) sensors in order to validate the methodology. The work presented in this thesis has been conducted pursuing the objectives described in the following section.

1.1 Research Aims

The MFL method has been utilised for research on pipeline monitoring techniques under numerous approaches that include computational perspectives, experimental work as well as signal processing techniques. However, some of said previous research have only focused on one aspect to study the MFL method, either by only exploring numerical simulations, MFL signals processing and analysis, or experimental approaches. Moreover, the focus has mainly been on regular-shaped defects (i.e. rectangular, ellipsoidal) as well as on measuring only one of the MFL components. The present thesis will look into the MFL method with a combination of simulated

and experimental modelling. The experimental model will work as the foundations of this study in order to design, implement and test a measuring tool that uses 3D GMR sensors and collect relevant data. The simulated model will be used to validate de experimentation and compare the generated data in order to develop an image-processing based method to carry out defect detection and quantification. For this reason, three main objectives are pursued here:

1) The use of 3D GMR sensors to collect MFL data for each of the three MFL components.

2) Detection of defects of irregular shape on the surface of a pipe by analysing the three MFL components to reconstruct 2D images.

3) Accurate size estimation of irregular defects by applying image processing techniques.

Additionally, a brief description of the activities conducted in this thesis is listed below:

1. A general review of previous work related to corrosion defects detection and characterisation using the MFL method.

2. An FEM simulation in which MFL data is generated using the software COMSOL Multiphysics is presented.

   (a) A modification of the magnetiser system is proposed to adhere to the conditions of the real pipe used for the experiments in this research project.

   (b) Different types of defects will be studied to provide a comparison of the generated signals based on the geometry of defects. Regular and irregular shaped defects will be explored.

   (c) Collection of simulated MFL signals as well as their decomposition in three orthogonal components $\boldsymbol{Bx}$, $\boldsymbol{By}$, and $\boldsymbol{Bz}$.

   (d) The effect of the sizes of the pipe and defect over the MFL signals is explored.

3. A methodology to analyse and process the simulated MFL signals.

(a) The simulated MFL signals will be used to reconstruct 2D images from which defects can be detected with an algorithm written in Matlab.

(b) A combination of image processing techniques such as edge detectors and morphological operations is implemented to carry out defect detection.

(c) Size estimation of the detected defects is achieved by geometrical and numerical approximations.

(d) A novel function that describes the relationship between the average MFL signals strength and defect's depth is described.

4. An experimental model to validate the results drawn from the simulations.

(a) Design and modelling of the experimental set-up in which a steel pipe is magnetised in order to carry out MFL inspection.

(b) Design and modelling of the modified magnetiser used to magnetise the steel pipe from outside.

(c) Design and modelling of the sensor system applied to collect MFL data using 3D GMR sensors.

(d) Design and modelling of the motor system to move the magnetiser and sensor system along the pipe surface.

5. A methodology to analyse experimental MFL signals.

(a) Iterative readings, collecting data from several points until a predefined scan surface is covered.

(b) De-noising and signal filtering to remove parasite signals.

(c) Image reconstruction using real MFL signals in order to test the proposed defect detection methodology.

1.2 Thesis Layout

This thesis is grouped into 7 chapters which include the theoretical, simulated and experimental aspects of irregular-shaped defects detection on a steel pipe with the utilisation of GMR sensors applying the MFL method.

June, 2020

Chapter 1 establishes the importance of health monitoring through NDE techniques in order to ensure the optimal operation of product transmission pipelines. The scope, motivation and structure of this thesis is also addressed here.

Chapter 2 includes a brief summary of pipeline health monitoring as well as NDE methods. The theory behind the MFL method is also presented. A general review of previous work regarding the MFL method is also included, and research gaps are identified as well.

Chapter 3 details the design and construction of the proposed experimental model. Design of the needed circuitry for this model is also included. The operation of the proposed experimental setup is described here. Data collection, organisation and storage is also addressed.

Chapter 4 addresses the processing, analysis and denoising of data generated in the experimentation. The principle of the proposed image-processing based method is described here.

Chapter 5 contains a detailed description of the proposed simulated system to detect defects on the surface of a pipe. A brief review of the software used to carry out the simulations is also given here. The algorithm proposed to analyse the simulated MFL data is described here. Width and length estimation of the detected defects is addressed here. A new mathematical function to estimate defect's depth is described in this chapter.

Chapter 6 discusses the results obtained from both the simulations and the experiments. The challenges of both aspects are also addressed here.

Chapter 7 summarises the present research and provides the conclusions derived from it as well as its contributions to the field. Suggestions for further work are given.

Chapter 2

Literature Review

2.1 Pipeline Health Monitoring

Pipeline health monitoring is achieved by the utilisation of NDE methods to detect anomalies that may be caused by long time deterioration or other external factors such as humidity or pressure. Depending on the nature of the specimen, there is a wide variety of NDE techniques available that can be used to determine its health state, ranging from simple methods such as Eye Inspection or Leak Testing [6] to more complex and specialised techniques such as Eddy Current Inspection, Radiographic Testing, Ultrasonic Testing, and Magnetic Flux Leakage, to name but a few. The common applications, advantages and limitations of each of these methods are shown in Table 2.1 [6–8].

Since pipelines are mostly buried or underwater, some of the aforementioned NDE methods may not be easy to implement for pipeline health monitoring. However, the MFL method has been widely applied using special tools and techniques called In-Line Inspection (ILI) [9]. These tools are also named Smart Pipeline Inspection Gauges (Smart PIGs), and their operation consists of travelling inside the pipe while gathering relevant data. Fig. 2.1 shows an example of a PIG tool [10]. Industrial PIG tools combine several types of sensors and techniques in order to acquire as much information as possible so that the accuracy of flaw detection increases. The focus of this thesis is on the PIG tools in which the main inspection technique is the Magnetic Flux Leakage (MFL) method for metal loss defects on the

surface of a pipe due to corrosion. The theory behind the MFL method is described in the following section.

Method	Applications	Advantages	Limitations
Eddy Currents	- Crack detection - Material thickness measurements - Coating thickness measurement	- Sensitive to small cracks - Detects surface and near surface defects - Immediate results - Contactless inspection	- Only for conductive materials - Surface must be accessible to the probe - Extensive training needed - Surface coating may interfere - Limited depth penetration
Radiographic	- Pressure vessels and valves inspection - Inspect weld repairs - Inspect corrosion under insulation	- Highly reproducible - Appliable to several materials - Little surface preparation needed	- Sensitive films can only be used once - Can be costly - Digital radiography is subject to image quality constraints
Ultrasonic	- Flaw detection - Dimensional measurements - Material characterization	- Sensitive to surface and subsurface discontinuities - Big depth penetration - High accuracy for location, size and shape estimation - Detailed images can be produced	- Surface must be accessible - Extensive training required - A coupling medium between transducer and specimen is needed - Difficult inspection of materials that are rough, irregular, thin, small or non-homogeneous
Magnetic Flux Leakage	- Crack, corrosion and pitting detection - Pipeline and fuel tanks inspection	- Detection of several types of defects - High-speed inspection - Inspection inside pipe - Defect detection inside and outside pipe	- Appliable only to ferromagnetic materials - Huge amounts of data to be analysed

Table 2.1: Comparison between NDT techniques

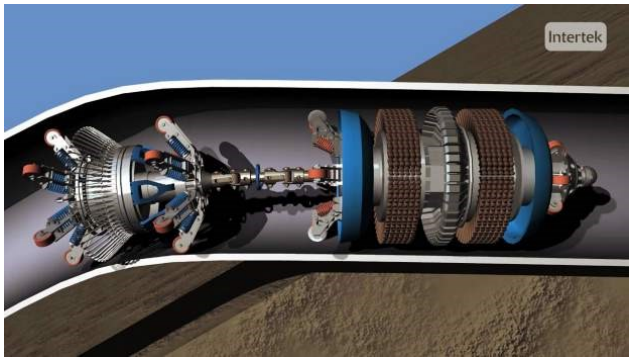


Figure 2.1: PIG tool

2.2 Theory of the MFL Method

The MFL method has been evolving since the 1960s and is one of the NDE techniques most commonly used for the inspection of oil and gas pipelines in the search for

metal loss defects due to its high accuracy [11]. The principle of the MFL method relies on the high-permeability property of ferromagnetic materials. This property allows the material to be magnetised by an external uniform magnetic field produced by a permanent magnet or a coil. When a ferromagnetic material (e.g. a steel pipe) is magnetised, a magnetic circuit is created in which the magnetic field lines are constrained in the material and flow through it. Contrarily, when there is a discontinuity on the material due to a metal loss defect, the magnetic field lines will leak out of the surface. This magnetic leakage can be measured by magnetic sensors such as Hall-effect sensors or GMR sensors [11–16]. The principle of the MFL method is summarised in Fig. 2.2.

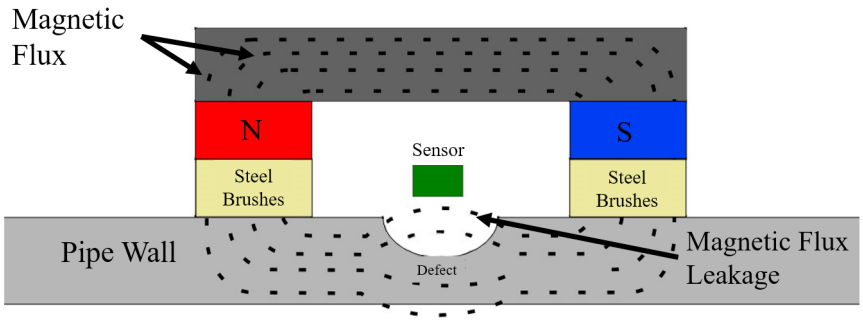


Figure 2.2: Magnetic Flux Leakage Principle

The MFL method is governed by Maxwell's equations for static magnetic fields, which establish the relationship between the *magnetic flux density* and the *magnetic field intensity*. Maxwell's equations are shown below:

$$\nabla \times \vec{H} = \vec{J} \tag{2.2.1}$$

$$\nabla \cdot \vec{B} = 0 \tag{2.2.2}$$

where $\vec{B}$, $\vec{H}$ and $\vec{J}$ are the magnetic flux density, the magnetic field intensity, and the current density, respectively. Furthermore, the aforementioned relationship between the magnetic flux density $\vec{B}$ and the magnetic field intensity $\vec{H}$ is given by

$$\vec{B} = \mu \vec{H} \tag{2.2.3}$$

where μ is the relative permeability of the material in which the magnetic fields flow, and is defined as the ability of a given material to be magnetised by an external

magnetic field. This relationship indicates that $\vec{B}$ depends on the changes of $\vec{H}$ as well as the variation in the relative permeability μ. Additionally, this relationship is affected when ferromagnetic materials are magnetised by an external magnetic field (e.g. permanent magnets, coils) [12, 17, 18], consequently (2.2.3) is modified as follows

$$\vec{B} = \mu \left(\vec{H} + \vec{M} \right) \tag{2.2.4}$$

where $\vec{M}$ is the magnetisation vector. According to the MFL principle previously described, when a magnetic field is applied to a ferromagnetic material with a certain μ value and there is a discontinuity on the material's surface, μ will vary at this location due to the different medium in which the magnetic flux is passing through (e.g. gas, oil, air), making the magnetic flux leak to this other medium and thus detectable by a magnetic sensor.

2.3 Influencing factors in MFL Inspection

The MFL inspection method is not exempt from being influenced by external factors. Examples of these influencing factors that affect the efficiency of the monitoring method include the thickness of the material, the distance between the magnetic poles, the magnetic remanence, variations in the liftoff of the sensors (i.e. the distance between the specimen and the sensors), and the scanning speed, but also the sensitivity of the sensors, the magnetising field, and noise have to be considered [11]. Furthermore, the probability of detection of pitting defects using MFL inspection can also be affected by unmaintained equipment, untrained operators and also the cleanliness of the scanning surfaces has an impact on the accuracy of the method. "With well-maintained equipment, trained and conscientious operators working on clean unpitted scanning surfaces on material thicknesses up to 10mm thick losses of 20 %(sometimes as low as 10%) can be reliably detected." [19]

To this extent, attempts to enhance the signal-to-noise ratio (SNR), which is the ratio between relevant signals with respect to noise signals, have been explored in [13]. In their approach, they collected MFL data from a seamless steel pipe, implemented a normalised least mean squared (NLMS) adaptive filter, and finally

applied wavelet denoising techniques to remove the remaining noise. Similarly, a signal compensation methodology was presented in [14] under the argument that the movement of the PIG induces parasite magnetisation on the pipe wall. Their approach uses a combination of neural networks, LMS adaptive filters and wavelet shrinkage denoising in order to clean the signals. Another research that analyses MFL signals from seamless pipes is presented in [20]. Here, signals from individual sensors are compared with a prediction of said signals in order to estimate the error and detect defects with it.

Furthermore, regarding the sensing speed, reference [21] mentions that the sensors' optimal location for high speed applications had been overlooked and for that reason proposed a 2D FEM simulated study which explored the effect of different moving speeds and found that the MFL signals distort due to the induced magnetisation. Additionally, these influencing factors are mentioned in [22] but the emphasis is on the scanning speed. Here, a spherical defect was analysed in a 2D simulated environment in order to compare the average magnetic flux in both static and moving scenarios. Moreover, the influence of the orientation of defects over the MFL signals was explored under the argument that this orientation hinders defect detection.

As stated in [23], because the intensity of MFL signals is strongly dependent on the liftoff of the sensors, the variations in this parameter produce distortion on MFL signals. For that reason, they proposed a liftoff compensation method with which they attempted to improve the SNR considering the influence of the defect region and non-defect region over the MFL signals. Alternatively, the research presented in [24] looked into the effect of different magnetisation strength over the generated MFL signals. To achieve this, they considered the MFL signals as a function of the defect's depth, arguing that this feature is strongly related to these signals. In their approach, they proposed a magnetiser with multiple stacked rectangular permanent magnets. The gap between each magnet was varied, and also the distance between the poles and the liftoff were modified. However, their measurements were limited to only one of the three MFL components.

As described in this section, most of the cited publications studied the impact of

different influencing factors over the MFL signals, and looked at this from a signal processing perspective, making evident the importance of signal denoising. The experiments conducted in the present research considered the sensing speed and the liftoff of the sensors to evaluate their effect over the MFL signals.

2.4 Defects Detection Through MFL

The MFL method has been investigated from different perspectives, focusing on various aspects. An example of this is the Magnetic Dipolar Method (MDM) used in [25], where a multiple neighbouring defects case was explored since past work had been limited to the study of individual simple defect geometries. Here, the interaction of the neighbouring defects and their effect over the MFL signals was explored, and accurate length estimation of defects was achieved. However, since this work concentrated on 2D defects, the study was expanded in [26] to predict the three orthogonal components of 3D defects. Here, a circular defect was examined and the results were used to conclude that the pattern of the MFL signal components can be used to predict the shape of the detected defects. In order to validate their conclusions, this work was extended in [27] to include simulations as well as looking into the effect of different lift-off values over the definition of the signals. Furthermore, arguing that the focus had been only on the $\boldsymbol{Bx}$ and $\boldsymbol{Bz}$ components of the MFL signals, they also investigated the contribution of the $\boldsymbol{By}$ to predict the location of defects as well as facilitate the characterisation of irregular defects. Finally, in order to study defects of irregular shape, they improved their model in [28] and validated their results with simulations.

The MDM was also applied in [29], and the MFL response obtained with this method was compared to that of the Finite Element Method (FEM) in order to test the efficiency of the proposed Basic Signal Combination Method (BSCM). This research focused on individual line scans (basic signal) and studied rectangular defects to find the relationship between the basic signal and the geometry of the defects. As the authors mention in their publication, their contribution could be used to create a database of common defects with their basic signals. However, their method is

limited to signal processing and, according to [30], the robustness of systems that rely on signal processing techniques is affected by noise in the measurements, and for this reason they applied image processing techniques such as edge detectors and convolutional neural networks (CNN) (i.e. a type of neural network specialised in extracting features from images). Furthermore, their work pursued a differentiation of injurious defects (i.e. crackline anomalies and metal losses) from non injurious defects (i.e. noise events, pipe deformations and manufacture irregularities) arguing that an incorrect identification of the detected defects in a pipe could result in a waste of time and resources, in the best case scenario, or hydrocarbon leakage in the worst case. The CNN approach was tested later in [31] in order to compare it against a new visual transformation CNN, which showed improvements in defect size estimation but limited to a specific size range. Finally, this work was expanded in [32] in order to classify the different MFL responses after inspection is completed.

Similarly, the work presented in [33] concentrated on identifying two types of defects: bump defects and concave defects. They followed this approach under the idea that bump defects had been overlooked and all the inspection equipment is calibrated for concave defects. Among their findings is that the bump defects produce negative magnetic leakage, i.e. the basic signal for this type of defect is the inverse of that of concave defects. They studied scenarios with rectangular concave defects, rectangular bump defects, and a combination of a rectangular bump defect with a rectangular concave defect on it, and proposed a methodology for obtaining the overall MFL signals for these scenarios.

Moreover, another MFL imaging method is presented in [34] in order to study the three components of MFL signals. According to this research, most present MFL inspections consist of 1D MFL measurement, which is simpler than 3D MFL measurement since the latter needs more sensors, making the MFL tool more complicated. However, they also state that it is crucial to measure the 3D signals since they contain information about the detected defects. For this reason, they proposed a methodology in which only one component is measured ($\boldsymbol{Bz}$) and the other two are calculated by applying partial derivatives to the measured signal. They tested their method by predicting the 3D MFL signals of a rectangular defect using the MDM

approach and compared this with the components that were calculated with partial derivatives. After the three components are obtained, image processing techniques (image binarisation) are used to carry out defect detection.

From the previously cited studies, the importance of decomposing MFL signals into their three components is crucial in order to develop accurate and efficient monitoring techniques. Additionally, most of the research reviewed in this section concentrated on regular-shaped defects and overlooked irregular-shaped defects. Finally, it can be concluded that MFL imaging methods have been taking over MFL signal processing methods.

2.5 Defects Characterisation Through MFL

The most important aspect of MFL inspection is to determine the size and depth of the detected defects. To that extent, the work presented in [35] and [36] proposed a methodology to estimate the Maximum Safe Operating Pressure (MAOP) of a pipe. As they mention in their publications, this parameter can be calculated by analysing the geometry of the detected defects, in other words by estimating their length, width and depth. Additionally, they state that the emphasis should be on the depth of defects since this feature has a greater impact on the safe operation of pipes. They proposed a methodology that provides a 2D representation of the measured signals by applying Haar wavelets and the Discrete Wavelet Transform. Furthermore, they also used Radial Basis Function Neural Networks to predict MFL signals for a given depth profile. They concluded that the results of their research can increase the accuracy of MAOP estimation.

A different experimental investigation presented in [37] used a simulated MFL model in order to design an image-based detection algorithm and validate it with MFL data collected from a real pipe using a PIG tool with a combination of Hall-effect sensors and coils. Moreover, different types of standardised defects were studied here: General, Pitting, Axial Grooving, and Circumferential Grooving, according to the definitions provided by [38]. Later, this research was extended in [39] with a width estimation method that was tested with both simulated and experimental

MFL data. Although the same types of defects were studied here, the main focus was on ellipsoidal defects. Defect detection was achieved in both the simulated and experimental models and the proposed width estimation algorithm produced accurate results. Finally, continuing this line of research, a length and depth estimation methodology was proposed in [40]. Once more, the same defect categories were examined, but mostly ellipsoidal defects were studied. Defect length was estimated by examining the contour plots of the MFL signals. Depth estimation was achieved with the use of a Gaussian Radial Basis Function Neural Network with the estimated width and length, and the signal peak values as inputs.

Moreover, a similar methodology was followed in [41], where the quadratic relationship between the defect's depth and the peak amplitude of the MFL signals was explored in order to design an equation with which the depth of defects could be estimated depending on the estimated width and length. Later, this depth estimation method was improved in [42], adding data normalisation and data averaging steps. However, both the original and enhanced methods depend on a lookup table from where the shape factors of standard defects are taken as the coefficients of the proposed Depth Equation and modified iteratively by polynomial surface fitting. Analogous to this, an almost linear relationship between the average strength of the MFL signals and the defect depth was explored in [43]. With this finding, the depth of defects could be accurately estimated via linear fitting. However, this study focused on flat surfaces and square defects of regular depths.

Finally, another experimental method is presented in [44] following an imaging approach in order to quantify defects of complex shapes and irregular depths. They used four different orientations for the magnetisation of the specimen and obtained images from MFL signals. However, these images are not created from MFL signals. They explain that these images are captured using a CCD camera, which is triggered when a magnetic sensor detects magnetic leakage.

With the reviewed publications in this section it can be concluded that imaging methods have been popular in MFL research to characterise corrosion defects. Additionally, approaches that use artificial intelligence techniques have been followed in those experimental research that obtained large amounts of MFL data. However,

most of the publications cited here concentrated on regular and standardised defects.

2.6 Measurement of MFL Signals

As previously stated, the experimental publications cited in the previous sections have used industrial-size PIG tools with large sensor arrays, allowing them to collect huge amounts of samples, which explains why these approaches utilised neural network techniques. Different types of magnetisation techniques were also explored, as shown in [45], which comments on the effects of axial and circumferential magnetisation over metal loss defects.

Furthermore, the majority of these experimental investigations have in common the use of Hall-effect sensors to carry out their measurements. According to [46], this type of sensor is cheap and reliable under tough temperature conditions. However, he also mentions that, apart from the Hall-effect, there are several other methods to measure the MFL signals. Examples are electromagnetic induction and the magnetoresistive effect, as well as magnetic resonance imaging and magneto optical methods. Although Hall-effect sensors are the most commonly used, they also have a high noise level and low sensitivity, thus requiring amplification circuitry. Additionally, the magnetic fields should be perpendicular to the device so several sensor arrangements have to be used in order to detect the three orthogonal components [47, 48]. An example of these specific sensor arrangements can be seen in [49] where a combination of the MFL method with the Pulsed Eddy Current method is proposed for high speed inspection applications.

For these reasons, the use of Giant Magneto Resistive (GMR) sensors was explored in [48]. Here, this type of sensor is described as a multilayer structure containing magnetic and nonmagnetic thin layers arranged in a specific way so that there is a unipolar response to the applied magnetic field. A similar approach was followed in [50], where a GMR probe was used to detect MFL signals of a ferrous plate that was axially magnetised, under the idea that defects cannot be detected when they are aligned with the magnetic flux. However, they focused their research on detecting regular-shaped defects using monopolar GMR sensors. Furthermore,

monopolar GMR sensors were also used in [51] in order to perform MFL inspection on a steel pipe in a similar manner as the publications cited before. With the advancement and the improvement of nanotechnology techniques, smaller layers are achieved, hence smaller GMR chips are manufactured with the ability of measuring the three components of magnetic fields such as the experiment performed in [52]. Here, 3D GMR sensors were used to study their possible advantages in MFL inspection. However, this study concentrated on cracked rail heads with long rectangular cracklines.

As described in this section, most of MFL inspection is conducted by using Hall-effect sensors although they are unipolar and the importance of measuring the three MFL components has been addressed.

2.7 Research Gaps

The previous general review of past MFL research makes evident that the focus has mainly been on the study of regular-shaped defects such as rectangular or ellipsoidal defects. These studies have been conducted by collecting MFL data using arrays of Hall-effect sensors. Signal processing techniques have been implemented in order to find the key features that define the presence of defects on the specimen's surface. However, most of these signals were obtained using unipolar magnetic sensors, despite the need for analysing the three MFL components. Attempts to measure the three MFL components have been made by combining Hall-effect sensors in specific arrangements so that they are perpendicular to each of the components, however this implies the use of one sensor chip for each magnetic axis, which leads to huge amounts of sensors in one PIG tool. Additionally, although 3D GMR sensors have been used before in MFL inspection to measure the three components with a single chip, this research has concentrated on detection of regular-shaped defects as well as their orientation [50].

For these reasons, this thesis presents a method in which 3D GMR sensors are used for MFL inspection in order to find irregular-shaped defects and estimate their size. Signal processing techniques will be used to clean the measured signals but

afterwards imaging techniques will be applied to detect and characterise the aforementioned defects. Moreover, it should be noted here that the conditions of the studied specimen in this thesis are different than those of the pipes analysed in the cited publications, where large pipeline structures were scanned. Conversely, the pipe used to carry out this investigation is sufficiently small to implement a PIG tool similar to those shown before in this chapter. Therefore, a modification of the regular MFL inspection tool will be designed in order to determine whether the MFL method can be effectively applied from the outside of the pipe, having more control over the sensing speed and liftoff value.

The following chapter describes the design of the proposed 3D-sensor based system as well as its implementation in an experimental setup with a steel pipe.

