

MODELING THE MAGNETIC FIELD GENERATED BY CORRODING METALS

BY

MELANIE DAWN JOHNSON

B.S., University of Texas at Austin, 2006

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Electrical and Computer Engineering  
in the Graduate College of the  
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Adviser:

Associate Professor Patrick L. Chapman

## **ABSTRACT**

Damage caused by corrosion of reinforcing steel plagues civil infrastructure in the United States. Corrosion damage causes an estimated 6 to 10 billion dollars of damage per year to bridge decks alone. Despite this cost, no reliable nondestructive testing method exists to detect the presence of corrosion or to measure the rate at which corrosion occurs. Existing methods, such as visual inspection and half-cell potential measurements, are unable to fully characterize damage. A new proposed sensing technique based on magnetic field measurements may accurately measure both the state and rate of corrosion in reinforcing steel. In order to make use of the magnetic field measurements, a model of the magnetic field must be developed. In this thesis, three models simulate the magnetic field caused by corrosion. These models calculate the magnetic field based on random current distributions designed to reflect the behavior of currents in general micro-cell corrosion. The models use finite element analysis and two variations on Biot-Savart law to calculate the magnetic field. The results of the models undergo examination using power spectral density analysis. Finally, a technique that would detect the change in magnetic field is discussed as a possible application for the magnetic field sensors.

## **ACKNOWLEDGMENTS**

I would like to thank Dr. Patrick Chapman, Dr. John Popovics, and Gonzalo Gallo for all their help with this project. Without their assistance and guidance this work would not have been possible.

I would also like to thank my wonderful husband Danny and my loving parents for their continued support in all that I do.

## TABLE OF CONTENTS

LIST OF FIGURES .....	v
<b>1. INTRODUCTION</b> .....	1
<b>1.1 Project Motivation</b> .....	1
<b>1.2 Problem and Solution Overview</b> .....	1
<b>1.3 Previous Work and Literature Review</b> .....	2
<b>2. BACKGROUND INFORMATION</b> .....	6
<b>2.1 Corrosion Process</b> .....	6
<b>2.2 Corrosion Detection and Assessment Methods</b> .....	8
<b>2.2.1 Half-cell potential measurements</b> .....	9
<b>2.2.2 Linear polarization resistance</b> .....	10
<b>2.3 GMR Magnetometer Corrosion Detection</b> .....	11
<b>3. CORROSION MODELS</b> .....	14
<b>3.1 Random Current Distribution</b> .....	14
<b>3.2 Relative Time Steps</b> .....	14
<b>3.3 Modeling Methods</b> .....	15
<b>3.3.1 Finite element analysis</b> .....	15
<b>3.3.2 Nonferromagnetic plate model using Biot-Savart law</b> .....	21
<b>3.3.3 Current loop model</b> .....	28
<b>4. RESULTS</b> .....	31
<b>4.1 Simulation Results</b> .....	31
<b>4.2 Postprocessing Analysis</b> .....	34
<b>4.3 Further Analysis</b> .....	42
<b>5. CONCLUSIONS</b> .....	43
<b>REFERENCES</b> .....	44
<b>APPENDIX A: FEA MODEL CODE</b> .....	46
<b>A.1 Finite Element Model Main Program</b> .....	46
<b>A.2 Shape Function Matrix Function</b> .....	47
<b>A.3 Finite Element Analysis Function</b> .....	48
<b>APPENDIX B: PLATE MODEL CODE</b> .....	50
<b>B.1 Plate Model Main Program</b> .....	50
<b>B.2 Anode-Cathode Selection Function</b> .....	51
<b>APPENDIX C: CURRENT LOOP MODEL CODE</b> .....	55
<b>C.1 Current Loop Model Main Program</b> .....	55
<b>C.2 Magnetic Field Calculation Function</b> .....	56

## LIST OF FIGURES

Figure 2.1: Iron Corrosion Chemical Process.....	7
Figure 2.2: Generalized LPR Results Plot.....	10
Figure 2.3: GMR Magnetometer Diagram.....	12
Figure 2.4: GMR Magnetic Field Sensing Test Results.....	13
Figure 3.1: Circular FEA Model and Mesh.....	16
Figure 3.2: FEA Model Current Density Output in $A/m^2$ .....	19
Figure 3.3: FEA Model Magnetic Field Output in $Am^{-1}$ .....	19
Figure 3.4: FEA Model Magnetic Field Contour Plot in $Am^{-1}$ .....	20
Figure 3.5: Plate Model Geometry.....	21
Figure 3.6: Sample Current Distribution.....	22
Figure 3.7: Biot-Savart Law for Finite Current Segment.....	23
Figure 3.8: Plate Model versus Closed Form Solution.....	26
Figure 3.9: Magnetic Field versus Grid Size.....	27
Figure 3.10: Magnetic Field versus Differential Current Length.....	27
Figure 3.11: Current Loop in Plate Element.....	28
Figure 3.12: Rectangular Loop Problem.....	29
Figure 4.1: Magnetic Field Contributions from a Plate Model Current Distribution.....	32
Figure 4.2: 200-Point Simulation Data using the Plate Model.....	33
Figure 4.3: 200-Point Simulation Data using the Loop Model.....	33
Figure 4.4: 200-Point Simulation PSD using the Plate Model.....	34
Figure 4.5: Five Point Padded Signal using the Plate Model.....	36
Figure 4.6: PSDs of Padded Signals from the Plate Model.....	36
Figure 4.7: 1024-Point Simulation 1 through 4 PSDs using the Plate Model.....	37
Figure 4.8: 1024-Point Simulation 5 through 8 PSDs using the Plate Model.....	38
Figure 4.9: PSD Correlation.....	39
Figure 4.10 Pulse Train PSD from a 1024-Point Plate Model Simulation.....	40
Figure 4.11: Pulse Train Generated with Standard Deviation Threshold Value.....	41

## **1. INTRODUCTION**

### **1.1 Project Motivation**

Corrosion damage poses a serious threat to the structural integrity of concrete infrastructure. When steel embedded in concrete corrodes, rust, which has no strength and has a larger volume, gradually replaces the steel. The reduced strength of the structure diminishes further when the larger rust volume causes the surrounding concrete to crack. Once the concrete cracks, water and deicing salt can reach the steel more quickly and exacerbate the deterioration.

Corrosion of the infrastructure causes an estimated six to ten billion dollars worth of damage each year [1]. Current nondestructive methods of detection require electrical connection to the embedded steel and may only indicate the likelihood of corrosion in a structure or the rate of corrosion rather than the extent. Monitoring corrosion through its magnetic field may offer an alternative solution. This method allows the detection of localized areas of corrosion and does not require electrical connection to the embedded steel. However, to implement this method, a model of the corrosion magnetic field is needed to verify the sensing method.

### **1.2 Problem and Solution Overview**

The motion of electrons released in the corrosion chemical reaction forms a current in the conductive metal and therefore a magnetic field. However, the nature of this magnetic field is unknown. In order to make effective measurements and identify a magnetic field

caused by corrosion rather than environmental factors, a model of the field is needed. Once a model is developed, measurements can confirm that the model accurately depicts characteristics of the field generated by corrosion. Past research [2-6] reveals that previous models of corrosion focused on large scale corrosion current density and general electrical properties, but no small scale models were offered. The anodes and cathodes of corrosion currents vary as concentrations of enabling chemicals change, which makes the current distribution challenging to accurately predict on a microscopic scale.

Each of the three models developed for this project generates random currents based on a normal distribution and the geometry of the physical structure. The current distributions then undergo analysis either by finite element method or a calculation based on the Biot-Savart Law. Later sections explain these models in detail. The results from the simulations are analyzed for meaningful information through frequency analysis and other techniques.

### **1.3 Previous Work and Literature Review**

The magnetic field generated by corrosion has been previously measured by researchers at Vanderbilt University. Using a superconducting quantum interference device (SQUID) magnetometer, they produced results that clearly show a magnetic field in the presence of corrosion [2]. The SQUID magnetometer is a highly sensitive measurement tool and requires cooling with liquid helium to obtain super conduction. The corroding samples used in the experiments were magnetically shielded so that external magnetic noise did not interfere with the measurements. Under these conditions, the presence of

magnetic field above the corroding sample was detected. The results also indicated that the oxygen concentration in the environment increased corrosion and produced higher magnetic field intensities [2].