Chapter 3

Design and Implementation of a 3D-Sensor Based System

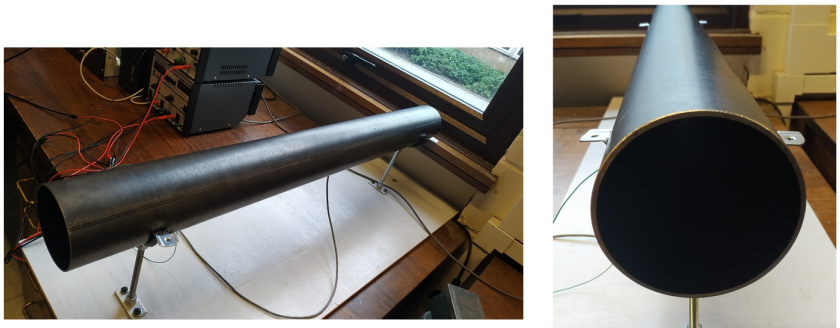
An experimental model is proposed which is based on the idea of measuring the three components of the magnetic fields with the use of 3D GMR sensors. This experimental model includes a steel pipe that is magnetised by a pair of permanent magnets joined by a steel bar. The magnetiser is mounted on a threaded rod and moves along the pipe with the help of a motor coupled to the rod, while a set of GMR sensors collects the generated MFL signals. These signals are logged into a PC using an Arduino UNO board via serial communication. The following sections will describe the construction process of the aforementioned experimental model including a description of the parameters used for it and the parts that it contains. A description of the operation of the model is also given in this chapter. The design and construction of the model presented in this chapter was conducted entirely by the author of this thesis, unless otherwise stated.

3.1 Construction Process

3.1.1 Steel Pipe

The studied specimen in this experimentation is a carbon steel pipe that can be seen in Fig. 3.1. The pipe is 1 m long and has an outer diameter of 101.6 mm with a

wall thickness of 2.5 mm. Moreover, this pipe has a rectangular crack that will be used for the initial test. The crack is shown in Fig 3.2.



a) b)

Figure 3.1: Steel Pipe. a) Length of pipe. b) Diameter of pipe.

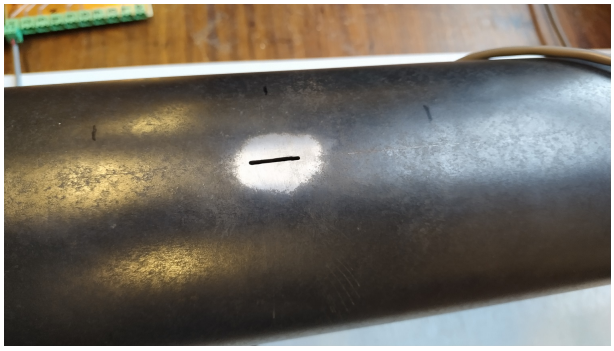


Figure 3.2: Rectangular crack on pipe

3.1.2 Magnetiser System

In order to magnetise the pipe described in the previous section, a magnetiser system was modelled using Solidworks. This magnetiser includes a pair of permanent magnets joined by a steel yoke and two steel pieces that work as the steel brushes in the regular MFL principle described in Chapter 2. The magnets used here are N52 NdFeB magnets with a pull force of 15.1 kg, which means they are strong enough to magnetise the pipe near saturation. The magnetiser system was constructed depending on the dimensions of the magnets, which are 40 mm $\times$ 20 mm $\times$ 5 mm. These magnets were attached to each end of a 92 mm $\times$ 40 mm $\times$ 12 mm steel bar, and the steel brushes are attached on the other face of each magnet. Since the pipe used for this study is small, construction of a PIG system similar to those found in

the literature was difficult and for this reason these brushes were modelled in such a way that the circular section on them has the same radius than the pipe, meaning that the magnetiser will move on top of the pipe in the axial direction. This modification was used in order to investigate the efficiency and reliability of the MFL method applied from the outside of a pipe. The modified magnetiser previously described is shown in Fig. 3.3.

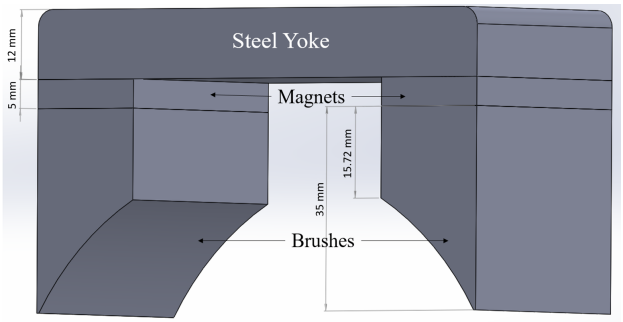


Figure 3.3: Magnetiser System

Furthermore, in order to achieve the axial movement of the magnetiser system on top of the pipe, a special holder was designed for it. This magnetiser holder is shown in Fig. 3.4.

Figure 3.4: Magnetiser Holder

3.1.3 Magnetiser Driver and Pipe holder

As mentioned before in this thesis, the MFL method is applied in the industry using PIG tools. However, due to the conditions of the pipe used in this study, the method had to be modified as described in the previous section with the magnetiser moving on top of the pipe. To achieve this movement, a driving system was created, which consists of a motor coupled to an M10 threaded rod so that the rotational movement is translated into linear movement, following the endless screw simple machine principle. The magnetiser holder previously shown is clamped to the threaded rod, whereas the pipe rests in its own holder. This setup is shown in Fig. 3.5, including the pipe, the threaded rod, the magnetiser system on its holder, and the motor coupled to the rod and fixed with a clamp.

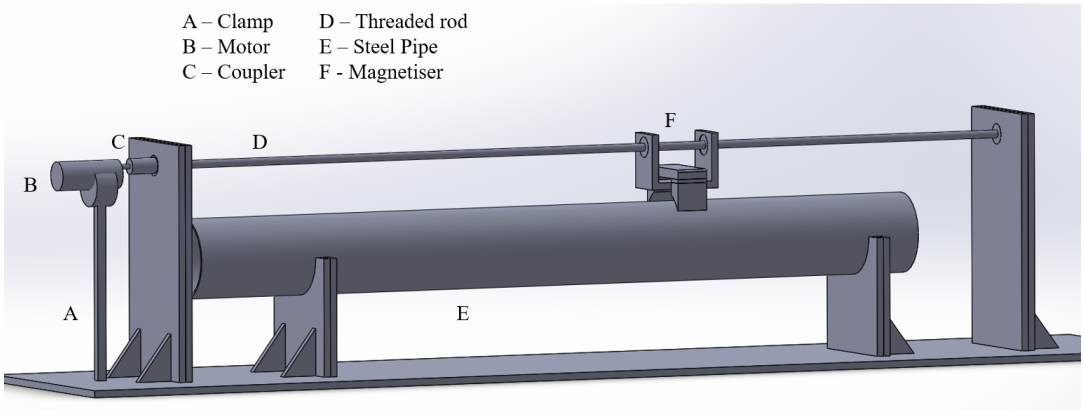


Figure 3.5: Complete experimental model

3.1.4 Model Assembly

Once the CAD model was finished, the materials were purchased and the assembly was conducted in the Mechanical Workshop by the technicians in charge. Fig. 3.6 shows the various components of the experimental model, including the freshly cut wooden pieces that support the pipe and the magnetiser holder in Fig. 3.6(a), the magnetiser holder with the threaded plastic bushes that fix it to the threaded rod in Fig. 3.6(b), the magnetiser system in Fig. 3.6(c), and the right end of the threaded rod clamped with two M10 hex nuts in Fig. 3.6(d). Moreover, as it can be seen in Fig. 3.6(e), there is a difference with the pieces that support the pipe from the original design, this was changed in order to modify the height of the pipe and thus vary the distance between the pipe and the sensors, i.e. the lift-off value.

3.1.5 Sensor System and PCB Design

As previously stated, past MFL research has concentrated on the use of Hall-effect sensors, although they are monopolar and need to be arranged in different configurations so that they can measure the three components of the MFL signals. For this reason, 3D GMR sensors are used here since they can measure the three MFL components with a single chip.

The sensors used here are the LSM9DS1, manufactured by ST Microelectronics, which are embedded in the MIKROE-1996 Inertial Measurement Unit (IMU) made by MikroElektronika. The chip on this board contains a combination of three types

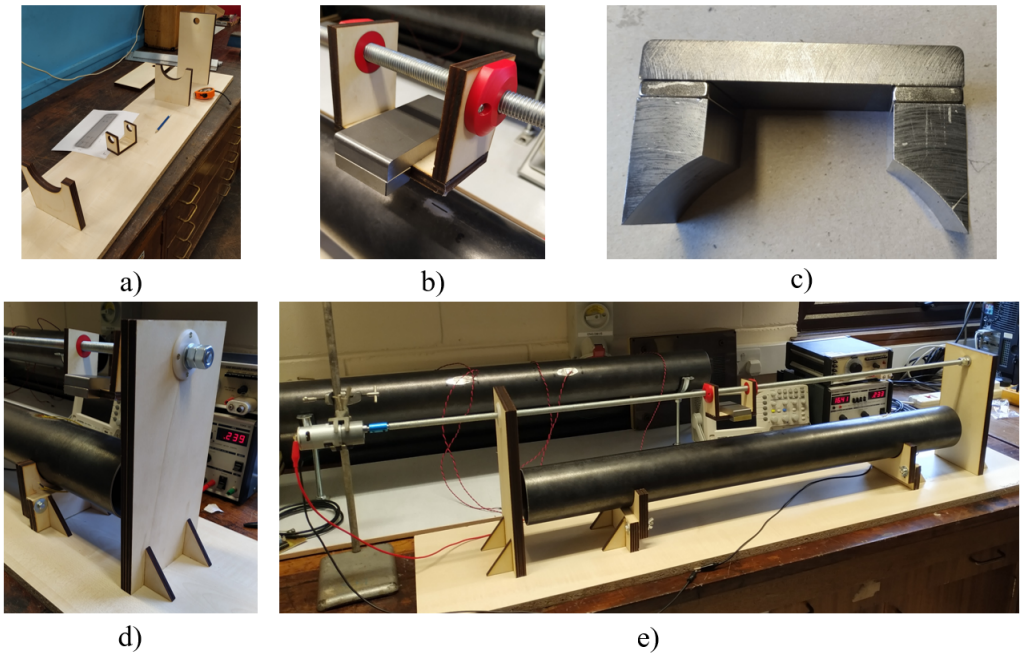


Figure 3.6: Experimental Model Assembly. a) Wooden cut out pieces. b) Magnetiser on holder with bushes. c) Magnetiser. d) Clamped threaded rod. e) Complete model

of sensors, including a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer. This combination of sensors allows the board to be used in navigation applications, smart user interfaces or virtual reality input devices. However, for the purposes of this thesis, only the 3D magnetometer will be used. The MIKROE-1996 IMU operates within a supply voltage range of 1.9 V and 3.6 V, has I^2C serial communication capabilities, and allows the user to enable the three sensors separately. The magnetometer's full scale sensitivity ranges from ± 4 to ± 16 gauss. The board also includes internal amplifiers and a 16-bit A/D Converter, as shown in the block diagram contained in Fig. 3.7. Here, the three GMR sensors can be seen on the left hand side in a Wheatstone Bridge configuration [53].

Two IMU boards are used in the proposed experimental model, they can be seen in Fig. 3.8 mounted on a special board that fits on top of an Arduino board. This setup was used for the initial tests with the sensors. A short code was written in the Arduino IDE to drive the sensors and verify their correct operation. The voltage supply for these two boards comes from the 3.3 V output of the Arduino board. Once the initial tests were concluded, a different printed circuit board (PCB) was designed using Livewire so that the sensors could be located below the magnetiser, separated

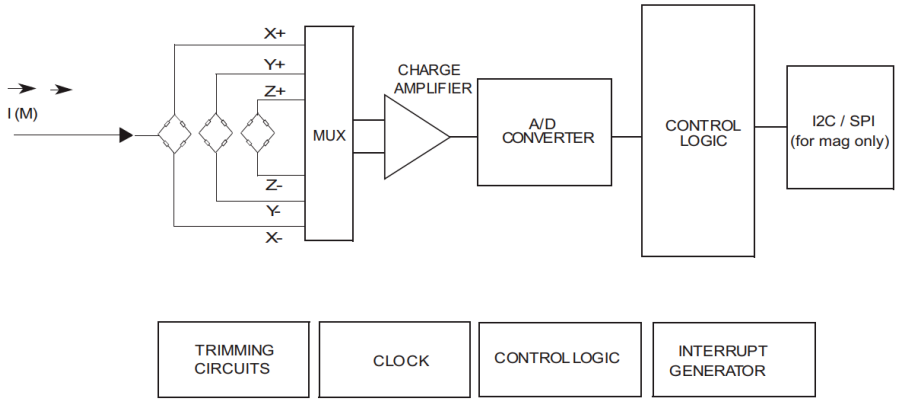


Figure 3.7: Magnetometer Block Diagram

by 10 cm so that the sensor right below the magnets collected direct MFL data whereas the sensor on the far side collected MFL from the residual magnetisation. This board only contains connectors arranged in a way that the sensors are separated so they can measure different parts of the pipe's surface.

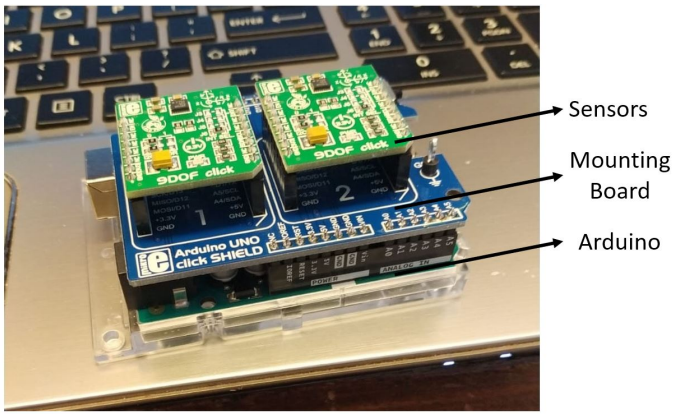


Figure 3.8: Sensors mounted on Arduino board

The motor used in this experimental model is a 919D DC motor that works within a voltage supply range of 4.5 V to 15 V, has a power rating of 11 W, and has a max output speed of 5216 rpm. However, the motor has a gearbox to reduce this speed in a 148:1 ratio [54], which will allow the magnetiser to move at a sufficiently low speed so that the sensors take enough samples of the MFL signals. In order to drive this motor, a second PCB was designed using ISIS Proteus. This board contained the needed circuitry to control the motor's rotation, using an L293D chip that can drive the unidirectional rotation of up to 4 motors, or the bidirectional rotation of 2 motors, with only two logical inputs. The chip needs two voltage sources, Vcc and

Vs, with the former being used for the chip's operation and the latter to power the motor [55]. For the chip to operate correctly, Vcc should be within the voltage range of 4.5 V to 7 V, whereas the voltage for Vs should be between the Vcc value and 36 V. The Vcc supply is taken from the 5 V output of the Arduino board, whereas the Vs supply was set to 15 V and is provided by the Digimess SM3040 DC power supply.

Additionally, two reed switches are implemented to indicate the location of the start and end points of the scanned area. These switches activate (close) when they are near a magnetic field, when this happens a specific instruction is sent from the Arduino to the motor driver. A more detailed description of the operation of the proposed experimental model is given in the following section. Furthermore, other passive elements were used such as resistors and capacitors, and an LED was used to indicate that the motor driver board was ON. This circuitry can be seen in Fig. 3.9, including the schematic diagrams, the PCB diagrams, a 3D model for the motor driver PCB, and the completed PCBs.

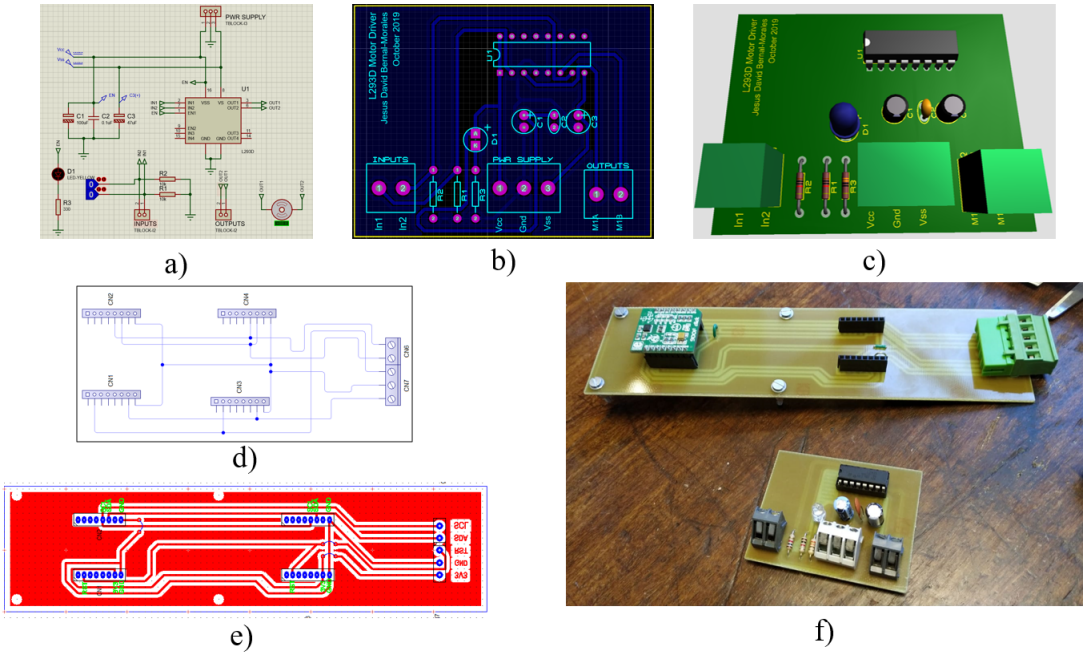


Figure 3.9: Circuitry modelling process. a) Motor driver schematic diagram. b) Motor driver PCB diagram. c) Motor driver 3D PCB. d) Sensors' connector schematic diagram. e) Sensors' connector PCB diagram. f) Completed PCBs

The PCB layouts were sent to the technicians at the Electronics Workshop for

their printing, and the assembly and soldering of components were conducted by the author of this thesis. Once this process was completed, the circuits were tested to ensure their correct operation, and were mounted on the experimental setup afterwards, as shown in Fig. 3.10, which includes the PCB with the two sensors separated by 10 cm below the magnets, the Arduino board connected to a PC, and the motor driver connected to both the Arduino and the power supply.

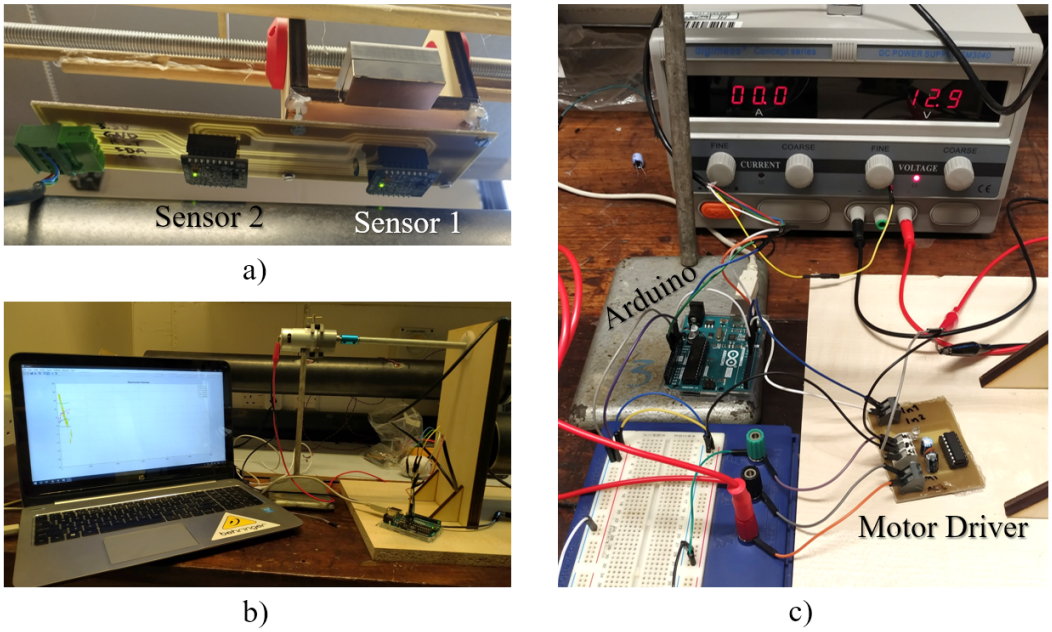


Figure 3.10: PCBs mounted on experimental rig. a) Sensors PCB under magnetiser system. b) Arduino board connected to Matlab. c) Motor driver PCB connected to Arduino and DC power supply

3.2 Basic Operation

The overall operation of the proposed MFL inspection system is controlled by an Arduino UNO board. This device controls the rotation of the motor by sending two control bits to the L293D motor driver board. It also receives the raw measurements made by the GMR sensors, as well as the state of the two reed switches. A serial communication interface between the Arduino device and a PC is created with Matlab. This interface allows the magnetic measurements to be sent from the Arduino device to the PC as well as other useful information for the correct operation of the system. Fig. 3.11 shows a block diagram of the proposed system and how these elements are interconnected.

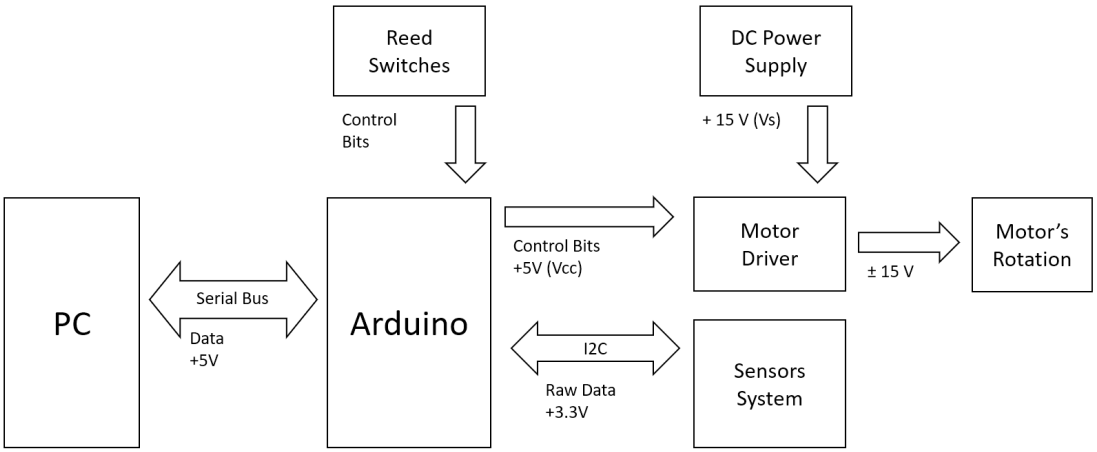


Figure 3.11: Block diagram of sensors and motor system

The proposed inspection system was programmed in a Finite State Machine (FSM) manner, which means that the system runs continuously in a loop changing from one state to another following certain conditions. Three states were defined for the operation of this setup as listed below:

- **State 0:** The motor is OFF until the Arduino device receives a predefined command through the serial port. When this condition is met, the current state changes to 1.

- **State 1:** The motor is ON, moving forward. The sensors are activated and readings are sent to the serial port. The state of the reed switch located at the end of the scan surface is checked. When the reed switch closes, the motor stops and changes its rotation to move backward to the initial position, and the current state changes to 2.

- **State 2:** The motor is ON, moving backwards. The sensors are deactivated. The state of the reed switch located at the start of the scan surface is checked. When the reed switch closes, the motor stops and the current state changes to 0. The cycle starts again.

The cycle described above is run repeatedly until the entirety of a predefined surface is scanned by the proposed monitoring system. The following section describes how data is collected as well as how the surface to be scanned is defined.

3.2.1 MFL Data Collection

As mentioned before, a pair of GMR sensors is used here in order to measure the MFL signals. These sensors detect the three components of the magnetic field. As the monitoring system moves along the pipe performing a line scan, the detected raw signals are sent to the Arduino where they are converted to numerical values given in Gauss (magnetic flux density unit). These measured signals are arranged in six columns, where each set of three corresponds to the Bx, By and Bz components for each of the two sensors. Finally, these signals are sent from the Arduino to Matlab via the serial port. Matlab receives the signals and stores them in a matrix. This principle is applied to collect all the MFL signals detected by the sensors coming from the predefined scan surface. This scan surface is a rectangular section projected on the surface of the pipe with a length of 20.5 cm and a width of 10.2 cm. Furthermore, the short side of this rectangular section is subdivided into 36 smaller sections from which a line scan will be conducted, with every line scan having around 3500 samples. The number of samples depends on the sensing speed (which depends on the supply voltage of the motor) and the delay between readings performed by the sensors. Once the scan is completed, the cycle described in the previous section finishes and the collected readings are stored in matrices ready for further processing, which is described in the following chapter.