The GMR sensors employed in the proposed [3] sensing method are less sensitive than the SQUID magnetometer, but do not require extreme cooling to function. The samples used in the SQUID research were also highly shielded, making the small corrosion field easier to detect. The GMR sensors will be implemented in an unshielded environment and will have to extract the corrosion magnetic field from environment noise. Therefore, the model developed in the following sections will be essential to determining what signals should be extracted.

One of the key challenges in modeling the corrosion magnetic field involves determining the current distribution on the surface of the metal at a microscopic level. Several papers suggest methods of modeling the current at a macroscopic level, but none developed microscopic techniques. In particular, a paper by Kranc and Sagüés [4] develops a technique for modeling corrosion macrocells. The work models the diffusion of corrosion-inducing agents and the probability that corrosion begins at certain concentrations. From this, the corrosion currents can be determined. Once the corrosion current densities are known, the corrosion potentials in the rebar may be determined. However, this method is only applied to a network of rebar imbedded in concrete and gives results that are similar to half-cell potential measurement [4].

More work by Ghods, Isgor, and Pour-Ghaz focuses on determining the corrosion rate of imbedded steel, which corresponds to the current density on the rebar [5]. This approach focuses on dividing the rebar into anode and cathode regions based on prior work relating the ratio of anode to cathode area to maximum corrosion rate. One of the primary assumptions in the paper states that the anode and cathode regions are uniformly spaced on the rebar, which makes the solution inherently macroscopic and yields little information about the current distribution on the surface of the rebar [5].

Research by Warkus, Raupach, and Gulikers compares microcell corrosion to an electric circuit driven by the potential created by the oxidation and reduction chemical reactions [6]. The model developed in the work incorporates the corroding agent concentrations and is highly complex computationally. An analytical solution is only possible for contrived simple geometries. The final solution develops a current density for a single microcell, but the paper does not discuss the distribution of microcells on the surface of the steel [6].

Another interesting technique for measuring corrosion activity, presented in [7], focuses on monitoring the electrochemical noise to assess the state of corrosion. The experiment explained in the paper was able to discern fluctuations and transient in current noise measurements. These experimentally monitored changes in the current indicate that similar fluctuations should be visible in the magnetic field. However, because the paper focused on experimental results and system behavior rather than local conditions, it does not contribute any insight on how to model these effects. There is a distinct lack of

literature on techniques for modeling the microscopic surface currents and magnetic field generated by corrosion. The models developed in the following sections attempt to fill this gap.

## **2. BACKGROUND INFORMATION**

When metal corrodes, it generates an electromagnetic field due to the movement of electrons through a conductor released by a chemical reaction. The following section describes the electrochemical process that governs the corrosion of steel and the generation of the minute electromagnetic field.

### **2.1 Corrosion Process**

Metals exist at a relatively high energy state. Corrosion occurs as these high energy metals attempt to reach a lower energy level. All metals corrode, though some are more likely do so than others. Seemingly inert metals, such as stainless steel, simply corrode at very slow rates, while active metals like zinc corrode quickly. The willingness of a metal to corrode, also known as its activity, is determined by the number of valence electrons each atom of the metal has and by the energy required to remove these electrons. More active metals require less energy to remove their electrons and initiate the corrosion process [8-9].

Corrosion occurs due to a consuming electrochemical reaction between a metal and its environment. When a metal is placed in a corrosive environment, oxidation and reduction reactions slowly dissolve the metal, putting metal ions into solution and leaving behind electrons in the metal. This forms an anode at the site where the oxidation occurs. The free electrons may move freely through the conductive metal to another location where the reduction reaction occurs. In the reduction reaction, the electrons

combine with water and oxygen to form hydroxide. The metal ions in solution then react with the hydroxide to form a metal oxide, such as rust. The following equations outline the corrosion process for iron corrosion in an aqueous oxygen rich environment [8].

Figure 2.1 illustrates the corrosion process. The oxidation reaction is given by



and the reduction reaction is



The products of these reactions combine according to



and create iron oxide.