3.2.2 Summary

In this chapter the design and implementation of an experimental setup was described. This test rig consisted of a set of 3D GMR sensors mounted over the top of a steel pipe and below a pair of magnets joined by a steel bar. The sensors and magnets traversed along the steel pipe with the help of a threaded rod coupled to a motor whose rotation direction was constrained by two magnetic switches that closed in the presence of a magnetic field. The whole sensing system and motor were controlled by an Arduino UNO board, which was connected to a PC where all the generated data was logged into a matrix within Matlab, ready for further analysis.

Chapter 4

Data Processing and Analysis

Based on the common behaviour of MFL signals as presented by [11, 12], an image-processing approach is followed here. The basic idea is to use all the collected signals and arrange every measurement in parallel in a 3D space, thus creating a surface in which the peaks and valleys of the signals would represent the contours of any detected defect. A visual example of this is shown in Fig. 4.1. By examining these surfaces from the top view, a 2D image is obtained, thus image-processing techniques can be implemented. This chapter describes how the acquired MFL data is processed before it can be used for defect detection.

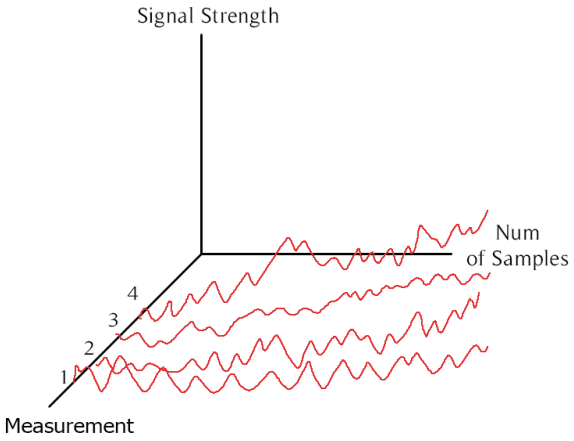


Figure 4.1: Signals parallel arrangement

4.1 Raw Measurements and Signal Filtering

Every project that uses sensors to detect and measure physical parameters is subject to parasite signals, and the MFL method is not exempt from this. Examples of the raw measurements made in the experimental part of this study are shown in Fig. 4.2 and Fig. 4.3. These signals correspond to a single line scan and show that there is a parasite low frequency attached to the signals. This can be seen in the plots of both the Bx and By components. Moreover, the Bz component does not show any variation along the entire reading, which indicates that this component is saturated on both sensors by the magnetic fields. This behaviour is consistent on all of the 36 line scans that were conducted, and for that reason, this component is going to be ignored for the rest of the data processing since its contribution to the overall MFL signals is null.

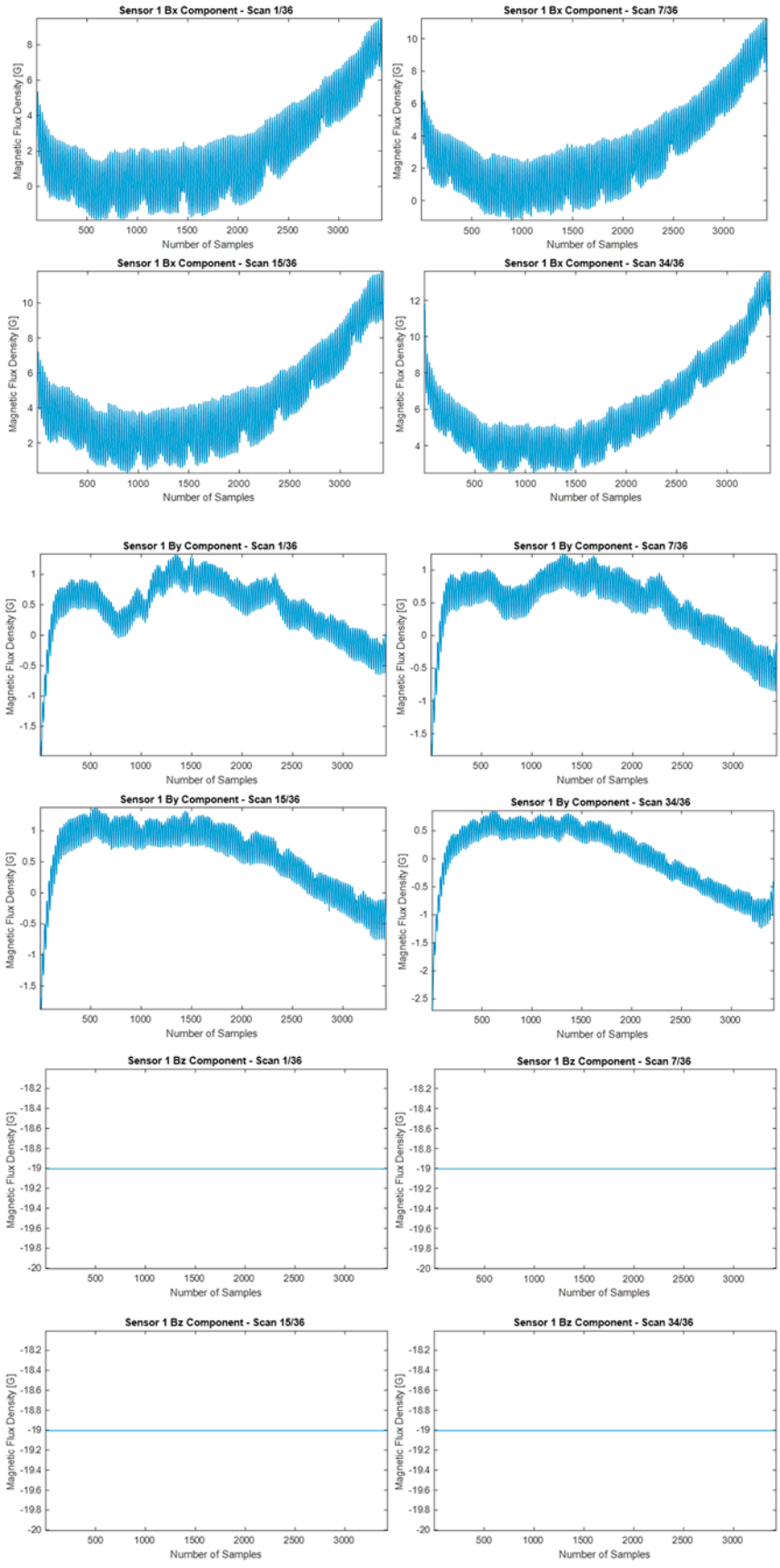


Figure 4.2: Raw measurements from Sensor 1

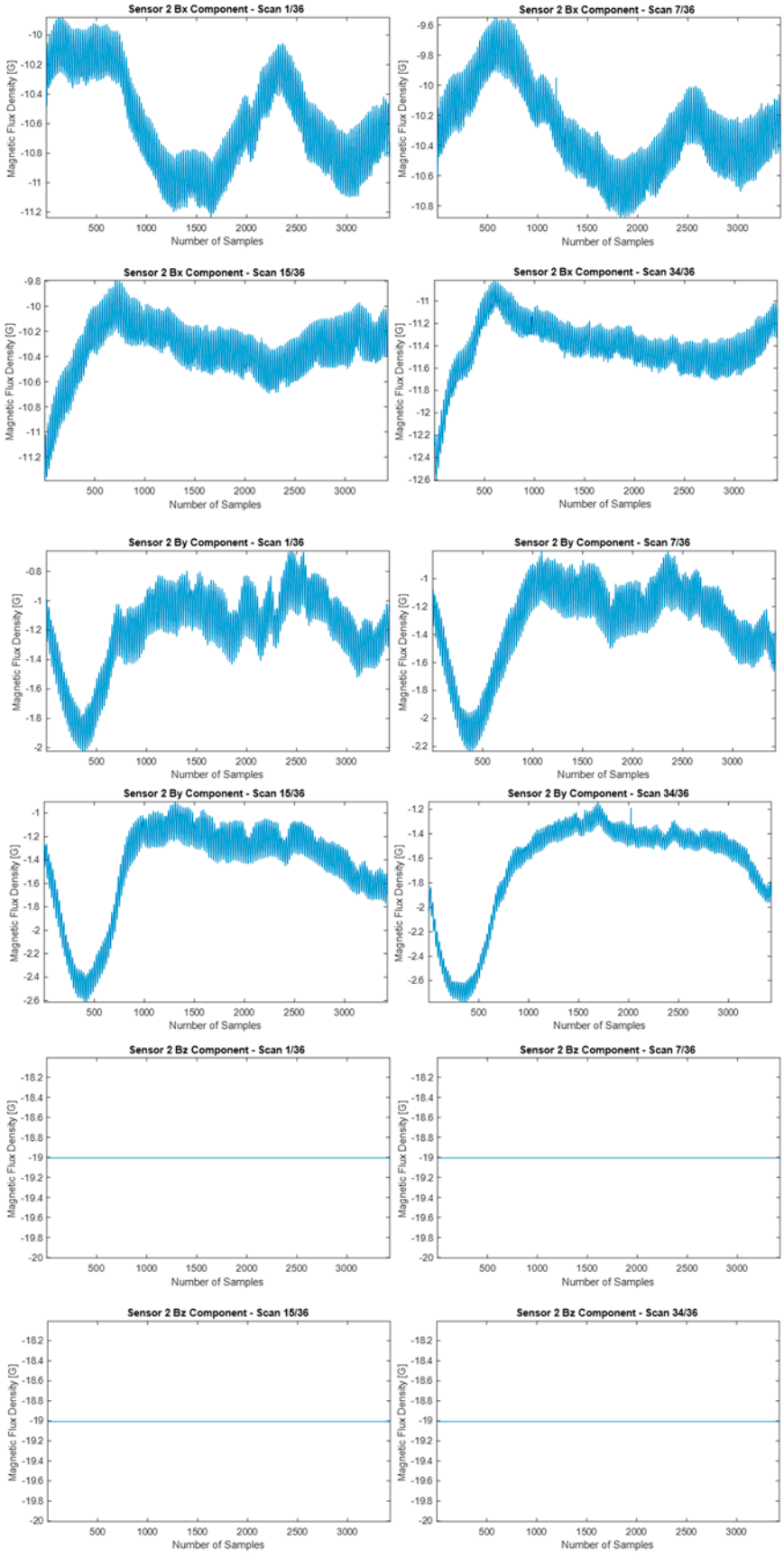


Figure 4.3: Raw measurements from Sensor 2

In order to find and eliminate the parasite low frequency, the frequency spectrum of the signals was obtained with the Fourier Transform. This technique allows the identification of all the frequencies that compose any given signal. Fig. 4.4 shows the frequency spectrum of one of the several signals. It can be seen here that there is a peak of 0.6 dB at the frequency of 0.09 π radians/sample (normalised frequency), which represents the parasite low frequency described before. By attenuating this peak, the parasite low frequency will be removed from the original signal. This is achieved by applying a Band Stop filter to the signals, which is a type of filter that combines both a low-pass and a high-pass filter in order attenuate the frequencies within the low and high cut-off frequencies. Examples of the cleaned signals are shown in Fig. 4.5 and Fig. 4.6. These graphs show that the low frequency was removed from the signals and that they kept their shape. Moreover, the remaining higher frequency noise shown in these images is due to harmonics of the previously identified low frequency noise, also shown in Fig. 4.4. However, these harmonics were ignored since they did not represent any major problem for the rest of the signal processing.

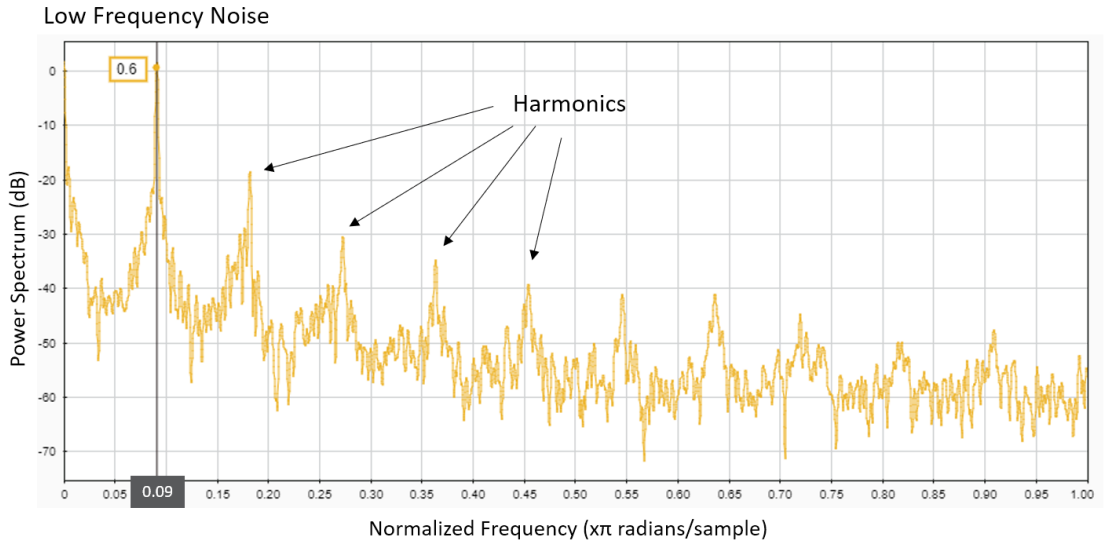


Figure 4.4: Frequency Spectrum

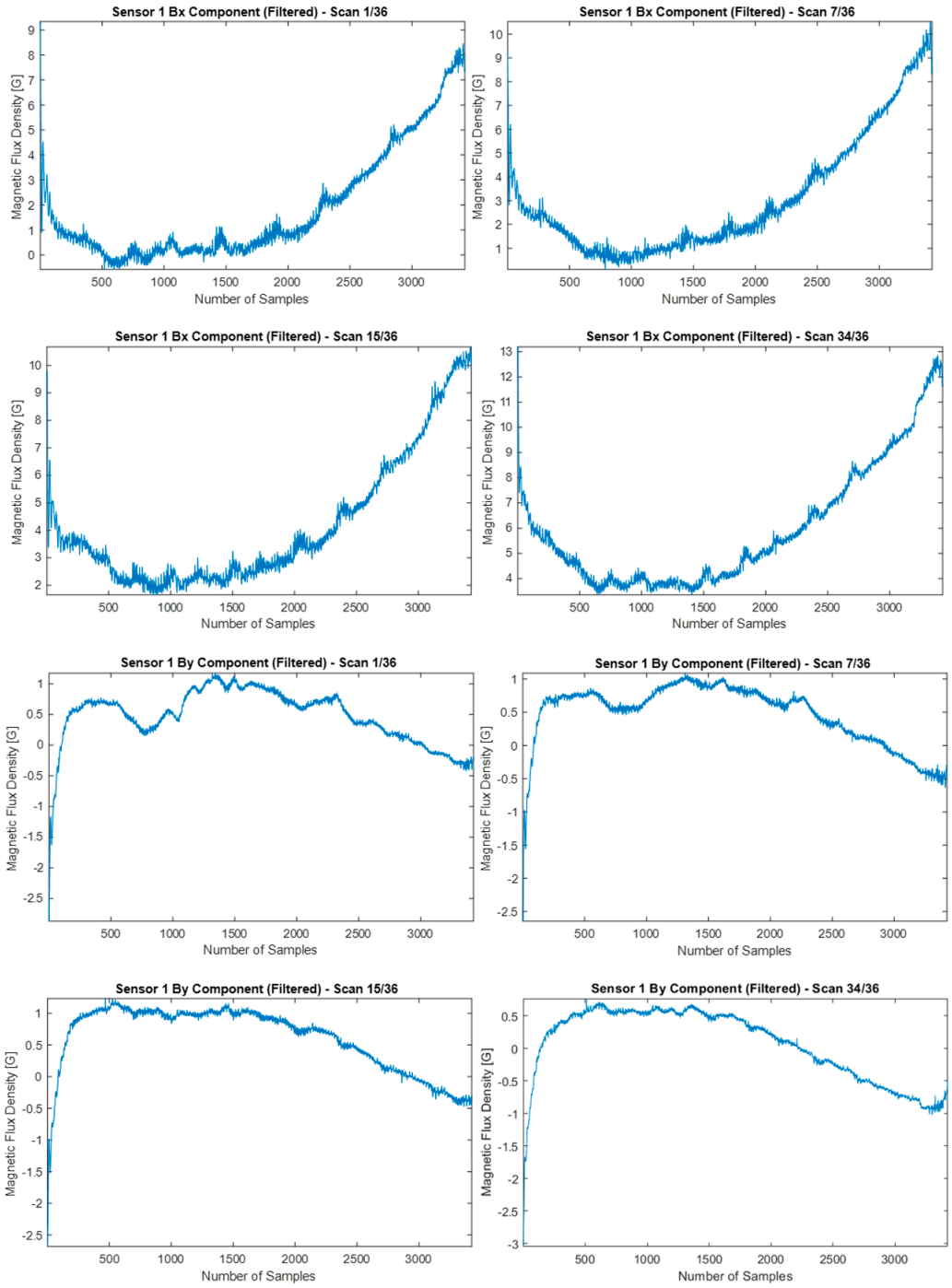


Figure 4.5: Filtered signals from Sensor 1

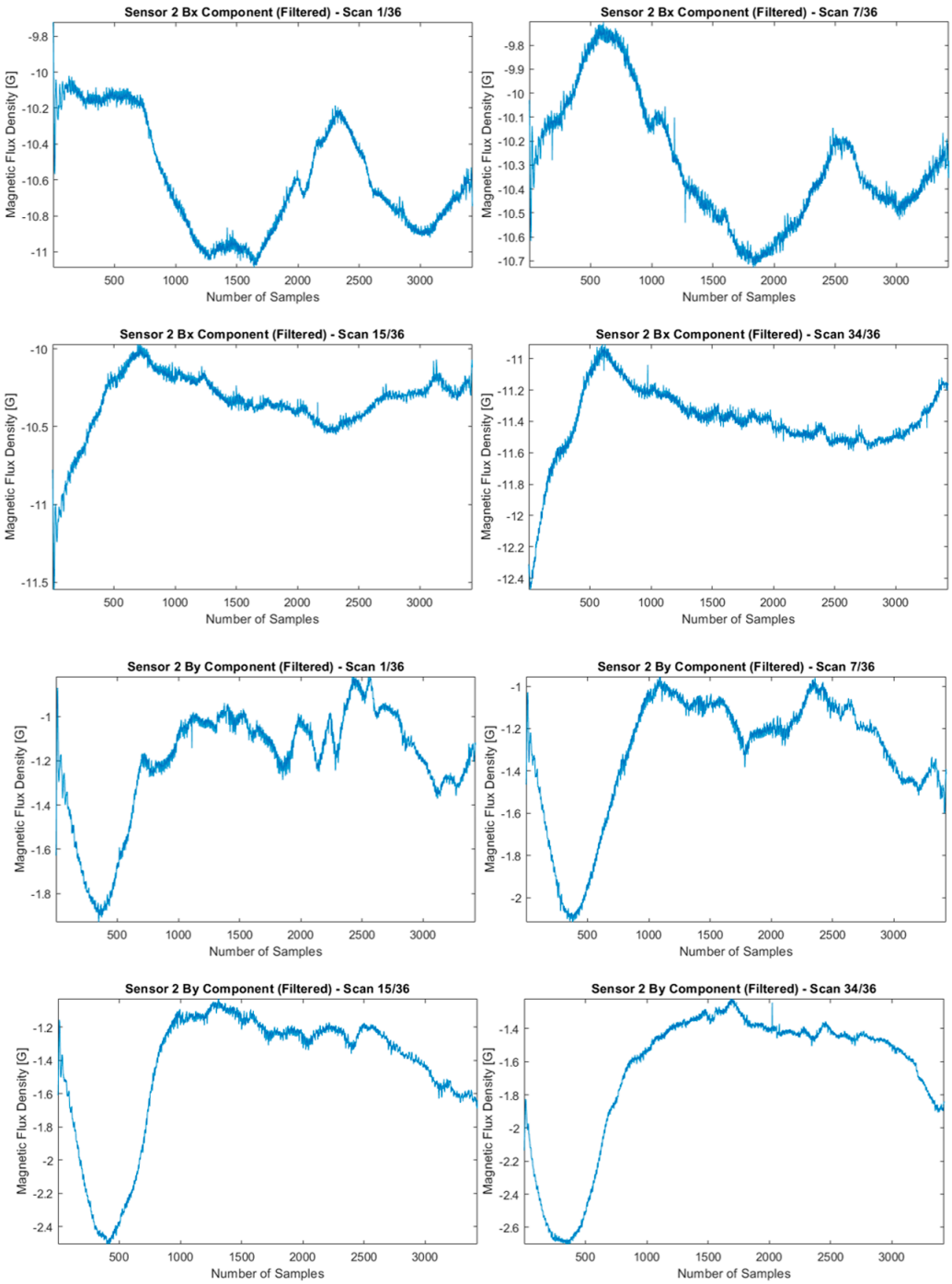


Figure 4.6: Filtered signals from Sensor 2

As mentioned before, one of the main objectives of the present thesis is to use MFL signals to reconstruct 2D images using the three components of the magnetic fields. However, having that the measurements described in this chapter showed no readings for the Bz component, simulations were conducted using COMSOL Multiphysics to produce the necessary data in order to determine whether the proposed method is adequate for defect detection.

Chapter 5

Simulated MFL Inspection

A simulated model is presented in this chapter, where a steel pipe is magnetised by a pair of permanent magnets joined with a steel yoke and MFL signals are collected from a predefined area below the magnetiser, similar to what was done in the experiments. The collected data is then used to recreate 2D images from which defects can be detected. This method will then be applied to the experimental data previously gathered in order to validate the results. The simulations were conducted using COMSOL Multiphysics and the process is detailed in the following sections. The proposed algorithm to create the 2D images was written in Matlab and is also described in this chapter.

5.1 COMSOL Multiphysics

COMSOL Multiphysics is a simulation platform that provides engineers and scientists with total control over all the steps in a modelling workflow, this is from the design of geometries in a CAD environment, materials assignation, and the definition of the governing physics that describe the specific phenomena of interest. It also provides the user with tools for the post-processing of results for a better and easier visualisation. Moreover, previously designed geometries can be imported into COMSOL easily since it is compatible with all standard CAD files [56]. Moreover, model solving is carried out in COMSOL by a combination of methods such as the finite element method (FEM), adaptive meshing and a variety of numerical solvers [57].

COMSOL is capable of simulating multiple real-life scenarios where various partial differential equations (PDEs) have to be solved. These scenarios include acoustics, chemical reactions, electrochemistry, electromagnetics, fluid dynamics, heat transfer, semiconductor devices, and many more [57]. These different areas are integrated in modules within the software and can be combined to create multiphysics systems. Additionally, COMSOL can solve the simulated systems under different studies such as stationary state and time-dependent studies, linear and nonlinear studies, as well as eigenfrequency, modal and frequency response studies [57]. Another useful tool provided by this software is the ability to perform Parametric Sweep type studies, which allow the simulated system to be solved with a combination of parameters that might be of interest, such as the size of a given object and the temperature around the system, including the assignation of different materials [56]. There is also an interface called Application Builder where the user can write short scripts of code in the Java programming language with which geometries can also be created and parameterised.

For the purposes of this study, the AC/DC Module is utilised since it provides the tools for simulating electromagnetic fields in 2D and 3D scenarios. The AC/DC Module itself is also divided in multiple sub modules that focus on specific areas of the electromagnetics field of study, such as Electric Currents, Electrical Circuits, Electrostatics, Magnetic Fields, Magnetic Fields (No Currents), and Magnetic and Electric Fields, to name but a few [57]. The following section will describe how the geometries, parameters and physics were set up to create the simulation environment relevant to this study.

5.2 Simulated MFL System Modelling

All the geometries used for the simulation are based on the dimensions of the real components (i.e. magnets, pipe). Furthermore, this reference steel pipe has a rectangular crack which was used to run the initial tests and simulations. Afterwards, the results obtained from this configuration will be compared with the results obtained from the experimentation. After that, different defects will be simulated including

ellipsoidal and circular defects, but emphasising on the irregular-shaped defects in adherence to the objectives of this thesis. The irregular defects will be modelled using a combination of ellipsoidal objects since this type of defects usually takes a coarse, circular shape.

5.2.1 System Definitions and Parameter Initialisation

In order to create a simulated MFL system, three main definitions must be predefined in COMSOL. These include the Space Dimension, the Physics to be solved, and the Study to perform. The space dimension was set to 3D, whereas the selected physics was the Magnetic Fields (No Currents) interface since permanent magnets will be simulated here and this interface solves the Gauss' Law for static magnetic fields. Finally, a stationary type of study was chosen for this system. Fig. 5.1 shows the previously described steps for the definition of the three initial settings discussed in this section, including a brief description of the purposes of the selected physics and type of study as provided by the software itself.

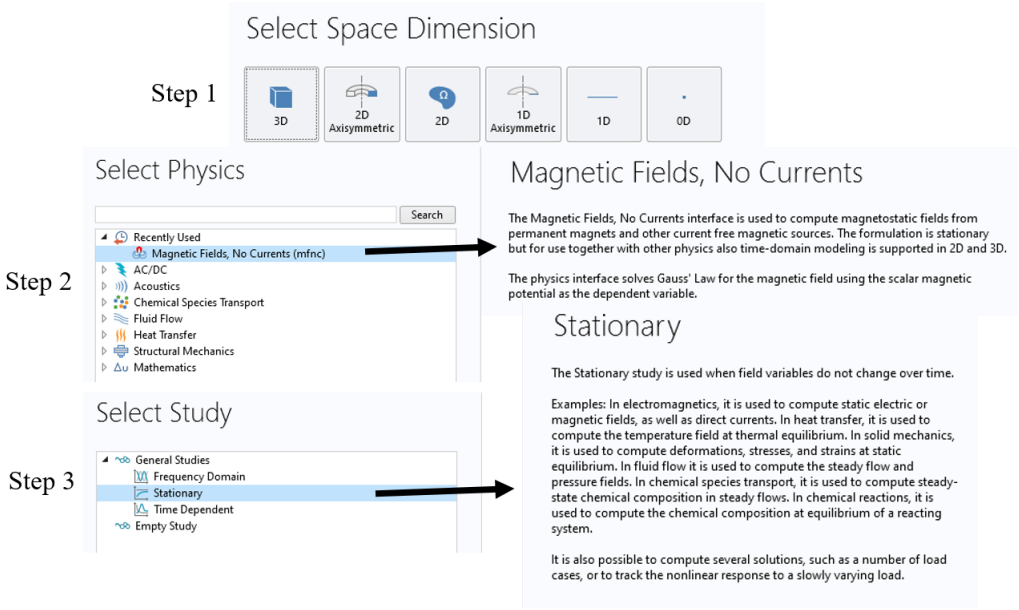


Figure 5.1: Steps for system settings definition

Once the space, physics and type of study have been properly defined, the software's interface changes to show a different window where the model can be designed. COMSOL has a user-friendly interface that facilitates progression of the modelling

that uses a ribbon on top of the window with buttons for each of the steps of the simulation process. As mentioned before, different parameters can be set in order to define the size or location of the geometries in the simulation. Here, several parameters were defined, including the dimensions for the magnets, the brushes and the steel yoke, as well as the diameter, thickness, and length of the pipe. These parameters will be used to build the different components of the simulated MFL method described in this chapter.

5.2.2 Steel Pipe

In order to model a pipe in COMSOL is by creating two cylindrical objects oriented towards the positive X-axis and subtracting the smaller cylinder to create a hollow tube. The steel pipe simulated for this study has a radius of 50.8 mm, a thickness of 2.5 mm and a length of 1 m, based on the dimensions of the real pipe that was used in the experimentation. The simulated defects are much smaller than the pipe and are modelled in a predefined area smaller than the pipe that corresponds to the area below the magnetiser and sensor system. All the information generated in the simulation will be extracted from this scan surface. The simulated pipe can be seen in Fig. 5.2

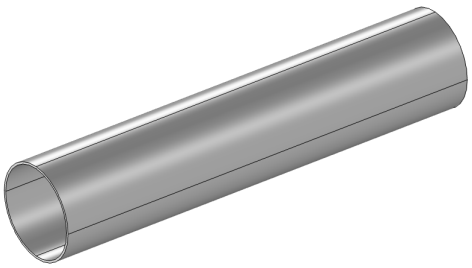


Figure 5.2: Simulated Pipe

5.2.3 Magnetiser System

As mentioned before, the magnetiser used here is a modification of the usual configuration used in the MFL method, due to the characteristics of the pipe that will be used in the experiments. This magnetiser consists of two permanent magnets, joined by a rectangular steel yoke, and attached to the magnets are two other steel

pieces that bring the magnetic field lines closer to the specimen. These brushes were designed in a way so that they have a circular section with the same radius as the pipe to ensure that both pieces have the same liftoff from the pipe. Furthermore, due to the size of the pipe, the magnetiser is located outside and on top of the pipe. The dimensions with which these geometries were modelled are shown in Table 5.1, whereas Fig. 5.3 shows the magnetiser on top of the pipe.

Object	Width	Length	Height
Magnets	20 mm	40 mm	5 mm
Brushes	20 mm	40 mm	35 mm
Yoke	40 mm	90 mm	12 mm

Table 5.1: Magnetiser dimensions

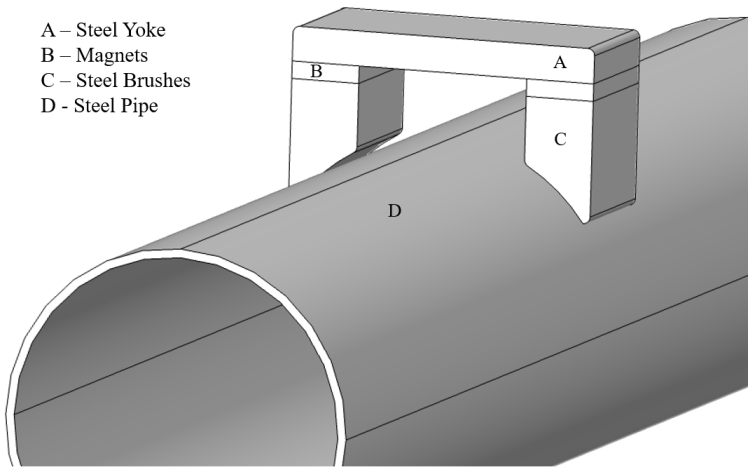


Figure 5.3: Magnetiser on top of pipe

The area below this magnetiser is defined as the aforementioned scan surface. The idea behind this partition is to simulate the area covered by either an array of several sensors or multiple line scans made by one sensor until the whole area is scanned. The simulated defects are modelled in this section of the pipe and the simulated MFL signals are collected from this surface. Additionally, it should be noted that the resolution of this defined surface is much higher than the resolution that could be achieved in a multi sensors or multi line scans scenario since the simulation was solved using an extra fine mesh for better results. The following

section describes how defects were modelled to adhere to the conditions mentioned before.

5.2.4 Simulated Defects

As previously stated, a rectangular crack similar to that on the reference steel pipe was modelled. This crack is 15 mm long, 2 mm wide, and its depth is 2.5 mm, equal to the thickness of the pipe. However, since one of the main objectives of this thesis is to study irregular defects, the emphasis is on them. In order to create these irregular shapes, the Application Builder interface in COMSOL was used. This interface allows the user to write short scripts of code in the Java programming language, which are internally called *Sequences*. Here, several operations can be performed with lines of code ranging from simple tasks such as defining the physics of the simulated system to more complex tasks such as automatically running a simulation with changing parameters and exporting the generated data.

In order to create the desired irregular defects, a sequence was written to randomly create 10 ellipsoidal objects within the range of 1 mm to 10 mm for the length and width, and the depth was set within a range of 10 % to 80 % of the pipe thickness. The position of the multiple ellipsoidal objects was delimited to the boundaries of the predefined area below the magnets. After these shapes were completed, adjustments were made to their orientation and position in the Z-axis so that they were in contact with the pipe's surface. Moreover, although 10 individual ellipsoidal shapes were built, the random generation of these multiple shapes made several new shapes, comprised of the overlapping of various ellipsoidal objects. This combination of objects created new asymmetric and irregular shapes with different widths, lengths and depths. These new shapes will be referred to as irregular defects for the rest of this thesis. This process was repeated if no irregular shapes were constructed. Finally, to simplify the simulation and reduce computation time, some of the ellipsoidal objects were removed from the model. The predefined smaller pipe section with the rectangular crack and the ellipsoidal objects are shown in Fig. 5.4. The magnetiser was removed from the image for better visualisation. It can be seen here that the rectangular crack is much larger than the ellipsoidal and irregular

defects. The difference in size and shape of these defects will be used to test the efficiency and robustness of the proposed methodology.

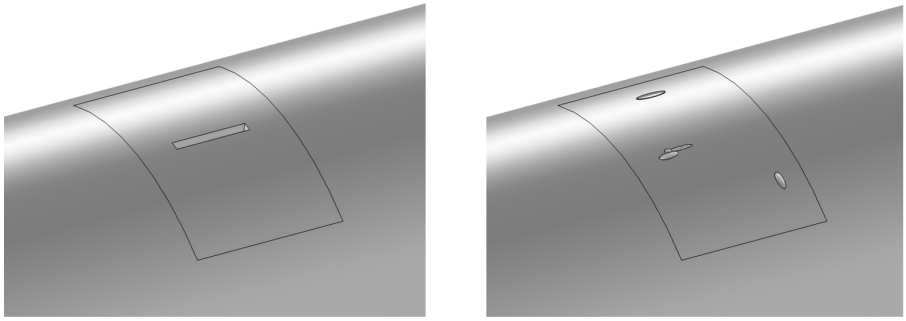


Figure 5.4: Simulated Defects. a) Rectangular Crack. b) Ellipsoidal Objects

5.2.5 Data Visualisation and Exporting

The simulation results can be visualised in COMSOL using a variety of plotting options, which include 3D, 2D and 1D plot groups. Using a Surface Plot within the 3D plot group, the three components of the MFL signals can be visualised. This is shown in Fig. 5.5, which includes surface plots of the scan surface described before. Here, the simulated rectangular crack and its influence on the magnetic fields can be seen. The colour code represents the intensity of the magnetic flux density ($\vec{B}$) measured in Tesla. The plot for the Bx component in Fig. 5.6(a) shows the expected peaks and valleys of the MFL signal on the corners of the simulated rectangular defect. Furthermore, Fig. 5.6(b) shows similar results as those of the literature, having a pair of peaks on both of the long ends of the defect and a valley in the centre of the defect. Finally, Fig. 5.6(c) shows that the Bz component has little contribution to the overall MFL signals. However, the expected behaviour is present here, having two opposite peaks on both of the longer sides of the rectangular crack. Furthermore, the surface plots of the multiple ellipsoidal defects are shown in Fig. 5.6. These plots show a different response of the three components, and this indicates that the shape of the simulated defects affect the behaviour of the signals. Furthermore, these plots do not show peaks and valleys that are easy to identify such as those of the rectangular crack. However, rather than using these

surface plots to conduct defect detection, only their numerical representation will be exported into a single text file organised in the form of a matrix. The processing made to the data obtained from the simulation is described in the following sections.

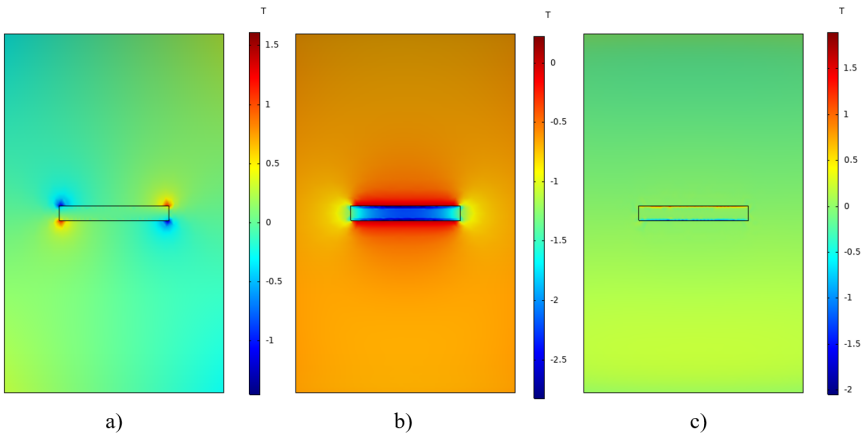


Figure 5.5: Surface Plots of Crack Defect MFL Components. a) Bx b) By c) Bz

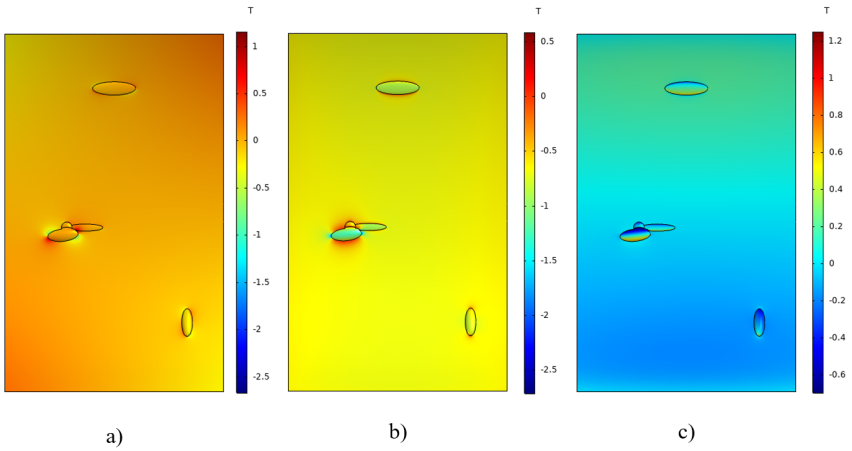


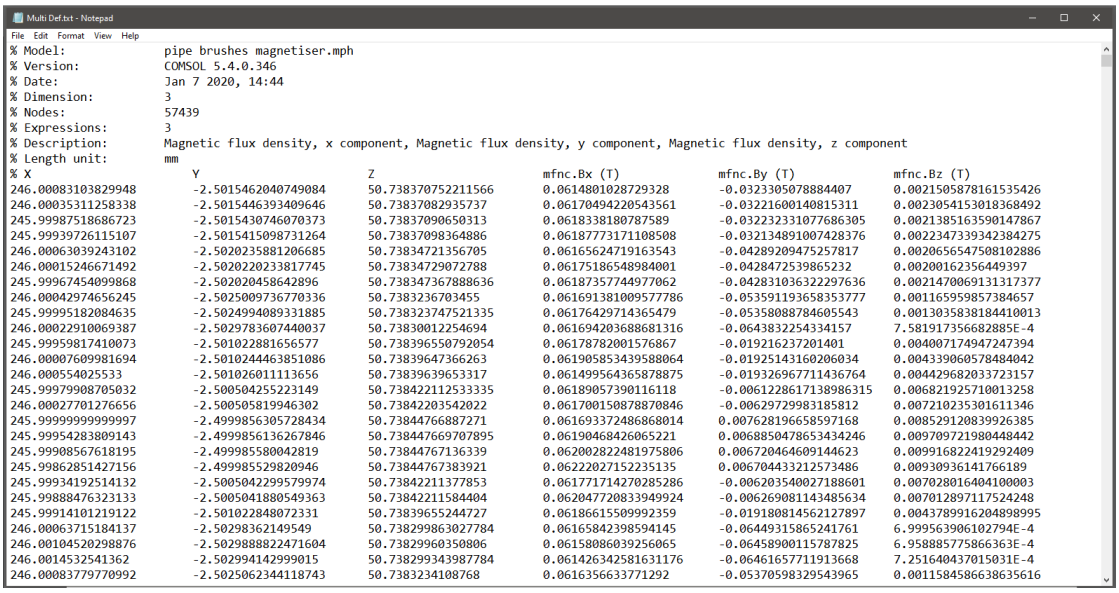
Figure 5.6: Surface Plots of Ellipsoidal Defects MFL Components. a) Bx b) By c) Bz

5.3 Simulated MFL Data Processing with Matlab

The main contribution of this thesis is detailed in this section, which describes how the MFL data is processed in order to reconstruct 2D images and the use of image processing techniques to detect and characterise defects. The algorithms developed to carry out image reconstruction and defect detection are described here.

5.3.1 Importing Data and Image Reconstruction

As previously mentioned, the data extracted from the simulation in COMSOL is organised in a matrix containing six columns with several rows. The first three columns correspond to the coordinates from which MFL data is taken, whereas the last three columns contain each of the three MFL components. Consequently, each row gives the spatial coordinates as defined in the simulation from which one measurement was made for each of the three components of the MFL signals. An example of this can be seen in Fig. 5.7. The matrix has as many rows as the simulation software needed to solve the system depending on the resolution of the mesh used in the FEM model.


Multi Def.txt - Notepad

File Edit Format View Help

```
% Model:              pipe brushes magnetiser.mph
% Version:            COMSOL 5.4.0.346
% Date:               Jan 7 2020, 14:44
% Dimension:          3
% Nodes:              57439
% Expressions:        3
% Description:        Magnetic flux density, x component, Magnetic flux density, y component, Magnetic flux density, z component
% Length unit:        mm
% X                       Y                       Z                       mfnc.Bx (T)             mfnc.By (T)             mfnc.Bz (T)
246.00083103829948      -2.5015462040749084     50.738370752211566      0.0614801028729328      -0.0323305078884407     0.0021505878161535426
246.00035311258338      -2.5015446393409646     50.73837082935737       0.06170494220543561     -0.03221600140815311    0.0023054153018368492
245.99987518686723      -2.5015430746070373     50.73837090650313       0.0618338180787589      -0.032232331077686305   0.0021385163590147867
245.99939726115107      -2.5015415098731264     50.73837098364886       0.06187773171108508     -0.032134891007428376   0.0022347339342384275
246.00063039243102      -2.5020235881206685     50.73834721356705       0.06165624719163543     -0.04289209475257817    0.0020656547508102886
246.00015246671492      -2.5020220233817745     50.7383472907278        0.06175186548984001     -0.0428472539865232     0.002001623564493 97
245.99967454099868      -2.502020458642896      50.738347367888636      0.06187357744977062     -0.042831036322297636   0.0021470069131317377
246.00042974656245      -2.5025009736770336     50.7383236703455        0.061691381009577786    -0.053591193658353777   0.001165959857384657
245.99995182084635      -2.5024994089331885     50.738323747521335      0.06176429714365479     -0.05358088784605543    0.0013035838184410013
246.00022910069387      -2.5029783607440037     50.73830012254694       0.061694203688681316    -0.0643832254334157     7.5819173566828885E-4
245.999598174100 73     -2.501022881656577      50.738396550792054      0.06178782001576867     -0.019216237201401      0.004007174947247394
246.00007609981694      -2.5010244463851086     50.73839647366263       0.061905853439588064    -0.01925143160206034    0.0043390605784840 42
246.000554025533        -2.501026011113656      50.73839639653317       0.061499564365878875    -0.019326967711436764   0.0044296820337231 57
245.99979908705032      -2.500504255223149      50.738422112533335      0.06189057390116118     -0.006122861713898631 5 0.006821925710013258
246.00027701276656      -2.500505819946302      50.73842203542022       0.061700150878870846    -0.00629729983185812    0.0072102353016113 46
245.99999999999997      -2.4999856305728434     50.73844766887271       0.061693372486868014    0.007628196658597168    0.0085291208399263 85
245.99954283809143      -2.4999856136267846     50.738447669707895      0.06190468426065221     0.006885047865343424 6  0.0097097219804484 42
245.99908567618195      -2.499985580042819      50.73844767136339       0.062002822481975806    0.006720464609144623    0.0099168224192924 09
245.99862851427156      -2.499985529820946      50.73844767383921       0.06222027152235135     0.006704433212573486    0.0093093614176618 9
245.99934192514132      -2.5005042299579974     50.73842211377853       0.061771714270285286    -0.006203540027188601   0.0070280164041000 03
245.99888476323133      -2.5005041880549363     50.73842211584404       0.062047720833949924    -0.006269081143485634   0.0070128971175242 48
245.99914101219122      -2.501022848072331      50.73839655244727       0.06186615509992359     -0.019180814562127897   0.0043789916204898995
246.00063715184137      -2.50298362149549       50.738299863027784      0.06165842398594145     -0.06449315865241761    6.9995639061027946E-4
246.00104520298876      -2.5029888822471604     50.73829960350806       0.06158086039256065     -0.06458900115787825    6.9588857758663 63E-4
246.0014532541362       -2.502994142999015      50.738299343987784      0.061426342581631176    -0.06461657711913668    7.2516404370150 31E-4
246.00083779770992      -2.5025062344118743     50.7383234108768        0.0616356633771292      -0.05370598329543965    0.0011584586638635616
```

Figure 5.7: Exported MFL data

These numerical values are then imported into Matlab and are rearranged in a 3000 × 3000 pixels grid in order to reconstruct a 2D image. Each pixel of this grid corresponds to each pair of values in the X and Y columns. The values in the Bx, By and Bz columns are used to create an intensity map by associating each of these numbers to their corresponding X-Y pair. Since the images are reconstructed using the same numerical values internally used by COMSOL, the output images created by the presented approach are similar to the surface plots created in COMSOL, as shown in Fig. 5.8 and Fig. 5.9 with the MFL response of the crack defect and the multiple ellipsoidal defects, respectively. This indicates that the image recon-

struction process was accurately achieved and that there is no loss of information. Moreover, the resolution of these reconstructed images can be increased if the size of the grid is increased, although this will also have an impact on the computing time.

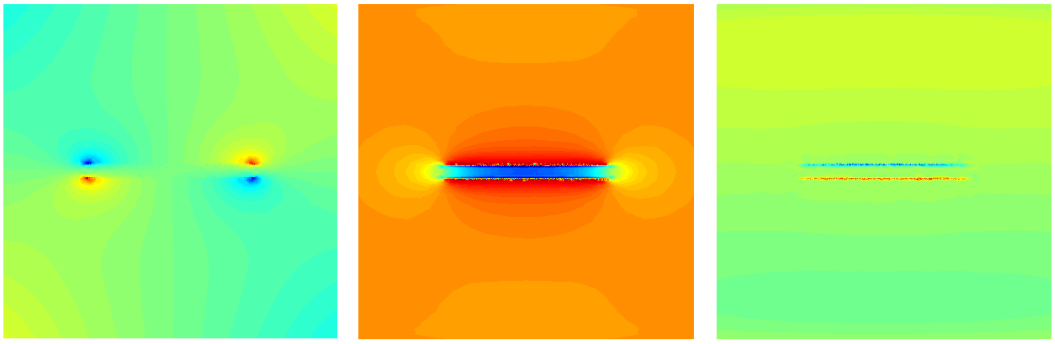


Figure 5.8: Reconstructed MFL Components of Crack Defect. a) Bx. b) By. c) Bz.

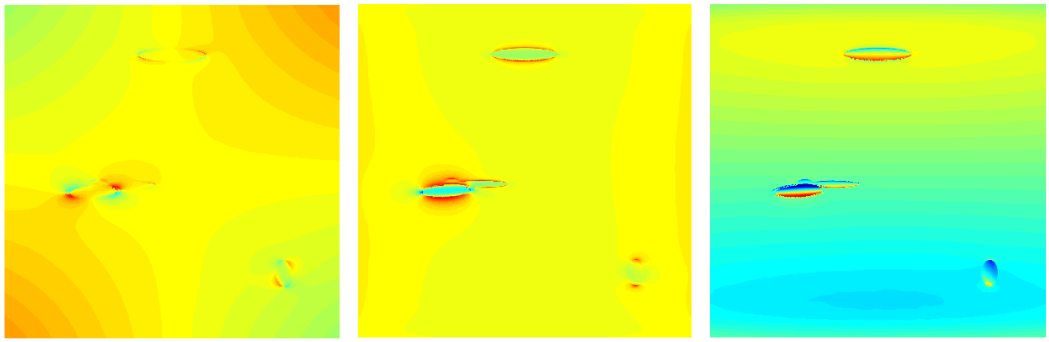


Figure 5.9: Reconstructed MFL Components of Ellipsoidal Defects. a) Bx. b) By. c) Bz.

5.3.2 Contrast Enhancement

Once the MFL images are reconstructed, they are converted into grayscale images before contrast enhancement is performed. This is applied in order to ensure true whites and blacks in the image, thus facilitating edge detection. This process is achieved by obtaining the histogram of the input image to determine the distribution of the pixels that compose the image, and then modifying said distribution so that it is centred in the histogram, thus increasing the contrast of the original image. The upper and lower limits of the input image are obtained with the Central Limit

Theorem (CLT), which states that for any given dataset with random population and distribution, if sufficient samples are taken from that population and the means of those smaller samples are calculated, their distribution will become approximately normally distributed [58, 59]. The upper and lower thresholds are calculated as follows

$$Min = \mu_i - (n \cdot \sigma) \tag{5.3.1}$$

$$Max = \mu_i + (n \cdot \sigma) \tag{5.3.2}$$

where μ_i is the mean of the grayscaled image, and $(n \cdot \sigma)$ is the number of standard deviations in which the histogram will be stretched. With this approach, the brightest and darkest parts of the grayscale image (Min and Max, respectively) are obtained and used as the input values for a Matlab function for contrast enhancement. From this point, only one of the reconstructed images will be used to illustrate the methodology presented in this thesis. A thorough discussion of the results obtained with it is included in Chapter 6. The images in Fig. 5.10 show the histograms of the input image and the contrast-enhanced image, corresponding to the By component of the rectangular crack simulation. It can be seen here that the distribution of pixels for the original image is biased to the right-hand side of the histogram, which makes the image brighter. Once the histogram is modified as described above, the histogram of the output image is centred. The X-axis of the histogram plots correspond to the grayscale colormap, which ranges from 0 to 255. The Y-axis is the count of pixels in a specific grayscale value.