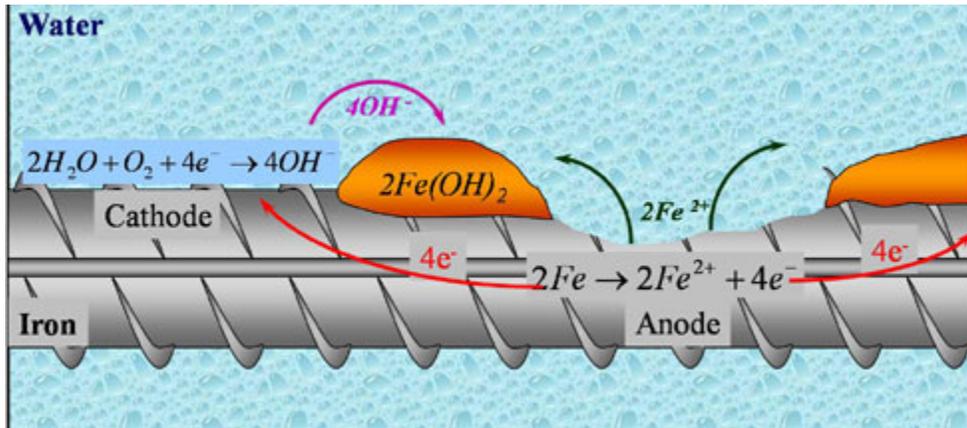


Figure 2.1: Iron Corrosion Chemical Process [10]

When embedded in concrete, steel corrodes very slowly because concrete provides an environment with very high pH. The high pH environment forms a passivating layer on the steel that prevents the oxidation and reduction reactions. However, as chloride ions and carbon dioxide seep into the concrete through its porous structure, this high pH environment is destroyed and corrosion commences. Carbon dioxide from the atmosphere gradually penetrates the concrete, reacting with the calcium in the concrete to

form calcium carbonate. The consumption of the calcium lowers the pH of the concrete, making corrosion more likely. The ingress of carbon dioxide can further decrease the pH of the concrete by releasing chloride ions that were bound to calcium in the concrete. Chloride ions also lower the pH of the concrete environment, but may also disrupt the passivating film and allow corrosion to initiate [9].

The reactions listed above occur commonly when water rich in oxygen permeates through the concrete. Other corrosion reactions are possible when the water does not contain oxygen. These reactions do not produce typical red rust, but do cause damage to the steel and the strength of the structure. All of the possible reactions involve the movement of electrons through the conductive metal bar and therefore may be detected by identifying their magnetic field [8-9].

## **2.2 Corrosion Detection and Assessment Methods**

Techniques exist that can detect corrosion within a structure; however, these methods may only indicate the likelihood of corrosion or the rate of corrosion and require direct electrical connections to the embedded steel structure. The following sections discuss two common methods of corrosion detection: half-cell potential measurements and linear polarization resistance testing.

### ***2.2.1 Half-cell potential measurements***

Half-cell potential measurements are one of the simplest ways to test for the likelihood of corrosion. When corrosion occurs, it generates a small potential difference that can be measured in comparison to a reference electrode. The potential difference of the corrosion reaction is the potential necessary to remove the electrons from the corroding metal.

The half-cell potential technique measures this small potential difference using a voltmeter and a standard reference electrode. The negative terminal of the volt meter must be electrically connected directly to the embedded steel rebar, while the reference electrode must be placed in contact with the surface of the concrete above the rebar.

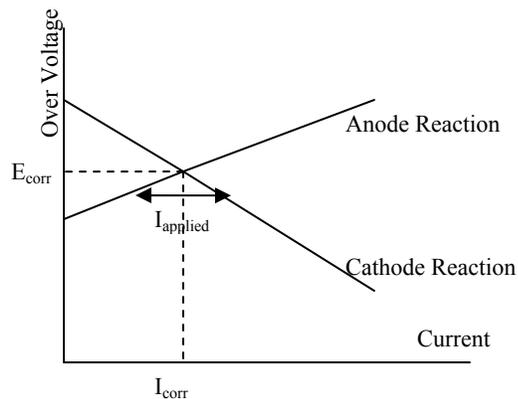
Typically, a copper-copper-sulfate electrode (CSE) is used in field tests. The voltmeter registers a voltage with respect to the CSE that indicates the corrosion potential. The magnitude of this voltage predicts the likelihood of corrosion in that region of the concrete. Usually, half-cell potential testing generates a map of potentials over a large area of the concrete. The map may then be used to predict regions where corrosion is likely. Generally, potential differences less than -350 mV indicate that corrosion is probably present [8-9].