5.3.3 Edge Detection and Morphological Operations

Once contrast enhancement is completed, Gaussian blur is applied to the image in order to make it smoother and eliminate possible noise. Afterwards, the Canny algorithm is implemented in order to find edges in the image. The Canny edge detector calculates the derivative of a Gaussian function searching for the intensity of the gradients of any given image [60]. The Canny edge detector output is a binarised image containing a black background and the detected contours in white. An example of

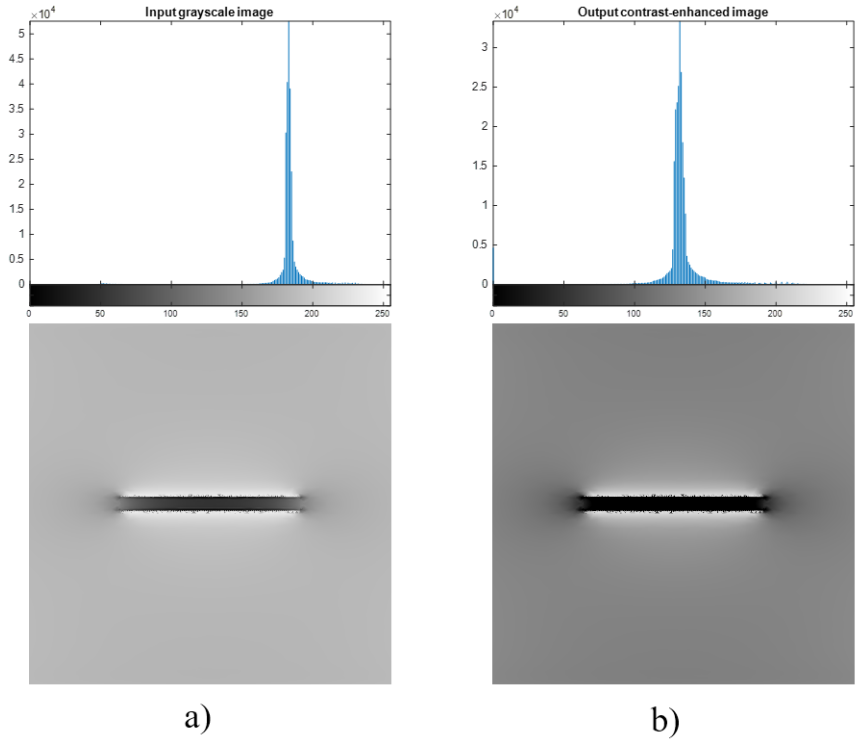


Figure 5.10: Contrast Enhancement. a) Input image. b) Output image.

this is shown in Fig. 5.11 with the colours inverted for better visualisation.

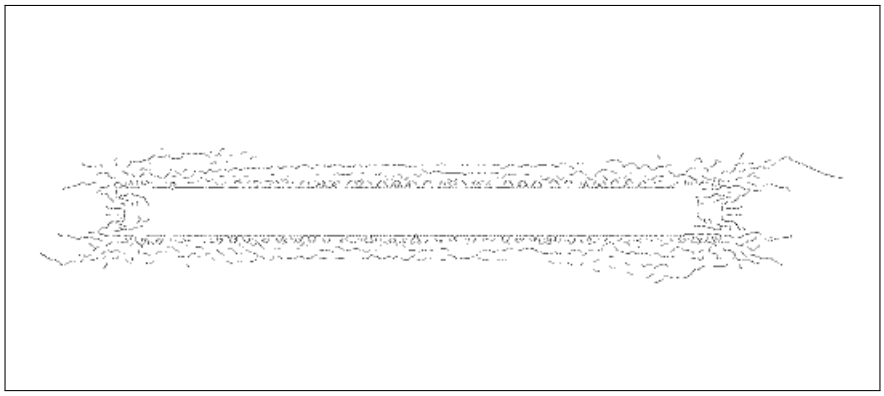


Figure 5.11: Detected Contours (zoomed, inverted colours)

After the contours have been detected, morphological operations are implemented in order to remove any irrelevant information in the image and emphasise the detected contours. Morphological operations relate to the shape of the features within an image, and are mostly implemented in binary images. These operations need a "structural element", which is probed and positioned in all possible locations in the image to compare it with the neighbouring pixels [61]. In this study, dilation,

erosion and opening operations are performed on the input image using disk and line structural elements. The specific parameters for these operations are dependent on the MFL component to which they are being applied. The flowchart in Fig. 5.12 briefly describes how the three components are processed.

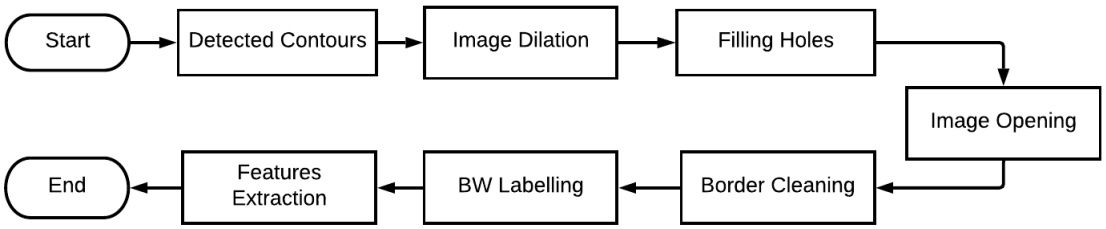


Figure 5.12: Basic processing of MFL images

Dilation is applied in order to thicken the detected contours so that the neighbouring pixels connect and create a closed shape. Once this is achieved, the filling operation is performed to fill the inner area of the closed shape (hole) and create a "solid shape". The opening operation is a combination of two other operations, an erosion followed by a dilation (erosion is the opposite of dilation). This operation is applied in order to eliminate those smaller shapes that may surround the bigger shapes. Here, more erosion operations could be applied to the resulting image to eliminate more noise, but this could also have a negative impact on the relevant shapes. The output image of each of the dilation, filling and opening operations included in the previous flowchart are shown in Fig. 5.13

After these morphological operations are completed, the border of the overall image is cleared so that any residual noise around it is deleted. Sometimes, small shapes are still present in the image and, in order to remove them, a size filter is used so that only those shapes with the largest area are included in the image. Here, a limit of six shapes per image was set. This process is conducted for each of the MFL images (i.e. MFL components) and the output binary images are stored individually before they are merged to create a unique map of the detected shapes. A final erosion operation is performed on the image containing all the detected shapes in order to remove any noise produced by the merging. Fig. 5.14 includes the individual output edges of the three components and the cleaned merged image.

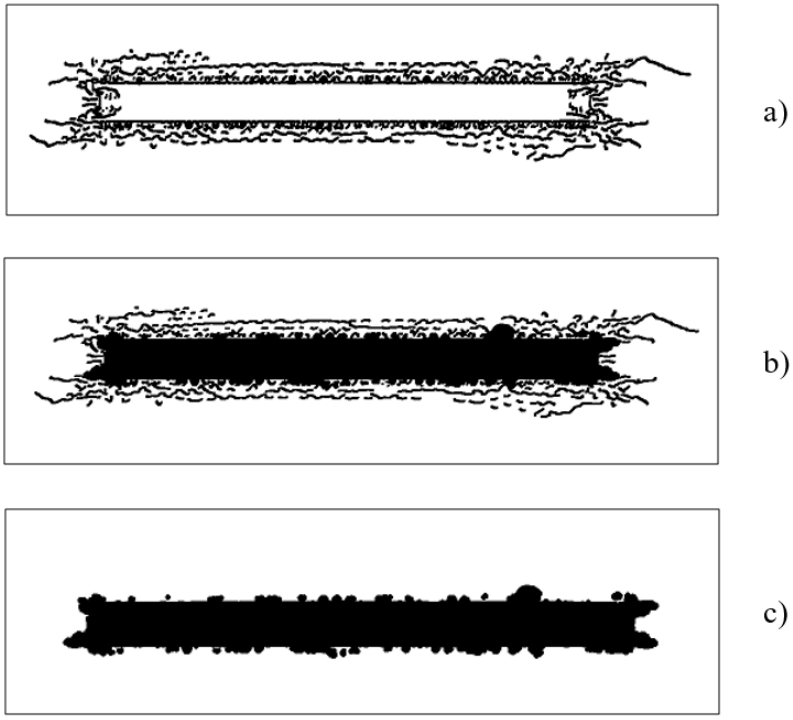


Figure 5.13: Morphological Operations Output (zoomed, inverted colours). a) Dilation. b) Filling. c) Opening

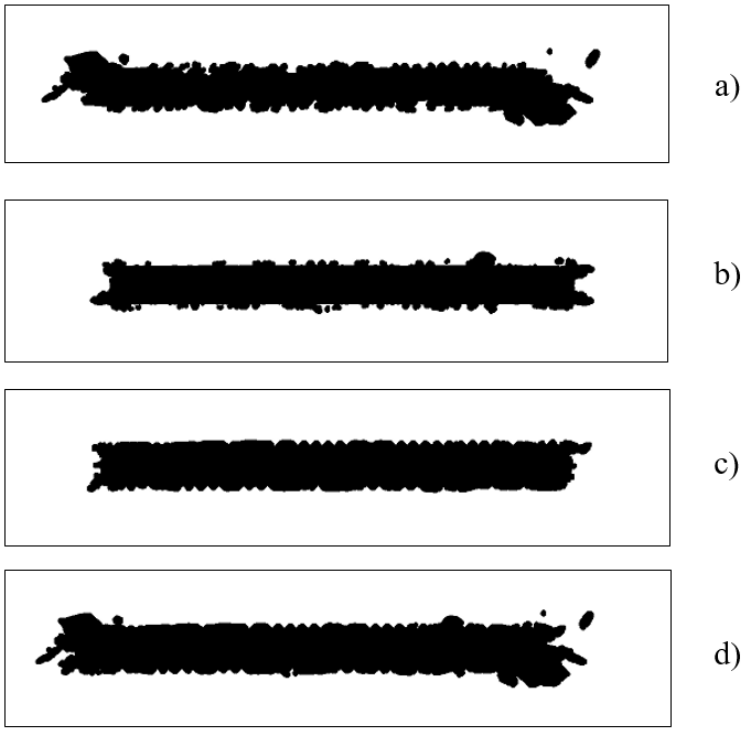


Figure 5.14: Detected Shapes (zoomed, inverted colours). a) Bx. b) By. c) Bz. d) Merged

5.3.4 Image Segmentation with the Watershed Transform

Once the final binary image is obtained, the watershed transform is used to carry out image segmentation in order to produce visual labelling of the detected shapes for the final output of the present method. Furthermore, this algorithm allows the extraction of the geometrical features associated to those blobs such as their centroid, area, orientation, to name but a few. These features will be used to carry out the size estimation of the detected shapes. The watershed transform is an emulation of a real-life watershed, where a ridge separates two areas that can be filled with water. This analogy is used in image segmentation so that the bright parts of a binary image are considered peaks or mountains, whereas the dark areas are valleys or basins, as shown in Fig. 5.15 [62].

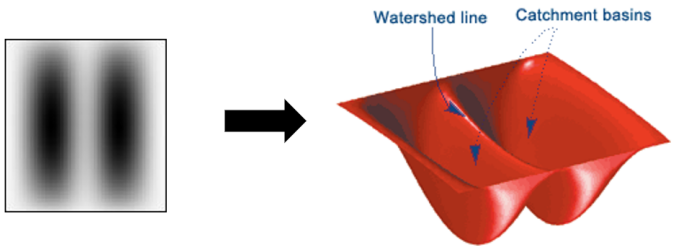


Figure 5.15: The Watershed Transform

In order to implement the watershed transform, the complement of the original grayscale image is obtained. The output is an image where the dark areas become brighter and vice versa. After this, a mask that contains only the perimeter of the blobs is created in order to calculate the local minima in the complement grayscale image, thus generating the ridges and valleys. Once this is completed, the output image is colour labelled and the geometrical features of the shapes are extracted. The output image of the watershed is shown in Fig. 5.16, which includes the colour-labelled image containing the detected shapes and the original MFL grayscaled images with the perimeter of the detected shapes overlayed on them.

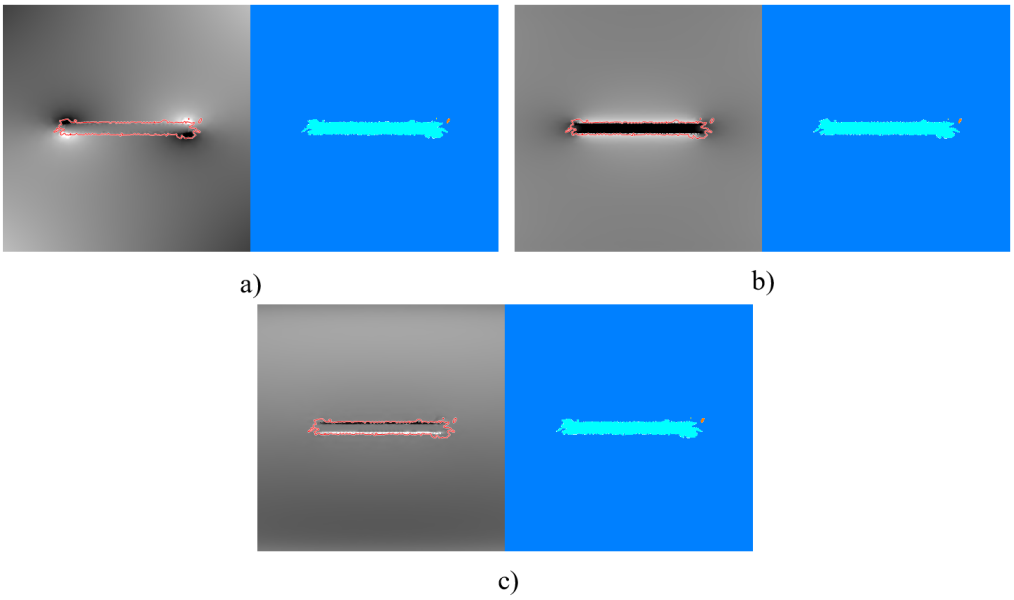


Figure 5.16: Watershed Transform Output. a) Bx. b) By. c)Bz.

5.4 Defect Characterisation

5.4.1 Width and Length Estimation

As previously stated, the geometrical features of the detected shapes in the MFL images will be utilised to carry out size estimation. To achieve this, the main feature to use is the bounding box, which is an imaginary square that encloses any given shape in an image. The image in Fig. 5.17 shows the total number of detected shapes in the reconstructed image with their respective bounding box.

The width and length of such a box are given by a number of pixels so the length in millimetres of such pixels has to be calculated. Since the analysed MFL data come from a predefined area in the simulation with known dimensions, such pixel-millimetre conversion can be calculated using the following expression

$$P_l = \frac{S_w}{I_w} \tag{5.4.3}$$

where P_l is the length of a pixel given in millimetres, S_w is the width of the scanned surface in the simulation given in millimetres, and I_w is the width of the reconstructed image given in pixels. Since pixels are always square, their width and length are always the same. For the simulations presented in this chapter, the width

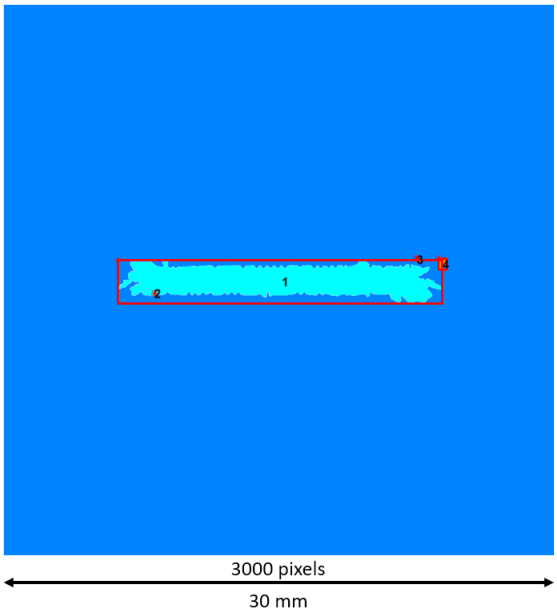


Figure 5.17: Detected Shapes with Bounding Box

of the scanned area is 30 mm and the reconstructed images have 3000 pixels on each side, as seen in Fig. 5.17. Consequently, $P_l = 0.01$ mm as obtained from 5.4.3.

5.4.2 Depth Estimation

As previously stated in this thesis, the investigation presented in [43] found a linear relationship between the average strength of MFL signals and the depth of defects. Their finding allowed them to estimate the depth of defects with low error rate. However, this relationship was studied with defects of regular shape and uniform depth on the bottom of a tank floor. This thesis will attempt to find the same relationship on the curved surface of the studied pipe with a rectangular crack.

To achieve this, a special simulation in which the defect's depth was iterated 32 times in a range between 10% and 80% of the pipe wall thickness ($t = 2.5$ mm) was performed. The images contained in Fig. 5.18 are the surface plots of the Bx component of three different steps of the aforementioned iterative simulation. Furthermore, it can also be seen here how MFL signals are affected by the changing depth of the defect. Fig. 5.18(a) shows the MFL response of the rectangular crack with a depth of 10% of wall thickness (10% t). Similarly, Fig. 5.18(b) shows the MFL response of a defect with 50% t, and Fig. 5.18(c) shows the response of a defect

with 80% t. Moreover, the increase of the intensity of the signals as the depth of the defect increases confirms that there is a connection between the defect's depth and the strength of MFL signals.

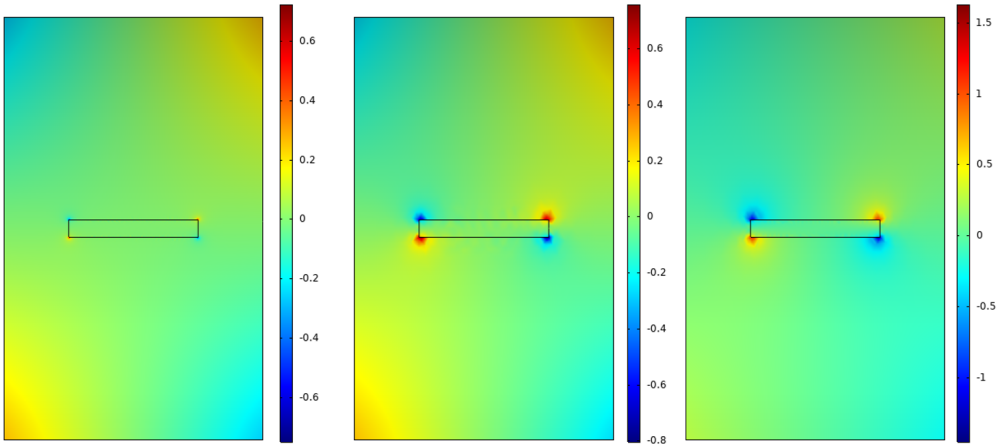


Figure 5.18: Crack defect with different depths. a) 10% t b) 50% t c) 80% t

Therefore, the relationship between these parameters can be described as a mathematical function $D = f(\overline{B})$, where D is the depth of the defect and $\overline{B}$ is the average strength of MFL signals over the 32 iterations. The average strength of the simulated MFL signals was obtained with the following expression

$$\overline{B} = \frac{\sqrt{B_x^2 + B_y^2 + B_z^2}}{M \times N} \tag{5.4.4}$$

where B_x, B_y and B_z are each of the MFL components and $M \times N$ is the size of the scanned area. The results of this approach are discussed later in Chapter 6.

5.4.3 Summary

In this chapter, a simulated model was designed in in order to replicate the experiments previously conducted. The simulations were run in COMSOL Multiphysics where a similar pipe, defects and magnetiser were modelled using the same parameters as in the experimentation setup. Initially, a crack defect was simulated, since the real pipe has an identical crack. This first simulation was made to visualise the three components of the signals as expected from the experiments. After that, irregular shapes were modelled with the help of a short code written in the *Application Builder* within COMSOL that created said irregular shapes using ellipsoidal

objects with random sizes and locations within a range of values. This code was run several times until the ellipsoidal objects overlapped and created irregular-shaped defects. Once they were created, adjustments were made so that the new shapes were adequately located inside the scan surface.

After the simulations were completed, the newly generated data could be visualised using the several plotting options that COMSOL offers. Afterwards, this information was exported to a text file before it was sent to Matlab, where it was stored and arranged in different matrices. Here the data was manipulated in order to reconstruct 2D images for each component of the magnetic signals, obtaining similar images as those shown by COMSOL. Having that the signals contain peaks and valleys in strategic points of the images that correspond to the borders of the defects, an image-processing approach was followed, where the Canny edge detector was implemented to identify those contours that are not easily visible in the original images. To achieve this, the images were converted to grayscale images and then contrast enhancement was conducted in order to emphasise the darker and brighter parts of the images. Once the contours were identified, morphological operations were applied to them in order to create a *blob* (i.e. irregular object in an image) for each detected shape in the image. This principle was applied to every reconstructed image (one per MFL component) and the three resulting images were then merged to create a fourth image in which the three components were contributing to the mapping of the detected shapes. After that, image segmentation techniques were applied to achieve a better labelling and visualisation of the detected shapes.

After all the reconstructed images were analysed with the previously described process, size estimation was carried out. Having that the simulated measurements were made over a predefined area with known dimensions, the size of the reconstructed images could be inferred by dividing the width of the scan surface (in mm) by the width of the image (in pixels). With this, the size of the bounding box containing all the detected shapes could be calculated, giving as a result the length and width of the detected defects. For the estimation of the defects depth, a different approach was followed, where a linear relationship between the average strength of the MFL signals and the depth of defects was explored with more simulations.

Chapter 6

Discussion

6.1 Simulation Results

The MFL imaging method proposed here shows promising results for both defect detection and size estimation. This methodology was implemented to detect a rectangular crack, similar to that of the experimentation, and irregular defects composed of a combination of ellipsoidal shapes. Results of these simulated MFL models are described below.

6.1.1 Defect Detection, Length and Width Estimation

The proposed MFL imaging methodology has proven accurate for both for regular defects and irregular defects detection. This approach was tested with the initial rectangular crack and with a multiple neighbouring irregular defects scenario. Afterwards, the sizes of the detected shapes were estimated. The results of both processes are discussed below.

Rectangular Crack

The images contained in Fig. 6.1 show the simulated rectangular crack on the left and its corresponding detected blob on the right. Moreover, it can be seen here that the shape labelled with the number 1 has a coarse rectangular shape in the centre of the image with a similar size than the rectangular crack. Furthermore, there are

smaller irregular shapes in this image that correspond to the residual contours (i.e. noise) detected by the technique presented in this thesis. Finally, the residual noise also affects the main object in the image, and this is due to the harsh changes on the MFL signals that are interpreted as contours by the Canny edge detector applied here.

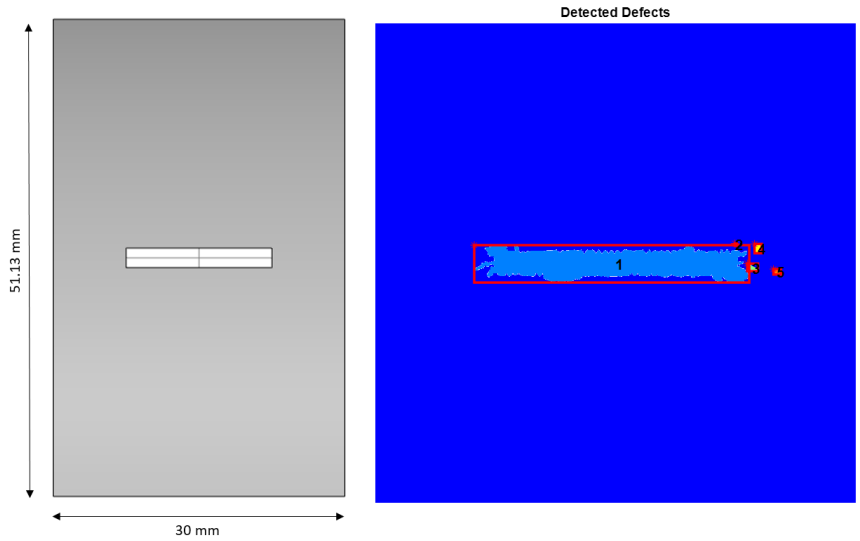


Figure 6.1: Simulated rectangular crack (left) and Detected shapes in MFL image (right)

The dimensions of the simulated rectangular crack, taken from the reference rectangular crack on the experimental pipe, and the estimated width and length of the detected defect are both shown in Table 6.1. The high relative error shown here is, as described before, due to the residual noise in the images. However, when scanning irregular defects, the relative error mostly decreases.

Dimension	Real Value	Estimated Value	Relative Error
Length	15 mm	17.18 mm	14.53 %
Width	2 mm	2.33 mm	16.50 %

Table 6.1: Size estimation for rectangular crack

Irregular Defects

Similarly, defect detection was achieved accurately for irregular shapes. In this thesis, a case with multiple neighbouring defects was designed, which included two

irregular defects and two simple, ellipsoidal defects. As mentioned before, the width, length, and location of these four defects were generated randomly within the simulation software and small adjustments were performed so that these random shapes were in contact with the surface to be scanned and to analyse several combinations of parameters. This can be seen in Fig. 6.2, where the simulated irregular defects are shown on the left, whereas the defected shapes are on the right. It can be seen here that the overall shape of the defects was contained in the MFL signals and therefore they could be detected by the proposed imaging method. No major residual noise is present in this image.

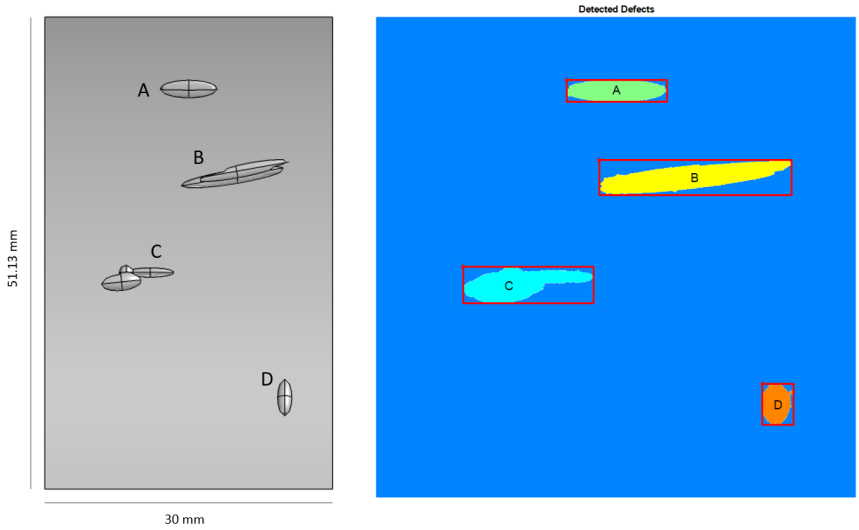


Figure 6.2: Simulated irregular defects (left) and Detected defects in MFL image (right)

Once these multiple shapes were identified, their dimensions were estimated. These parameters are contained in Table 6.2, which also includes their real dimensions and their corresponding relative error. It can be seen here that accurate length estimation was achieved up to 6.44 % and width estimation up to 13.35 %. However, this table also shows some high relative error values.

The proposed MFL imaging method shows promising results regarding defect detection, producing accurate identification of the simulated defects. Furthermore, the accuracy and reliability of the size estimation algorithm is dependent on the size of the defects as described below. It was also confirmed that analysing the three MFL components is needed and useful for better defect quantification. However, the highly varying relative error shown in Table 6.2 suggests that the size of the small

Defect	Real Values	Estimated Values	Relative Error
A	L = 5.9 mm	L = 6.28 mm	6.44 %
	W = 1.81 mm	W = 1.37 mm	24.30 %
B	L = 10.92 mm	L = 12 mm	9.89 %
	W = 2.9 mm	W = 2.21 mm	23.79 %
C	L = 7.56 mm	L = 8.18 mm	8.2 %
	W = 2.62 mm	W = 2.27 mm	13.35 %
D	L = 3.78 mm	L = 2.54 mm	32.80 %
	W = 1.48 mm	W = 1.94 mm	31.08 %

Table 6.2: Dimensions of Irregular Defects

size of defects has a negative effect on the accuracy of the proposed methodology. It should be emphasised here that these simulated irregular defects are mostly below the 10 mm size range. Furthermore, the MFL method is applied for corrosion defects detection and this type of defects is usually big in shape due to the combination of several neighbouring defects. Literature has focused on studying defects of standardised shape and size, which are proportional to the thickness of the pipe (t), as referenced by [42], where a 9.5 mm thick pipe was examined and the length and width of all the studied defects depended on this parameter of the pipe. For example, a $1t \times 1t$ defect is a circular defect with a diameter as long as the thickness of the pipe. Following this idea, the dimensions of the defects analysed in this thesis should not be smaller than the thickness of the examined pipe ($t = 2.5$ mm). However, as shown in Tables 6.1 and 6.2, some of the simulated defects did not meet this condition and coincidentally the estimation of these smaller sizes present the higher relative error values.

In order to verify that the previous results are caused by the small size of the simulated irregular defects, a new simulation was conducted in which only standard ellipsoidal defects were modelled. The rest of the parameters are identical to those of the previous simulations presented in this thesis. Fig. 6.3 shows 9 new ellipsoidal defects with their dimensions given in terms of the pipe wall thickness ($t = 2.5$ mm), the depth of these defects ranges from 10% t to 80% t. Furthermore, Fig. 6.4 shows

the MFL response of these new defects. These plots confirm that the MFL response is dependent on the size and depth of defect since the intensity of the MFL signals is higher on the bigger defects.

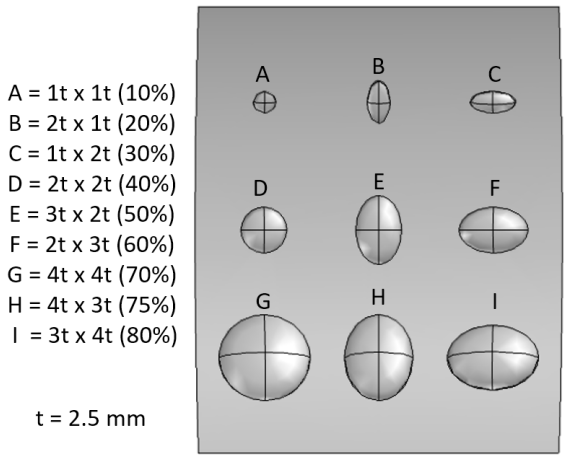


Figure 6.3: Standard ellipsoidal defects

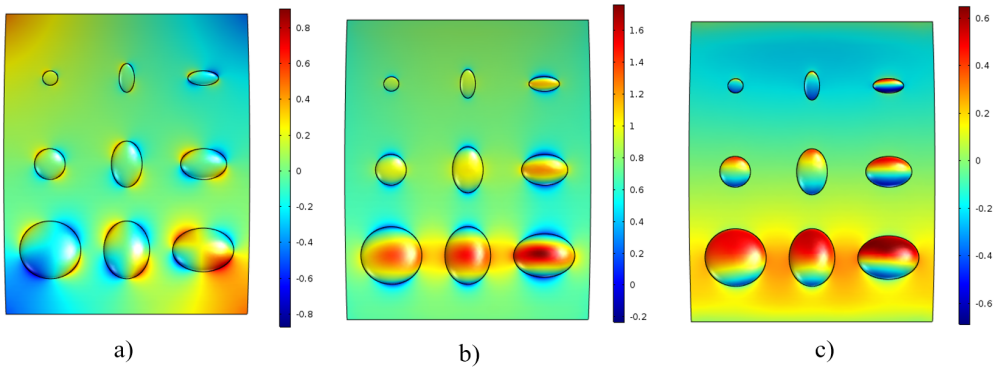


Figure 6.4: MFL response of standard ellipsoidal defects. a) Bx Component. b) By Component. c) Bz Component

Once the simulation was completed, the proposed MFL imaging method was tested with the simulated results. Defect detection was achieved accurately as shown in Fig. 6.5, whereas the estimated width and length of the detected shapes are shown in Table 6.3. The accuracy achieved by studying standardised defects is much higher than that of the irregular defects. These results suggest that both the complex geometry and size of the defects have an important influence over the MFL signals, and could explain why literature has focused on studying defects of regular shapes and standard sizes.

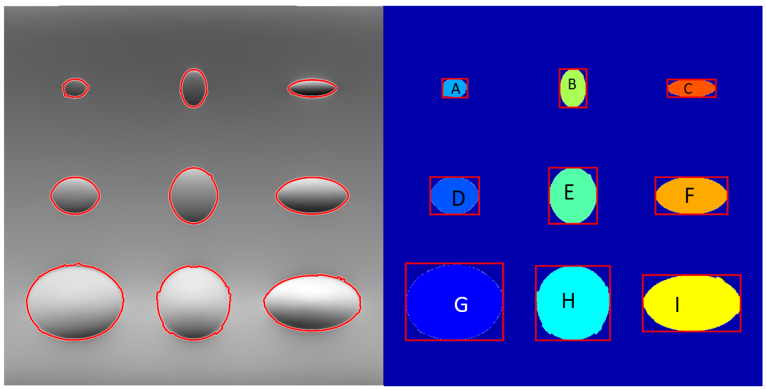


Figure 6.5: Detected Standard Ellipsoidal Defects

Defect	Real Values	Estimated Values	Relative Error
A	L = 2.5 mm	L = 2.65 mm	6 %
	W = 2.5 mm	W = 2.54 mm	1.6 %
B	L = 5 mm	L = 5.23 mm	4.6 %
	W = 2.5 mm	W = 2.83 mm	13.2 %
C	L = 2.5 mm	L = 2.39 mm	4.4 %
	W = 5 mm	W = 5.09 mm	1.8 %
D	L = 5 mm	L = 5.16 mm	3.2 %
	W = 5 mm	W = 5.08 mm	1.6 %
E	L = 7.5 mm	L = 7.62 mm	1.6 %
	W = 5 mm	W = 5.06 mm	1.2 %
F	L = 5 mm	L = 5.04 mm	0.8 %
	W = 7.5 mm	W = 7.61 mm	1.46 %
G	L = 10 mm	L = 10.26 mm	2.6 %
	W = 10 mm	W = 10.47 mm	4.7 %
H	L = 7.5 mm	L = 7.82 mm	4.26 %
	W = 10 mm	W = 10.11 mm	1.1 %
I	L = 7.5 mm	L = 7.66 mm	2.13 %
	W = 10 mm	W = 10.31 mm	3.1 %

Table 6.3: Dimensions of Standard Defects

6.1.2 Depth Estimation

As mentioned before, estimating the depth of the detected defects is the most important aspect of the MFL method since this parameter is crucial for the correct operation of a given pipe. Furthermore, it was also stated that this parameter can be calculated by obtaining the linear relationship between the average strength of MFL signals and the depth of a defect. This operation can be performed within COMSOL by iterating the depth of the defect. The output is a table and its corresponding graph, as shown in Fig. 6.6 containing the 32 iterations of this simulation. However, this graph shows that the calculated relationship is not linear as expected.

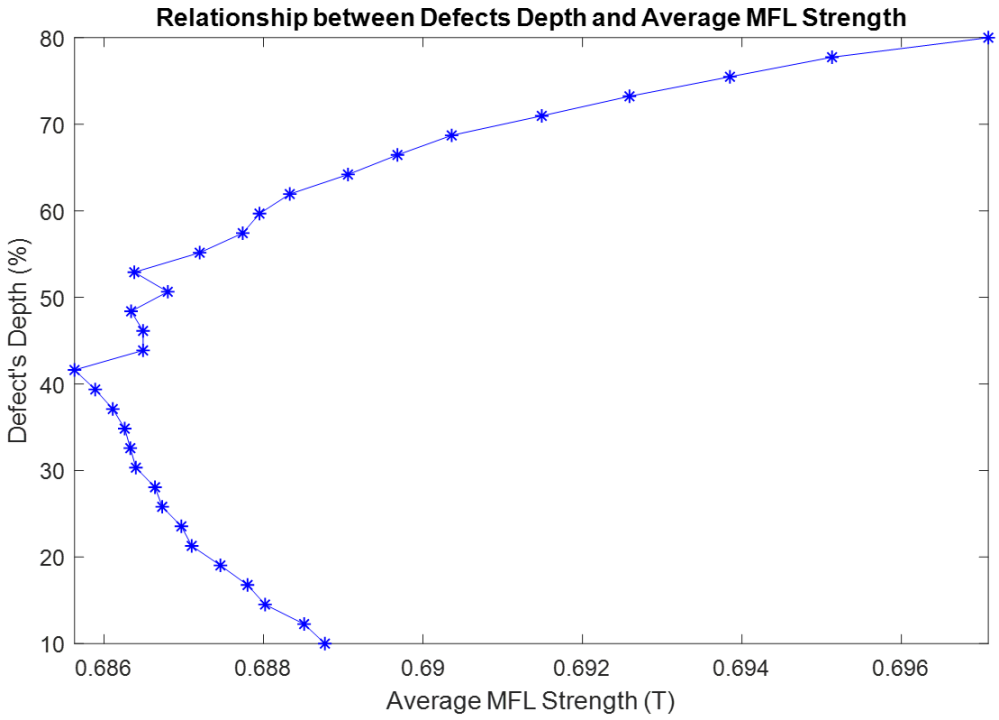


Figure 6.6: Relationship between defect's depth and average MFL signal strength

This approach was originally tested on the flat surface of a tank floor that was also thicker than the pipe being studied in this thesis. Moreover, the publication that investigated this approach focused on a single rectangular defect with uniform depth. This could explain why the results are different. Furthermore, in order to explore if the thickness of the specimen influences on these results, an additional simulation was performed. The original model was re-scaled to have a new, bigger pipe with a diameter of 406 mm and a thickness of 12.7 mm. This new thickness is

similar to that of the tank floor from which this approach is being replicated. The dimensions of the simulated crack were also modified so that they were similar to those of the crack studied in the aforementioned study. The new rectangular crack is 48 mm long and 24 mm wide. Its depth was iterated 32 times within a range of 10-80% of the pipe's thickness. The scanned surface was a 110 mm × 100 mm rectangular area. The result of this new simulation is shown in Fig. 6.7

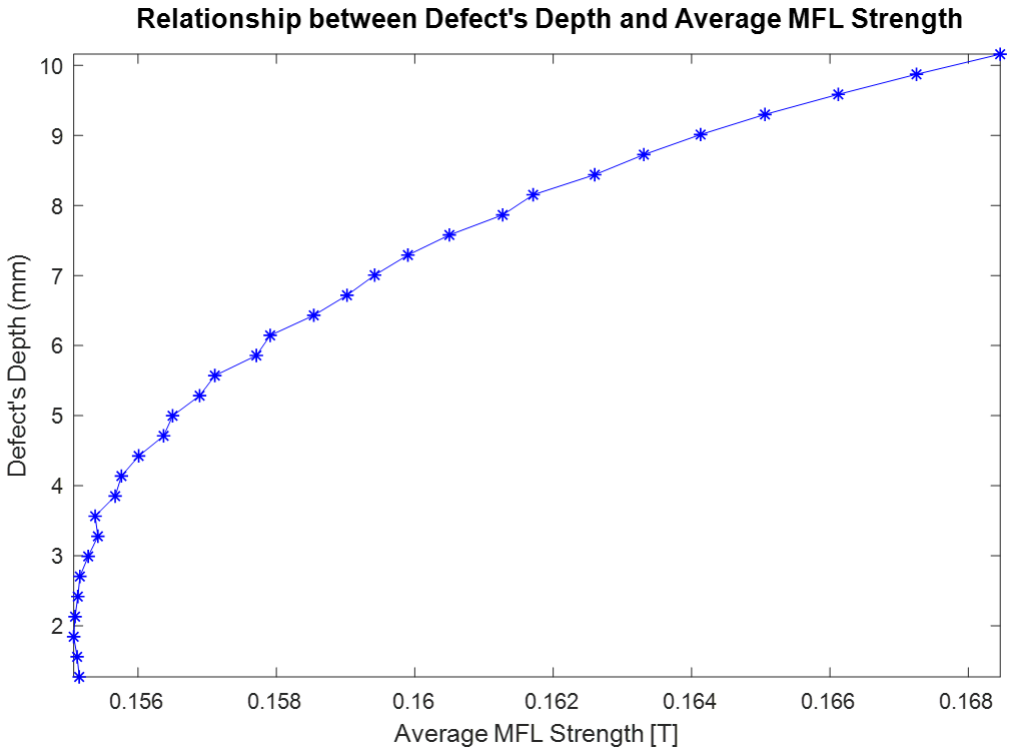


Figure 6.7: Relationship between defect's depth and average MFL signal strength

This graph confirms that the size and thickness of the specimen does have an influence over the aforementioned relationship since the new graph has smoother changes. Furthermore, it could also be confirmed that the curvature of the scanned surface affects this relationship since, once more, the linear relationship could not be found. After implementing curve fitting techniques, it was determined that the previous graph describes a more complicated function, which is given by the following expression:

$$f(x) = \frac{p_1 x + p_2}{x^4 + q_1 x^3 + q_2 x^2 + q_3 x + q_4} \tag{6.1.1}$$

where x is the input average MFL strength, $f(x)$ is the depth of the defect (D), and having $p_1 = 3812$, $p_2 = -589.8$, $q_1 = 174.7$, $q_2 = -1509$, $q_3 = 515.7$ and $q_4 = -44.24$ as coefficients.

This newly found expression was used to estimate the depth of a rectangular defect with the same dimensions used before, except for the depth, which was set to a different value than those used to find the previous function. Afterwards, another simulation was conducted in order to test this approach with a single ellipsoidal defect, and one more simulating an irregular defect. The length of the ellipsoidal defect was 80 mm and its width 24 mm. The length of the irregular defect was set to 59.88 mm and its width to 32.79 mm. The three shapes that compose this irregular defect have different depths, and only the maximum depth is considered for this test. These three defects can be seen in Fig. 6.8, whereas the results from these simulations are shown in Table 6.4 and are marked in Fig. 6.9.

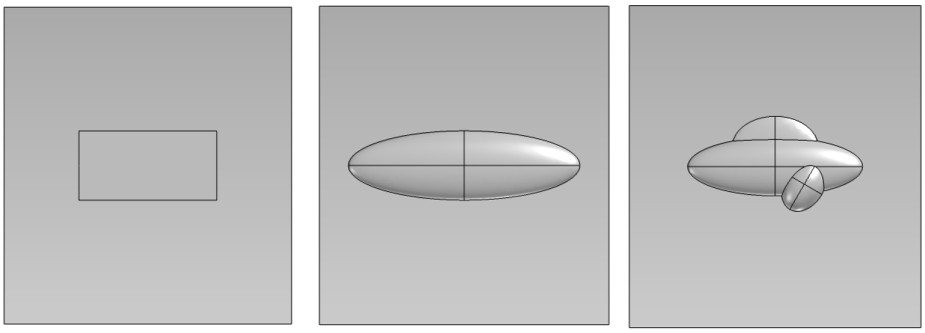


Figure 6.8: Test Defects for Depth Estimation

Defect	Real Depth (Max)	Estimated Depth	Relative Error
Rectangular	5.33 mm	5.24 mm	1.68 %
Ellipsoidal	5.33 mm	5.69 mm	6.75 %
Irregular	6.3 mm	5.34 mm	15.23 %

Table 6.4: Real Depth vs. Estimated Depths

These results suggest that the approach discussed in this section is reliable when working with simple defects such as rectangular and ellipsoidal shapes. However, when a defect composed of several shapes with different widths, lengths, depths and orientations is quantified, the method can not correctly estimate the maximum

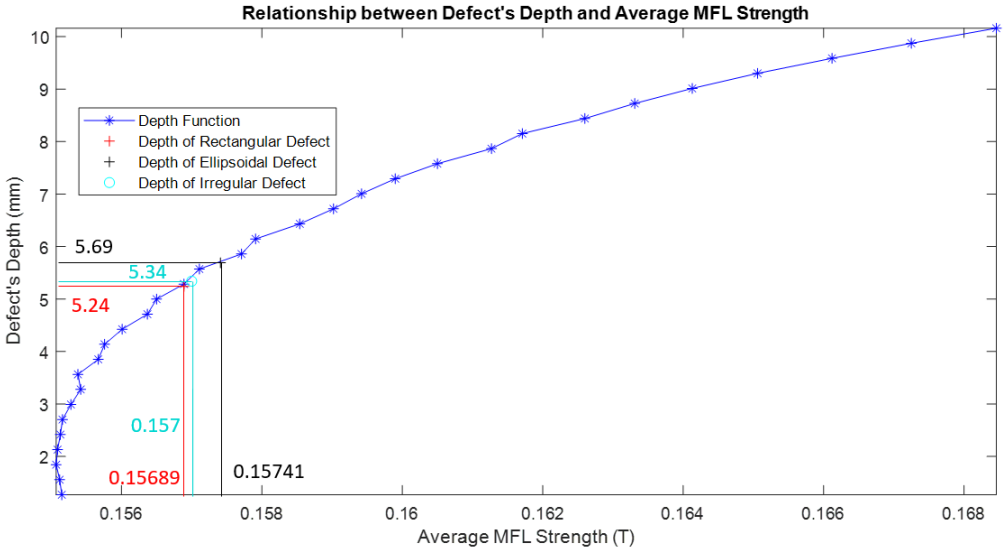


Figure 6.9: Estimated Depth of the Tested Defects

depth of said defect. Hence, a different method should be investigated in order to accurately estimate the depth of irregular defects. Finally, this approach should also be applied in a multiple defects scenario.

6.2 Experimentation Results

The proposed imaging method was implemented on the MFL signals collected from the experimentation. 2D images were successfully reconstructed containing the several measurements that were performed. As mentioned before, the experiments in this thesis consisted of performing 36 line scans with around 3500 samples, covering a predefined area to be scanned. These multiple scans were used to create a surface plot for each of the MFL components (excluding the Bz component due to the null information obtained from it, as explained before), which are shown in Fig. 6.10, including the Bx and By components of both sensors. These plots correspond to the multiple measurements made to the pipe having a rectangular crack. The colour code on these plots indicates the intensity of the signals. The X-axis of these plots correspond to the number of samples, whereas the Y-axis contains every individual line scan. Each XY pair represents a specific point of the predefined scan surface on the real steel pipe.

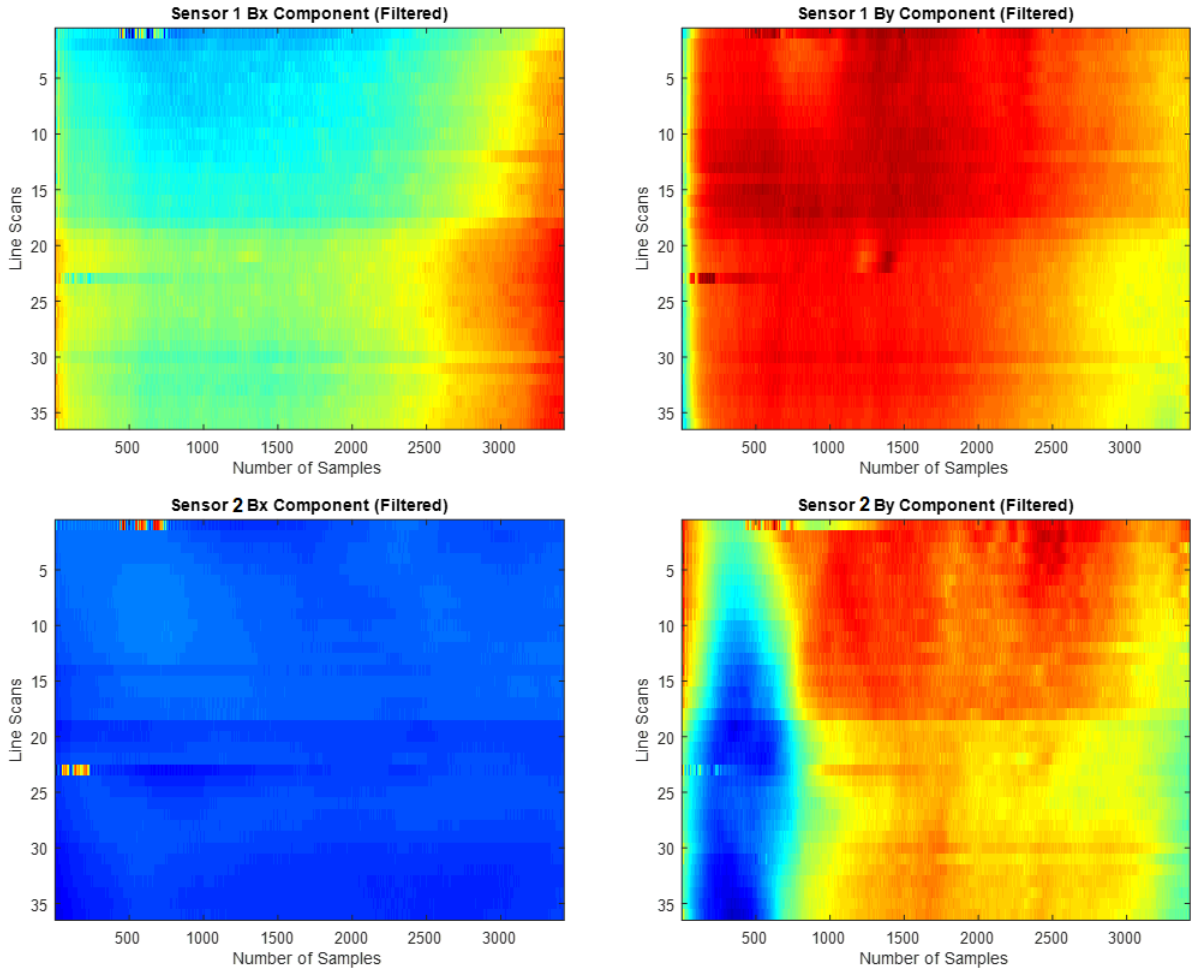


Figure 6.10: Surface plots of experimental data

The previous plots indicate that the modified MFL monitoring tool proposed in this thesis effectively detected the magnetic signals coming from the steel pipe. The variations on the intensity of the magnetisation of the pipe are evident, which also indicates that the utilised 3D GMR sensors are suitable for MFL measuring applications. However, based on the results of the simulations shown in 5, the plots above should show the MFL response of the rectangular crack and the different peaks and valleys should be easily visible and identifiable with eye inspection. These results suggest that applying the MFL imaging methodology proposed in this thesis to the reconstructed MFL images would not produce the expected defect detection as that achieved with the simulations. This result would limit one of the objectives of this thesis, which is to explore the MFL method using the three components of the magnetic signals. Additionally, as stated before, the reconstructed images

for the Bx and By components of both sensors did not show the expected results compared to those of the simulations, neither with the raw signals nor when they were filtered. Other investigations showed that defects are easily identifiable to the human eye even in the raw signals, and better visualisation was achieved after signal de-noising [13, 22, 30].