The primary drawbacks associated with half-cell potential measurements include the required electrical connection to the embedded steel and the necessity of an electrolyte. Steel embedded in concrete may be inaccessible electrically unless provisions for testing

were made in the structure's design, and the measurements may be meaningless if there is no electrolyte, such as water, present in the concrete.

### 2.2.2 Linear polarization resistance

Unlike half cell potential measurements, linear polarization resistance (LPR) testing yields information on the rate of corrosion. To take linear polarization resistance measurements, a small current is applied to the corroding system and the resulting potential difference, or overvoltage, from the corrosion potential is measured with respect to a standard reference electrode using a voltmeter. The applied current is varied in increments to achieve small values of potential difference. The slope of the plotted values is inversely proportional to the corrosion rate, which can be derived from this point [8]. Figure 2.2 shows generalized plot for LPR measurements.



**Figure 2.2: Generalized LPR Results Plot**

Like half-cell potential measurements, LPR testing has some drawbacks. It requires direct electrical connection to the embedded steel and the presence of an electrolyte to

conduct the applied current. There can also be error in the calculation of the corrosion current density if incorrect approximations are made when deriving the value from the measurements. The actual slope of the polarization curves shown in Figure 2.2 may not be linear, which can lead to further error when a linear approximation is made.

Using both half-cell potential measurements and LPR, the general condition of embedded steel can be assessed. Half-cell potential measurements indicate the likelihood of corrosion, while LPR yields information on how quickly the steel is corroding. However, conducting both tests can be time-consuming and either test may have significant error. GMR sensor corrosion detection proposes a solution to measuring both the progress and intensity of corrosion with one tool and without direct electrical connection to the embedded steel.

### **2.3 GMR Magnetometer Corrosion Detection**

The proposed corrosion detection system makes use of giant magneto-resistive (GMR) sensors to sense the magnetic field generated by corrosion. Like the SQUID magnetometers in [2], these devices can detect small magnetic fields. However, unlike the SQUID sensor, GMR magnetometers are small, low cost, low power, and have high temperature stability [11]. This combination of characteristics makes GMR magnetometers a practical device for corrosion testing where environmental conditions and cost constraints would rule out other magnetometers [1]. The GMR system also has several advantages over the previously described corrosion detection methods. GMR

detection does not require an electrical connection to the embedded steel, nor does it require the presence of an electrolyte to conduct electricity through the concrete.

These sensors are composed of several layers of ferromagnetic and non-ferromagnetic material which change their resistance in the presence of a magnetic field when a biasing voltage is applied. Figure 2.3 displays a diagram of the sensors and how magnetic field affects their resistance.