These results might be caused because of the small diameter and thickness of the pipe being inspected here, as well as the size of the crack defect itself. The reviewed literature mostly focused on much larger and thicker pipes, also studying longer, wider and deeper defects. The diameter of the pipe studied in this thesis is at least 4 times smaller than that of the reviewed literature and the standard pipes used in real life contexts, and the thickness is at least 5 times thinner. Moreover, the area of the defect examined in this investigation is at least 38 times smaller. These differences also indicate that the size of the specimen and the size of the defects should be taken into account for the MFL method, and that these parameters could pose as the limiting factors of this inspection technique. Furthermore, these ideas could be validated by the last simulation presented in the previous section, where a larger, thicker pipe with bigger defects was studied. The results of said simulation showed that the relationship between the average MFL strength and the depth of defect is affected by both the size of the pipe and defect. This could also explain why the literature has mostly concentrated on the aforementioned larger pipeline networks and fuel containers (i.e. tanks).

Although the entire proposed methodology could not be tested with experimental data due to these limitations, some aspects of it were proven correct. Examples of this are that 3D GMR sensors are reliable enough to be used in the MFL method, and that MFL images can also be reconstructed from the measured signals. Larger arrays of this type of sensors could be implemented in the larger MFL inspection tools in order to generate bigger images with much better resolution, which would also have a positive impact on the result of the proposed MFL imaging method. Finally, the newly found depth function described in the previous section could also be implemented to estimate the depth of experimental defects detected on the larger pipes.

Chapter 7

Conclusions

The main objective of this thesis was to investigate the MFL method and its application as an NDT technique for defect detection on product transmission pipelines. In order to achieve this, the use of 3D GMR sensors was addressed to study the influence of the three orthogonal components of the magnetic signals. Furthermore, since previous work has mostly concentrated on regular-shaped defects, this study focused on analysing defects of irregular shapes and non-uniform depths. In order to detect these defects, both an FEM simulated model and an experimental model were designed.

The simulated model was used to develop the proposed methodology which uses imaging techniques in order to reconstruct 2D images from the simulated MFL signals. Defect detection was conducted by implementing image processing techniques such as edge detectors and morphological operations. Width and length estimation was performed by geometrical approximation, calculating the dimensions of the bounding box that encloses the detected defects, and converting these parameters from pixels to millimetres. Defect depth was estimated by obtaining a novel, modified depth function by calculating the average MFL strength and its relationship with the depth of defects.

The experimental model was constructed in order to replicate the results from the simulations and validate the methodology presented in this thesis. A modified MFL monitoring system proposed here was tested to determine whether the MFL method is applicable to smaller pipelines from the outer surface. Although the

results were not as expected, the use of 3D GMR sensors was proven accurate and reliable for measuring the three components of the MFL signals, and the collected measurements were successfully used to reconstruct 2D images. The entirety of the methodology described in this thesis could not be tested due to some limiting factors that were encountered.

One of the factors that negatively affected the results was the small size of the simulated irregular defects since, according to standards referenced in the literature, defects should not be smaller than the thickness of the pipe. The accuracy of the proposed methodology increased when it was tested with standardised defects. Moreover, the small size and thin thickness of the pipe being studied in the experimental model also affected the results. This was confirmed when an attempt of replicating a previously cited depth estimation approach was being conducted. In said approach, a depth function was defined by the linear relationship of the average strength of MFL signals and the depth of a given defect. This linear relationship could not be found with the parameters used in the elaboration of this study, and instead a more complex function was defined after running a new simulation where a larger, thicker pipe with an also bigger defect was run. This new depth function yielded accurate results in estimating the depth of three types of defects (rectangular, ellipsoidal, irregular), but became less precise as the complexity of the defect increased.

Furthermore, the results obtained by the multiple measurements that were conducted in the experimentation did not show any relevant information. This could also be due to the size and thickness of the pipe, but also the strength of the magnets could have affected the measurements since the Bz was saturated in every readings. Another possible explanation could be that the sensor used in these experiments was not entirely suitable for the purposes of this thesis.

Further investigation is needed in order to determine if the novel depth function can accurately estimate the depth of a defect found on the surface of a larger and thicker pipe with experimental MFL data. A revision of the strength of magnets and the number of 3D GMR sensors should also be considered in order to confirm that the size and thickness of the specimen has the negative effect on the MFL method

described before, which would also revalidate that 3D GMR sensors are reliable enough to be used in this inspection technique. Moreover, the proposed MFL imaging technique for defect detection and quantification has yet to be tested with real irregular defects, which could not be investigated here since relevant results were not acquired. The novel depth function could also be applied to the experimental MFL signals from a larger pipe with multiple irregular defects in future investigations. Finally, if sufficient data can be gathered, machine learning techniques could be applied in order to create a more robust methodology.

References

[1] U.S. Central Intelligence Agency, "The world factbook," 2019. Available: https://www.cia.gov/library/publications/resources/the-world-factbook/geos/print_uk.html, Accessed: November 21st, 2019.

[2] U.S. Energy Information Administration, "Country Analysis Brief: United Kingdom," 2018.

[3] BBC News, "HMRC: UK oil and gas tax revenues pick up," June 2018. Available: https://www.bbc.co.uk/news/uk-scotland-scotland-business-44660336, Accessed: June 23rd, 2019.

[4] Office for Budget Responsibility, "Tax by tax, spend by spend: Oil and gas revenues." Available: https://obr.uk/forecasts-in-depth/tax-by-tax-spend-by-spend/oil-and-gas-revenues/ Accessed: June 13th, 2019.

[5] COMAH Strategic Forum, "The Buncefield major incident 11 December 2005," 2015. Available: https://www.hse.gov.uk/comah/buncefield/buncefield-10-years-on.pdf, Accessed: June 23rd, 2019.

[6] NDT Resource Center, "Non destructive testing: Introduction." Available: https://www.nde-ed.org/EducationResources/CommunityCollege/NDTIntro/cc_intro001.htm, Accessed: December 5th, 2019.

[7] The American Society for Nondestructive Testing, "Recommended practice No. SNT-TC-1A: Personnel qualification and certification in nondestructive testing," Technical Report, The

American Society for Nondestructive Testing, 2011. Available: https://inspectortrainingacademy.learningcart.com/uploads/SNT_TC_1A_2011.pdf, Accessed: December 5th, 2019.

[8] M. Coramik and Y. Ege, "Discontinuity inspection in pipelines: A comparison review," *Measurement*, vol. 111, pp. 359 – 373, 2017.

[9] A. Khodayari-Rostamabad, J. P. Reilly, N. K. Nikolova, J. R. Hare, and S. Pasha, "Machine learning techniques for the analysis of magnetic flux leakage images in pipeline inspection," *IEEE Transactions on Magnetics*, vol. 45, pp. 3073–3084, Aug 2009.

[10] G. Beers, "Energy pipeline: Tech talk — aliso canyon gas leak renews pressure on pipeline integrity measures," 2017. Accessed: November 21st, 2019.

[11] S. Huang and S. Wang, *New technologies in electromagnetic non-destructive testing*. Springer series in measurement science and technology, Springer, Singapore, 2016.

[12] S. Huang and W. Zhao, *Magnetic flux leakage theories and imaging technologies*. Advances in Electrical and Electronic Engineering, Berlin, Boston: De Gruyter, 2016.

[13] M. Afzal and S. Udpa, "Advanced signal processing of magnetic flux leakage data obtained from seamless gas pipeline," *NDT and E International*, vol. 35, no. 7, pp. 449 – 457, 2002.

[14] B. Mao, Y. Lu, P. Wu, B. Mao, and P. Li, "Signal processing and defect analysis of pipeline inspection applying magnetic flux leakage methods," *Intelligent Service Robotics*, vol. 7, pp. 203–209, Oct 2014.

[15] N. S. Kathir, G. Srinath, T. R. Kumar, and M. Prabhakaran, "Image based defect analysis using ndt," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 528–531, May 2018.

[16] A. Sophian, G. Y. Tian, and S. Zairi, "Pulsed magnetic flux leakage techniques for crack detection and characterisation," *Sensors and Actuators A: Physical*, vol. 125, no. 2, pp. 186 – 191, 2006.

[17] W. H. Hayt and J. A. Buck, *Teoría electromagnética (Spanish Edition)*, vol. 210-258. México: McGraw-Hill Interamericana Editores S.A. de C.V., 7th ed., 2006.

[18] M. M. O. Sadiku, *Elementos de Electromagnetismo (Spanish Edition)*. México: Alfaomega - Oxford, 2006.

[19] A. M. J.C. Drury, "Detection of natural gas pipeline defects using magnetic flux leakage measurements," 2000. Available: https://www.ndt.net/article/wcndt00/papers/idn701/idn701.htm Accessed: December 14th, 2020.

[20] S. Lu, J. Feng, F. Li, J. Liu, and H. Zhang, "Extracting defect signal from the mfl signal of seamless pipeline," in *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 5209–5212, May 2017.

[21] L. Zhang, F. Belblidia, I. Cameron, J. Sienz, M. Boat, and N. Pearson, "Influence of specimen velocity on the leakage signal in magnetic flux leakage type nondestructive testing," *Journal of Nondestructive Evaluation*, vol. 34, p. 6, March 2015.

[22] A. L. Pullen, P. C. Charlton, N. R. Pearson, and N. J. Whitehead, "Magnetic flux leakage scanning velocities for tank floor inspection," *IEEE Transactions on Magnetics*, vol. 54, pp. 1–8, Sep. 2018.

[23] J. Feng, S. Lu, J. Liu, and F. Li, "A sensor liftoff modification method of magnetic flux leakage signal for defect profile estimation," *IEEE Transactions on Magnetics*, vol. 53, pp. 1–13, July 2017.

[24] H. Q. Pham, Q. T. Trinh, D. T. Doan, and Q. H. Tran, "Importance of magnetizing field on magnetic flux leakage signal of defects," *IEEE Transactions on Magnetics*, vol. 54, pp. 1–6, June 2018.

[25] C. Mandache and L. Clapham, "A model for magnetic flux leakage signal predictions," *Journal of Physics D: Applied Physics*, vol. 36, no. 20, pp. 2427–2431, 2003.

[26] S. M. Dutta, F. H. Ghorbel, and R. K. Stanley, "Dipole modeling of magnetic flux leakage," *IEEE Transactions on Magnetics*, vol. 45, no. 4, pp. 1959–1965, 2009.

[27] S. M. Dutta, F. H. Ghorbel, and R. K. Stanley, "Simulation and analysis of 3-D magnetic flux leakage," *IEEE Transactions on Magnetics*, vol. 45, no. 4, pp. 1966–1972, 2009.

[28] D. A. G. Trevino, S. M. Dutta, F. H. Ghorbel, and M. Karkoub, "An improved dipole model of 3-D magnetic flux leakage," *IEEE Transactions on Magnetics*, vol. 52, no. 12, pp. 1–7, 2016.

[29] S. L. Huang, L. Peng, S. Wang, and W. Zhao, "A basic signal analysis approach for magnetic flux leakage response," *IEEE Transactions on Magnetics*, vol. 54, pp. 1–6, Oct 2018.

[30] J. Feng, F. Li, S. Lu, J. Liu, and D. Ma, "Injurious or noninjurious defect identification from mfl images in pipeline inspection using convolutional neural network," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, pp. 1883–1892, July 2017.

[31] S. Lu, J. Feng, H. Zhang, J. Liu, and Z. Wu, "An estimation method of defect size from mfl image using visual transformation convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 213–224, Jan 2019.

[32] F. Li, J. Feng, S. Lu, J. Liu, and Y. Yao, "Convolution neural network for classification of magnetic flux leakage response segments," in *2017 6th Data Driven Control and Learning Systems (DDCLS)*, pp. 152–155, May 2017.

[33] Y. Sun, B. Feng, S. Liu, Z. Ye, S. Chen, and Y. Kang, "A methodology for identifying defects in the magnetic flux leakage method and suggestions for

standard specimens," *Journal of Nondestructive Evaluation*, vol. 34, p. 20, June 2015.

[34] L. Peng, S. Huang, S. Wang, and W. Zhao, "A 3-D pseudo magnetic flux leakage (pmfl) signal processing technique for defect imaging," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–5, May 2019.

[35] A. Joshi, L. Udpa, S. Udpa, and A. Tamburrino, "Adaptive wavelets for characterizing magnetic flux leakage signals from pipeline inspection," *IEEE Transactions on Magnetics*, vol. 42, pp. 3168–3170, Oct 2006.

[36] A. V. Joshi, L. Udpa, S. Udpa, and A. Tamburrino, "Adaptive wavelets for characterizing magnetic flux leakage signals from pipeline inspection," in *2006 IEEE International Magnetics Conference (INTERMAG)*, pp. 652–652, May 2006.

[37] M. R. Kandroodi, B. N. Araabi, M. N. Ahmadabadi, F. Shirani, and M. M. Bassiri, "Detection of natural gas pipeline defects using magnetic flux leakage measurements," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, pp. 1–6, 2013.

[38] Pipeline Operators Forum, "Specifications and requirements for in-line inspection of pipelines," Technical Report, Pipeline Operators Forum, 2016. Available: https://www.pipelineoperators.org/downloadfilespublic.php?getfile=public-documents/01_2016_Version.pdf.

[39] M. R. Kandroodi, F. Shirani, B. N. Araabi, M. N. Ahmadabadi, and M. M. Bassiri, "Defect detection and width estimation in natural gas pipelines using MFL signals," in *2013 9th Asian Control Conference (ASCC)*, pp. 1–6, 2013.

[40] M. R. Kandroodi, B. N. Araabi, M. M. Bassiri, and M. N. Ahmadabadi, "Estimation of depth and length of defects from magnetic flux leakage measurements: Verification with simulations, experiments, and pigging data," *IEEE Transactions on Magnetics*, vol. 53, no. 3, pp. 1–10, 2017.

[41] H. M. Kim and G. S. Park, "A study on the estimation of the shapes of axially oriented cracks in CMFL type ndt system," *IEEE Transactions on Magnetics*, vol. 50, pp. 109–112, Feb 2014.

[42] H. M. Kim, C. G. Heo, S. H. Cho, and G. S. Park, "Determination scheme for accurate defect depth in underground pipeline inspection by using magnetic flux leakage sensors," *IEEE Transactions on Magnetics*, vol. 54, pp. 1–5, Nov 2018.

[43] L. Peng, S. Huang, S. Wang, and W. Zhao, "Three-dimensional magnetic flux leakage signal analysis and imaging method for tank floor defect," *The Journal of Engineering*, vol. 2018, pp. 1865–1870(5), November 2018.

[44] A. F. John, H. Yu, L. Bai, Y. Cheng, Y. Wang, and J. Mi, "Quantification of complex defects in magnetic flux leakage (mfl) testing using gradient gray level analysis based on least square approximation," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, May 2019.

[45] H. M. Kim, Y. W. Rho, H. R. Yoo, S. H. Cho, D. K. Kim, S. J. Koo, and G. S. Park, "A study on the measurement of axial cracks in the magnetic flux leakage NDT system," in *2012 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 624–629, Aug 2012.

[46] Y. Shi, C. Zhang, R. Li, M. Cai, and G. Jia, "Theory and application of magnetic flux leakage pipeline detection," *Sensors*, vol. 15, no. 12, pp. 31036–31055, 2015.

[47] H. Q. Pham, B. V. Tran, D. T. Doan, V. S. Le, Q. N. Pham, K. Kim, C. Kim, F. Terki, and Q. H. Tran, "Highly sensitive planar Hall magnetoresistive sensor for magnetic flux leakage pipeline inspection," *IEEE Transactions on Magnetics*, vol. 54, pp. 1–5, June 2018.

[48] L. Chen, P. Que, and T. Jin, "A giant-magnetoresistance sensor for magnetic-flux-leakage nondestructive testing of a pipeline," *Russian Journal of Nondestructive Testing*, vol. 41, pp. 462–465, Jul 2005.

[49] G. Piao, J. Guo, T. Hu, and Y. Deng, "High-speed inspection method fusing pulsed eddy current and magnetic flux leakage," in *2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 1–6, May 2019.

[50] J. Aguila-Muñoz, J. H. Espina-Hernández, J. A. Pérez-Benitez, F. Caleyo, and J. M. Hallen, "Crack detection in steel using a gmr-based mfl probe with radial magnetization," in *CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing*, pp. 104–108, March 2013.

[51] Y. Ege and M. Coramik, "A new measurement system using magnetic flux leakage method in pipeline inspection," *Measurement*, vol. 123, pp. 163–174, 2018.

[52] Y. Li, J. Wilson, and G. Y. Tian, "Experiment and simulation study of 3D magnetic field sensing for magnetic flux leakage defect characterisation," *NDT and E International*, vol. 40, no. 2, pp. 179 – 184, 2007.

[53] ST Microelectrinics, "Lsm9ds1 inemo inertial module: 3d accelerometer, 3d gyroscope, 3d magnetometer," Datasheet, 2015. Available: https://www.st.com/resource/en/datasheet/lsm9ds1.pdf, Accessed: December 5th, 2019.

[54] MFA COMO Drills, "919d series 35mm single ratio metal gearbox," Datasheet. Available: https://docs.rs-online.com/1531/0900766b813806a6.pdf, Accessed: December 5th, 2019.

[55] Texas Instruments, "L293x quadruple half-h drivers," Datasheet, 2016. Available: http://www.ti.com/lit/ds/symlink/l293d.pdf, Accessed: December 5th, 2019.

[56] "Understand, predict, and optimize physics-based designs and processes with COMSOL Multiphysics." Accessed: June 3rd, 2019.

[57] "COMSOL Multiphysics v5.4 user's guide," 2018.

[58] W. W. LaMorte, "Central limit theorem." Available: http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Probability/BS704_Probability12.html.

[59] M. Mirshra, "Understanding the Central Limit Theorem." Available: https://towardsdatascience.com/understanding-the-central-limit-theorem-642473c63ad8.

[60] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 679–698, Nov 1986.

[61] The University of Auckland, "Morphological image processing." Available: https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm, Accessed: June 23rd, 2019.

[62] S. Eddins, "The watershed transform: Strategies for image segmentation." Available: https://uk.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html.

Appendix A

Publications

The following is a one-page digest that was submitted and accepted for the Conference on Electromagnetic Field Computation (CEFC 2020), which includes the simulation part of this thesis. A full paper is being written to be submitted for the IEEE Transactions on Magnetics journal.

Simulated Irregular-Shaped Defect Detection and Quantification through Magnetic Flux Leakage Signals

Jesus David Bernal-Morales[1], Qing Wang[1]

[1]Durham University, Department of Engineering, Durham DH1 3LE, United Kingdom
jesus.d.bernal-morales@durham.ac.uk, qing.wang@durham.ac.uk

***Abstract*—To ensure the optimal state and performance of hydrocarbon transportation pipelines, a Non-destructive Evaluation (NDE) methodology that uses the Magnetic Flux Leakage (MFL) method is proposed. A simulated model is tested to generate MFL data that is used to produce 2D images and detect and quantify corrosion defects on a steel pipe. An experimental model using magnetoresistive (GMR) sensors is being constructed in order to validate the simulated results.**

***Index Terms*—Magnetic Flux Leakage, Magnetoresistive devices, Non Destructive Testing**

I. Introduction

The malfunctioning of hydrocarbon transportation pipelines can be costly and dangerous both for the environment and the population. Non-Destructive Evaluation (NDE) techniques are applied to ensure their optimal state and performance.

The MFL method is widely used for pipeline inspection, and relies on the high permeability of ferromagnetic materials. This method consists of magnetizing the pipe-wall so that magnetic field lines flow inside it but leak out in the presence of a discontinuity in the material [1]. Defects can be detected and quantified from MFL signals by examining their three orthogonal components *(Bx, By, Bz)*. Previous work has concentrated on the use of Hall-effect sensors to measure MFL signals and on the study of regular-shaped defects. [2], [3].

In this paper, a methodology is proposed in which simulated MFL data is generated and then analyzed in order to produce 2D images from which defects can be detected and quantified. An experimental model is being constructed in which 3D magnetoresistive (GMR) sensors are used to collect the three MFL components in order to validate the simulated results.

II. Methodology

A simulated environment was modelled in COMSOL Multiphysics, in which the surface of a pipe with four ellipsoidal shapes representing corrosion defects was magnetised. For this study, the magnetiser is located on top of the pipe. Simulated MFL signals are collected from the area below the magnetiser for each of the three components. These signals are then used to reconstruct 2D images from where defects can be detected. These images are treated with image processing techniques such as the Canny edge detector, and image segmentation algorithms are applied to these images. Once the defect is detected and properly labelled, size estimation is carried out. The simulated results are promising but the method has yet to be validated with experimental data, which will be obtained in a similar form using 3D GMR sensors in order to collect MFL signals for the three components.

III. Results and Discussion

The simulated pipe is shown in Fig. 1(a). Here, two defects have been modelled with ellipsoidal shapes whereas the third defect is a combination of three ellipsoidal shapes. As previously stated, images are generated for each component of the MFL signals. These are shown in Fig. 1(b). The proposed algorithm applies edge detectors and image segmentation techniques to these images and outputs a color-labelled image with the detected objects. by comparing Fig.1(a) and (c), it can be seen that these objects correspond to the simulated ellipsoidal defects and the irregular-shaped defect. These results suggest that the proposed methodology can accurately detect irregular-shaped defects. A size estimation algorithm is currently being tested for the detected shapes.

Finally, this methodology is being validated with experimentation in which GMR sensors will collect MFL data and 2D images will be generated from the scanned surface.

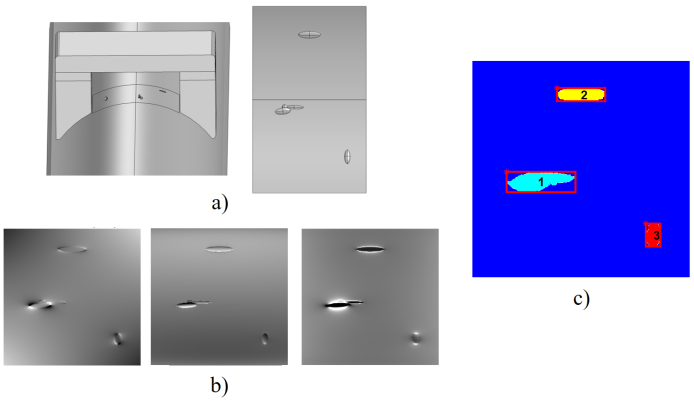


Fig. 1. Proposed methodology. a) Simulated pipe and defects. b) Images from MFL Components (Bx, By, Bz respectively). c) Color-labelled detected shapes

References

[1] S. Huang and W. Zhao, "Magnetic flux leakage theories and imaging technologies," 2016, p. 242.

[2] M. R. Kandroodi, B. N. Araabi, M. M. Bassiri, and M. N. Ahmadabadi, "Estimation of Depth and Length of Defects From Magnetic Flux Leakage Measurements: Verification With Simulations, Experiments, and Pigging Data," *IEEE Transactions on Magnetics,* vol. 53, no. 3, pp. 1-10, 2017, doi: 10.1109/TMAG.2016.2631525.

[3] S. Huang, L. Peng, Q. Wang, S. Wang, and W. Zhao, "An Opening Profile Recognition Method for Magnetic Flux Leakage *IEEE Transactions on Instrumentation and Measurement,* pp. 1-8, 2018, doi: 10.1109/TIM.2018.2869438.

Appendix B

Programming codes

B.1 Matlab Code

The Matlab coding performed for the modelling of the algorithms presented in this thesis is separated in various programmes that interact with each other as described as follows.

B.1.1 Main Code

This short code initialises important variables and calls on other codes and functions to carry out image reconstruction as well as defect detection and quantification.

```
%% Importing data from "filedir" location
%Calling 'importMFL' function to read data
[nFiles,nMtx,simData]=importMFL(filedir);
% nFiles = Number of text files found in folder (number of
    simulations)
% nMtx = Number of matrices to create  (number of iterations per
% simulation)
% simData = Cell structure where MFL data is stored
surfW = 30; % Known width of scanned surface (mm)
meshSz = [3000,3000]; % Size of grid to create (image size)
% Output text file for size estimations
fid = fopen('sizeEstimations.txt','wt');
% Initializing 4-D matrix the size of 3000*3000*3*nMtx
Z = zeros(meshSz(1),meshSz(2),3,nMtx);
```

```
14
15 % Loop to reconstruct MFL images per interation per simulation
16 for nSim=1:nFiles
17     for nStep=1:nMtx
18         % Creating mesh to reconstruct images
19         Z(:,:,:,nStep)=createMesh(nStep,nSim,meshSz,simData);
20         plotMFL % Shows the reconstructed images
21         findDef % Implements image processing to find and quantify
       defects
22       end
23 end
```

B.1.2 Importing Data

This function reads the text file(s) available in a specific location, scans the information and reorganises it in various matrices for further handling.

```
1 function [nF,nM,out] = importMFL(str)
2 % Reads a text file containing simulated MFL data generated in
      COMSOL Multiphysics
3
4 % INPUT
5 % str = location of the text file(s) to read
6
7 % OUTPUT
8 % nF = number of read files (simulations)
9 % nM = number of steps per simulation (iterations)
10 % out = Cell storing matrices with MFL data
11
12     cd(str); % Go to text file(s) location
13     txtFiles=dir('*.txt'); % Obtaining text file(s) names
14     nF=length(txtFiles); % Obtaining number of parametric sweeps
15     aux1=cell(nF,1); % Creating auxiliar variables
16     aux2=cell(nF,1); % to store read data
17
18     % Reading text files to obtain number of iterations per
       simulation, each text file is scanned looking for text
       characters
```

```
19    for i=1:nF
20        tmp=fopen(txtFiles(i).name,'r');
21        aux1{i,1}=textscan(tmp,'%s','Delimiter','\n');
22        aux2{i,1}=aux1{i,1}{1,1};
23        fclose(tmp);
24        % Storing appearances of the word "Nodes"
25        idx1(:,i) = find(contains(aux2{i,1},'Nodes'));
26    end
27    nM = length(idx1); % This length determines the number of
      iterations
28
29    % Each iteration has a different number of nodes (lines), this
      is determined by
30    % Comsol's solver. So a matrix is created which contains the
      number of
31    % nodes for each iteration
32    nLines = zeros(nF*nM,1); % Initialising matrix to store the
      different number of nodes (lines) per iteration
33    p=1; % Counter
34    for j=1:nF
35        for i=1:nM
36            tmp = string(aux2{j,1}(idx1(i,j)));
37            nLines(p,1) = str2double(regexp(tmp,'[\d.]+','match'));
38            p=p+1;
39        end
40    end
41
42    % Reading text files to collect MFL data per simulation, text
43    % files are scanned again to store them as number cells rather
      than
44    % string cells
45    for i=1:nF
46        tmp = fopen(txtFiles(i).name,'r');
47        aux1{i,1} = textscan(tmp, '%f %f %f %f %f %f', ...
48            'CommentStyle', '%', 'CollectOutput',true);
49        fclose(tmp);
50        aux2{i,1}=aux1{i,1}{1,1};
```

```
51      end
52
53      p=1; % Counter
54      out = cell(size(idx1)); % Creating output cell structure
55      % Storing MFL data in output matrices
56      % A vector is created for each MFL component
57      % The length of each vector is determined by the number of
        nodes of
58      % each iteration
59      for k=1:nF
60          nLn=1; % Start of new vector
61          endLn=nLines(p,1); % End of new vector
62          for j=1:nM
63              tmp=zeros(nLines(p,1),6,nM);
64              n=1;
65              for i=nLn:endLn
66                  tmp(n,:,j) = aux2{k,1}(i,:);
67                  n=n+1;
68              end
69          out{j,k} = tmp(:,:,j);
70          if p==length(nLines)
71              break
72          end
73          nLn=endLn+1;
74          p=p+1;
75          endLn=endLn+nLines(p,1);
76          end
77      end
78  end
```

B.1.3 Reconstructing Images

3

This function uses the previously rearranged information to create a grid where the MFL data is organised to reconstruct an image.

```
function out = createMesh(step,sim,sz,mtx)
% Creates a mesh by interpolating data in a 2D matrix
%INPUT
% step = current iteration number
% sim = current simulation number
% sz = size of the grid to create
% mtx = matrix from which data will be interpolated

% the columns of mtx are organized as follows
%    col 1    col 2      col 3    col 4  col 5  col 6
%  X coord   Y coord    Z coord    Bx     By     Bz

% OUTPUT
% out = grid that can be shown as an image

    tmp = mtx{step,sim}; % Reading matrices iteratively
    % Creating grid
    xlin = linspace(min(tmp(:,1)),max(tmp(:,1)),sz(1));
    ylin = linspace(min(tmp(:,2)),max(tmp(:,2)),sz(2));
    [X,Y] = meshgrid(xlin,ylin);

    % Initializing output 3D matrix
    out = zeros(sz(1),sz(2),3);

    % Scattering MFL data of each component
    x=griddata(tmp(:,1),-tmp(:,2),tmp(:,4),X,Y); %MFL Bx
    y=griddata(tmp(:,1),-tmp(:,2),tmp(:,5),X,Y); %MFL By
    z=griddata(tmp(:,1),-tmp(:,2),tmp(:,6),X,Y); %MFL Bz
    out(:,:,1) = x(:,:);
    out(:,:,2) = y(:,:);
    out(:,:,3) = z(:,:);
end
```

B.1.4 Defect Detection

This code analyses the previously generated images and implements image-processing techniques in order to find defects and quantify them.

```
% Printing to output text file
fprintf(fid,'**Simulation %d Step %d**\n', nSim, nStep);
str1 = ['MFL Components | Detected Defects | Simulation: ', num2str
    (nSim),' Step: ', num2str(nStep)];

%%% Defect Detection %%%

% Iterating per MFL component
for c=1:3
    switch c % Creating name for separators
        case 1
            str = 'Bx Comp';
        case 2
            str = 'By Comp';
        case 3
            str = 'Bz Comp';
    end
    % Converting image to grayscale
    tmp = mat2gray(Z(:,:,c));

    % Enhancing contrast
    n=4; avg = mean2(tmp);
    sigma = std2(tmp);
    Min=avg-n*sigma;
    Max=avg+n*sigma;
    if (Min<0), Min = 0; end
    if (Max>1), Max = 1; end
    imEq = imadjust(tmp,[Min Max],[]);

    % Detecting edges (Canny Algorithm)
    imBlur = imgaussfilt(imEq,3); % Gaussian filter
    imEdg = edge(imBlur,'Canny'); % Canny edge detector

```

```
33    % Implementing different morphological operations to each MFL
      component
34    switch c
35        case 1 %Bx comp
36            bw = imdilate(imEdg,strel('disk',3)); % Dilation
37            % Dilation and clearing border
38            bw = imclearborder(imdilate(bw,strel('line',8,45)));
39            bw = bwareaopen(bw,80); % Removing small objects
40            bw2 = imfill(bw,'holes'); % Filling holes
41            % Opening and removing small objects
42            bw3 = bwareaopen(imopen(bw2,strel('disk',10)),50);
43            bxEdges = bwareafilt(bw3,6); % Size filter
44            bwLabel = logical(bxEdges); % Labelling image
45            % Extracting properties of detected shapes
46            bxProps = regionprops(bwLabel,'Centroid', 'Area');
47
48        case 2 %By comp
49            bw = imdilate(imEdg,strel('disk',2));
50            bw = imdilate(bw,strel('line',3,45));
51            bw = imclearborder(imdilate(bw,strel('line',3,90)));
52            bw = bwareaopen(bw,50);
53            bw2 = imfill(bw,'holes');
54            bw3 = imopen(bw2,strel('disk',10));
55            byEdges = bwareafilt(bw3,6);
56            bwLabel = logical(byEdges);
57            byProps = regionprops(bwLabel,'Centroid', 'Area');
58
59        case 3 %Bz comp
60            bw = imdilate(imEdg,strel('disk',3));
61            bw = imclearborder(imdilate(bw,strel('line',3,45)));
62            bw = bwareaopen(bw,80);
63            bw2 = imfill(bw,'holes');
64            bw3 = imopen(bw2,strel('disk',10));
65            bzEdges = imclearborder(bwareafilt(bw3,7));
66            bwLabel = logical(bzEdges);
67            bzProps = regionprops(bwLabel,'Centroid', 'Area');
68    end
```

```
69 end
70
71 % Merging all the detected shapes in a single image
72 totalEdges = (bxEdges|byEdges|bzEdges);
73 % Erotion and removing small objects
74 bw4 = bwareaopen(imerode(totalEdges,strel('disk',5)),80);
75 % Obtaining image complement
76 imComp = imcomplement(tmp);
77 % Finding local minima
78 imMod = imimposemin(imComp, ~totalEdges | bw4);
79 % Applying watershed transform for image segmentation
80 def = watershed(imMod);
81 % Color labelling of detected shapes
82 Lrgb = label2rgb(def,'jet','w');
83 % Showing output image
84 figure, imshowpair(imEq,Lrgb,'montage'),hold on
85 % Showing boundaries of detected shapes on original image
86 imBnd = bwboundaries(bw4);
87 boundCount = size(imBnd,1);
88 visboundaries(imBnd)
89
90
91 %%% Size estimation %%%
92
93 % Getting properties of defects
94 stats = regionprops('struct',def,'Area','Centroid',...
95     'MajorAxisLength','MinorAxisLength','Orientation',...
96     'Eccentricity','BoundingBox');
97 stats(1,:)=[];
98
99 if ~isempty(stats) % If there are detected defects
100     defCount = length(stats); % Number of detected defects
101     pixW = surfW/size(tmp,2); % Pixel length in mm
102     % Iterating per detected defects
103     for r = 1:defCount
104         % Obtaining bounding box of current defect
105         thisBB = stats(r).BoundingBox;
```

```
106         % Drawing bounding box
107         plot(thisBB(1),thisBB(2),'r')
108         rectangle('Position',[thisBB(1),thisBB(2),thisBB(3),thisBB
    (4)],...
109             'EdgeColor','r','LineWidth',2)
110         % Obtaining width and length of current bounding box
111         thisLen = thisBB(3);
112         thisWid = thisBB(4);
113         % Calculating current defect's width and length
114         if (thisLen < thisWid)
115             % Converting pixels into milimeters
116             defLength = round(thisWid*pixL,4,'significant');
117             defWidth = round(thisLen*pixW,4,'significant');
118         else
119             % Converting pixels into milimeters
120             defLength = round(thisLen*pixW,4,'significant');
121             defWidth = round(thisWid*pixL,4,'significant');
122         end
123         % Printing to output text file
124         sprintf('Estimated Dimensions of Defect %d: \n Length: %f
    mm \n Width: %f mm \n\n',r,defLength,defWidth)
125         str3 = sprintf('%d', r); % Labelling defects in plot
126         text(stats(r).Centroid(1), stats(r).Centroid(2), num2str(r)
    , 'Color', 'k', 'FontSize', 12, 'FontWeight', 'Bold', '
    Interpreter', 'None');
127         fprintf(fid,'\t  Defect No. %d\n\t  Length: %.2f mm\n\n', r
    , defLength, defWidth);
128     end
129     hold off
130 else
131     defCount = 0;
132 end
```

B.1.5 PC - Arduino Interface

This code creates a communication interface to send commands to the Arduino board and then receive the measurements made by the sensors. This is achieved by working in conjunction with the programme shown in B.2

```
%% Serial Communication Matlab - Arduino
clearvars s % Clearing variables
% Initializing serial port
s = serialport("COM3",74880,"Timeout",25);
configureTerminator(s,"CR/LF");
disp('Initializing...') % Print to screen
loopCnt = 0; % Scans counter
% Iterating per number of line scans
while (loopCnt <= 36)
    i = 0; % Sample counter
    disp('Reading...')
    % Reading current state from serial port
    inputData = str2double(read(s,1,"char"));
    if(inputData >= 0)
        currState = inputData;
        if (currState == 0) % If State 0, write 'f' to serial port
            write(s,102,"char")
            flush(s); % Clearing port
        end
        if(currState == 1)
            disp ('FWD')
        end
        while(currState == 1)
            inputData = readline(s); % Reading measurements
            inputData = str2num(inputData)
            if(~isempty(inputData))
                if(inputData==2) % Checking if state changed
                    inputData = str2double(read(s,1,"char"));
                    currState = inputData;
                    break;
                end
                if(inputData==0) % Checking if arduino reset
```

```
33                  pause(0.5);
34                  write(s,102,"char") % Write 'f'
35                  flush(s);
36                  i = i - 1;
37                  continue
38              end
39              currData(i,:) = inputData; % Storing readings to
    matrix
40              i = i + 1; % Increasing sample counter
41
42          else
43              disp('Empty') % Print to screen if empty
44              inputData % Printing content of 'inputData' to
    screen
45              currData(i,:) = NaN;
46              i = i + 1;
47          end
48      end
49      if(currState == 2) %If State 2, return to initial position
50          disp('BWD')
51      end
52      while(currState == 2) % Storing last reading to cell
    structure
53          inputData = cell2mat(textscan(readline(s),'%f','
    Delimiter','CR/LF'));
54          if(inputData==0)
55             MFL_Data{loopCnt,1} = currData;
56             % Saving to file
57             save('filename.mat','MFL_Data','-append')
58             loopCnt = loopCnt + 1; % Increasing counter
59             disp('Ready for next scan?')
60             pause(); % Waiting for user keypress
61             disp(['Starting iteration ',num2str(loopCnt)])
62             pause(1); % Starting next loop
63             write(s,102,"char") % Write an f
64             flush(s);
65             currData = zeros(3400,6); % Clearing last readings
```

```
                break;
            end
        end
    end
end
```

B.2 Arduino Code

The following code creates an interface between the Arduino board and a PC with which the motor and sensors are controlled with a series of instructions sent from the PC to the Arduino.

```
// Importing needed libraries
#include <Wire.h>              // Libraries for
#include <SPI.h>               // Serial Communication
#include <SparkFunLSM9DS1.h>   // Library for driving the sensors
#include <avr/wdt.h>           // Library for CPU's watchdog timer

// Initializing variables
int motor_pins[] = {10, 11};   // Motor control pins
int LED = 13;                  // ON-OFF monitoring LED pin
int ledloopState = LOW;        // Current LED loopState
int REED_START = 5;            // Start reed switch pin
int REED_END = 6;              // End reed switch pin
int ON_switch = 2;             // General ON switch pin
int loopMaster;                // Control variable
int loopCnt = 0;               // Iterations Counter
unsigned int timeCnt = 0;      // Time counter
int loopState;                 // Current loop state
static unsigned long lastPrint = 0; // Keeping track of print time
LSM9DS1 sensor1;               // Creating LSM9DS1 objects
LSM9DS1 sensor2;               // to initialize sensors

// Defining I2C addresses for each sensor

#define LSM9DS1_M_1   0x1C  // Magnetometer Sensor 1
#define LSM9DS1_AG_1  0x6A  // Accelerometer and Gyroscope Sensor 1
```

```
#define LSM9DS1_M_2   0x1E  // Magnetometer Sensor 2
#define LSM9DS1_AG_2  0x6B  // Accelerometer and Gyroscope Sensor 2

// Output settings

#define PRINT_CALCULATED    // Print measurements in Gauss
#define PRINT_RAW           // Print raw measurements (binary)
#define PRINT_SPEED 100     // 100 ms between prints
// Earth's magnetic field compensation
#define DECLINATION 1.1166 // Declination (degrees) in Durham, UK.

// Functions definition
void printMag();        // Print readings
void motor_stop();      // Stops motor
void motor_forward();   // Motor moves clockwise
void motor_backward();  // Motor moves counter clockwise
void setup_sensors();   // Setting up sensors

// Setting up Arduino board. This code runs only once
void setup() {
  wdt_enable(WDTO_2S);  // Enabling watchdog timer count (2 sec)
  // Setting up OUTPUT pins
  pinMode(LED, OUTPUT);
  pinMode(motor_pins[0], OUTPUT);
  pinMode(motor_pins[1], OUTPUT);
  // Setting up INPUT pins
  pinMode(REED_END, INPUT_PULLUP);
  pinMode(REED_START, INPUT_PULLUP);
  pinMode(ON_switch, INPUT_PULLUP);
  // Setting up Serial port and Baudrate
  Serial.begin(74880);
  while (!Serial); // Waiting for serial to be ready
  // Setting up sensors
  setup_sensors();
  // Changing current state
  if (digitalRead(REED_START) == HIGH) loopState = 0;
}
```

```
63
64 // Main code. Runs continuously
65 void loop() {
66   loopMaster = !digitalRead(ON_switch); // Is main switch on?
67   if (loopMaster == LOW) loopState = 0; // Setting initial state
68   char readData = 0;  // Initializing serial input variable
69
70   // Finite State Machine
71   switch (loopState) {
72     case 0: // Initial State: Standby
73       motor_stop();
74       // Send current state to serial port
75       Serial.println(loopState);
76       // Reading serial port
77       while (readData != 'f') {
78         if (Serial.available()) {
79           readData = Serial.read();
80           // Changing current state
81           if (readData == 'f') { // Change to State 1
82             loopState = 1;
83           }
84           if (readData == 'b') { // Change to State 2
85             loopState = 2;
86             break;
87           }
88         }
89       }
90       break;
91
92     ////////////////////////////////////////////////////////
93
94     case 1: // State 1: Motor moves clockwise
95       // Send current state to serial port
96       Serial.println(loopState); // Printing current state
97       delay(100); // Delay 100 ms
98       // Motor keeps rotating until REED_END changes its state,
99       // this means that the magnetizer reached the end of the
```

```
100       // scan surface
101       while (digitalRead(REED_END) == HIGH && !digitalRead(
    ON_switch) == HIGH) {
102         if (timeCnt < 20000) wdt_reset(); // Reset watchdog timer
103         // Update sensor values when new data available
104         if ( sensor1.magAvailable() || sensor2.magAvailable()) {
105           sensor1.readMag(); // Reading magnetometer measurements
106           sensor2.readMag(); // in both sensors
107         }
108         // Printing last readings
109         if ((millis() - lastPrint) >= PRINT_SPEED) {
110           printMag(); // Prints six comma-separated values
111           motor_forward();
112           lastPrint = millis(); // Update last print time
113           // Change state of monitoring LED
114           ledloopState = !ledloopState;
115           digitalWrite(LED, ledloopState ? HIGH : LOW);
116         }
117         Serial.flush(); // Clearing serial port
118         timeCnt++; // Increase time counter
119       }
120       // Change to State 2
121       if (digitalRead(REED_END) == LOW) loopState = 2;
122       break;
123
124     ////////////////////////////////////////////////////////////
125
126     case 2: // State 2: Motor stops, then moves counter clockwise
127       motor_stop();
128       delay(1000);
129       // Motor keeps rotating until REED_START changes its state,
130       // this means that the magnetizer reached its original
    position
131       while (digitalRead(REED_START) == HIGH && !digitalRead(
    ON_switch) == HIGH) {
132         motor_backward();
133         Serial.println(loopState); // Printing current state
```

```
134         delay(60);
135         wdt_reset(); // Reset watchdog timer
136       }
137       Serial.flush(); // Clearing serial port
138       if (digitalRead(REED_START) == LOW) { // Change to initial
      state
139         loopState = 0;
140       }
141       timeCnt = 0; // Reset timer count
142       loopCnt++;   // Increase iterations counter
143       break;
144   }
145 }
146
147 // Implementing predefined functions
148
149 // Sending motor's control bits
150 void motor_stop() {
151   digitalWrite(motor_pins[0], LOW);
152   digitalWrite(motor_pins[1], LOW);
153 }
154 void motor_forward() { // Clockwise
155   digitalWrite(motor_pins[0], HIGH);
156   digitalWrite(motor_pins[1], LOW);
157 }
158 void motor_backward() { // Counter clockwise
159   digitalWrite(motor_pins[0], LOW);
160   digitalWrite(motor_pins[1], HIGH);
161 }
162
163
164
165 // Setting up sensors' parameters
166 void setup_sensors() {
167   sensor1.settings.device.commInterface = IMU_MODE_I2C; // Setting
      up I2C interface
```

```
168  sensor1.settings.device.mAddress = LSM9DS1_M_1;   // Assigning
      I2C
169  sensor1.settings.device.agAddress = LSM9DS1_AG_1; // addresses
170  // Setting magnetometer's scale can be 4, 8, 12, or 16
171  sensor1.settings.mag.scale = 16;
172  // Setting magnetometer's data sample rate can be 0-7
173  // 0 = 0.625 Hz  4 = 10 Hz
174  // 1 = 1.25 Hz   5 = 20 Hz
175  // 2 = 2.5 Hz    6 = 40 Hz
176  // 3 = 5 Hz      7 = 80 Hz
177  sensor1.settings.mag.sampleRate = 7;
178  // Setting magnetometer's performance can be any value between
      0-3
179  // 0 = Low power mode      2 = high performance
180  // 1 = medium performance  3 = ultra-high performance
181  sensor1.settings.mag.XYPerformance = 3;
182  sensor1.settings.mag.ZPerformance = 3;
183  sensor1.settings.mag.lowPowerEnable = false;
184  // Setting operatingMode can be 0-2
185  // 0 = continuous conversion
186  // 1 = single-conversion
187  // 2 = power down
188  sensor1.settings.mag.operatingMode = 0;
189  sensor1.begin(); // initializing sensor 1
190
191  ////////////////////////////////////////////////
192
193  sensor2.settings.device.commInterface = IMU_MODE_I2C;
194  sensor2.settings.device.mAddress = LSM9DS1_M_2;
195  sensor2.settings.device.agAddress = LSM9DS1_AG_2;
196  // mag scale can be 4, 8, 12, or 16
197  sensor2.settings.mag.scale = 16;
198  // mag readData rate can be 0-7
199  // 0 = 0.625 Hz  4 = 10 Hz
200  // 1 = 1.25 Hz   5 = 20 Hz
201  // 2 = 2.5 Hz    6 = 40 Hz
202  // 3 = 5 Hz      7 = 80 Hz
```

```
203   sensor2.settings.mag.sampleRate = 7;
204   // magPerformance can be any value between 0-3
205   // 0 = Low power mode      2 = high performance
206   // 1 = medium performance  3 = ultra-high performance
207   sensor2.settings.mag.XYPerformance = 3;
208   sensor2.settings.mag.ZPerformance = 3;
209   sensor2.settings.mag.lowPowerEnable = false;
210   // magOperatingMode can be 0-2
211   // 0 = continuous conversion
212   // 1 = single-conversion
213   // 2 = power down
214   sensor2.settings.mag.operatingMode = 0;
215   sensor2.begin();
216 }
217
218 void printMag() {
219   // Now we can use the mx, my, and mz variables as we please.
220   // Either print them as raw ADC values, or calculated in Gauss.
221
222 #ifdef PRINT_CALCULATED
223   // If you want to print calculated values, you can use the
224   // calcMag helper function to convert a raw ADC value to
225   // Gauss. Give the function the value that you want to convert.
226
227   // Printing Sensor 1 readings
228   Serial.print(sensor1.calcMag(sensor1.mx), 4);
229   Serial.print(F(",")); // Bx
230
231   Serial.print(sensor1.calcMag(sensor1.my), 4);
232   Serial.print(F(",")); // By
233
234   Serial.print(sensor1.calcMag(sensor1.mz), 4);
235   Serial.print(F(",")); // Bz
236
237   // Printing Sensor 2 readings
238   Serial.print(sensor2.calcMag(sensor2.mx), 4);
239   Serial.print(F(","));
```

```
240
241   Serial.print(sensor2.calcMag(sensor2.my), 4);
242   Serial.print(F(","));
243
244   Serial.print(sensor2.calcMag(sensor2.mz), 4);
245   Serial.println(F(","));
246 #endif
247 }
```