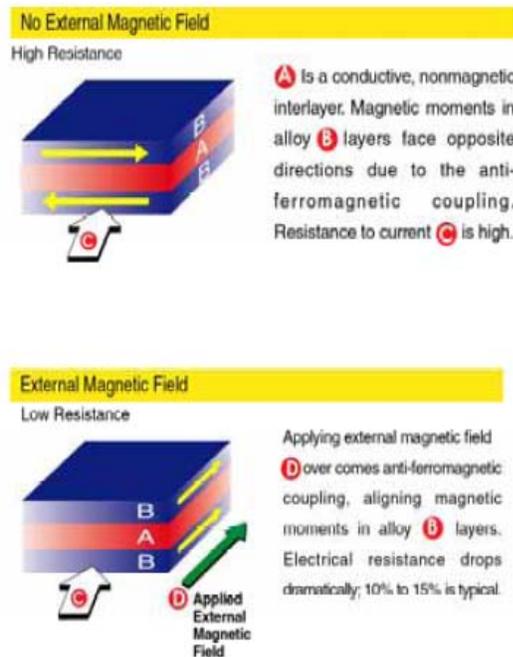
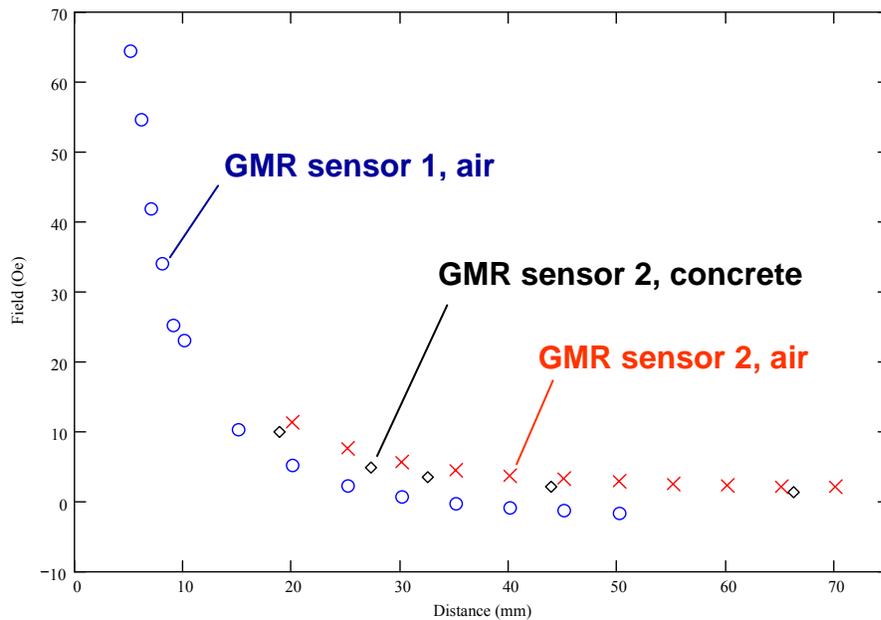


Figure 2.3: GMR Magnetometer Diagram [11]

The goal is to passively detect magnetic field using these sensors and determine the immediate corrosion rate. The sensors will also listen for deviation in an actively injected magnetic field from an external source to detect the extent of corrosion. In passive detection, the project plans to find correspondence between the magnetic field and the current density in the embedded steel. In active detection, the plan is to

determine the corrosion product thickness by detecting how the injected magnetic field changes when corrosion product is present. The amount corrosion product, such as rust, can help determine how much corrosion has taken place [3].

Preliminary results using the GMR sensors to passively measure the magnetic field of a sample of rebar show promising results. Rebar has residual magnetic field due to its forging process, but the sensor clearly shows this field decaying with distance. Figure 2.4 summarizes one test performed with a GMR magnetometer.



**Figure 2.4: GMR Magnetic Field Sensing Test Results [11]**

The primary challenge in implementing the GMR system is accurately sensing the tiny corrosion magnetic field and filtering it from background noise. The purpose of the models developed in the following chapters is to identify the characteristics of the magnetic field that will aid in sifting the meaningful sensor data from environmental and irrelevant noise.

### **3. CORROSION MODELS**

The three modeling techniques attempt to accurately predict the magnetic field generated by corrosion. In order to accomplish this, a model geometry and current distribution must be decided. With this information, the magnetic field intensity can be obtained through several types of analysis. The models developed in the following sections analyze the magnetic field using finite element analysis and the Biot-Savart law applied first to finite current segments and then to current loops on the surface of the geometry.

#### **3.1 Random Current Distribution**

The currents that result from the corrosion process are determined by the chemical reactions that take place at the metal's surface. The electrons released by the oxidation reaction can travel through the metal until they are absorbed by a reduction reaction. This means that anodes and cathodes appear and disappear on the metal's surface during the corrosion process in a seemingly random manner. For this reason, the simulations all start by generating currents based on a normal distribution. The normal distribution allows the simulation to generate currents based on a median and standard deviation that reflect known average values of corrosion current density.

#### **3.2 Relative Time Steps**

The simulation operates in time steps. For each time step random currents are generated and the solution engine calculates the resulting magnetic field. When a series of time steps is generated, the change in time between each step may be determined in

postprocessing. By changing the relative time between each step, the simulation can model corrosion currents changing at various rates. The rate of anode and cathode appearance in a corroding system is unknown. Postprocessing routines may change the value of the time step in an attempt to find correspondence between fluctuations in experimentally measured magnetic field and the simulation results.

### **3.3 Modeling Methods**

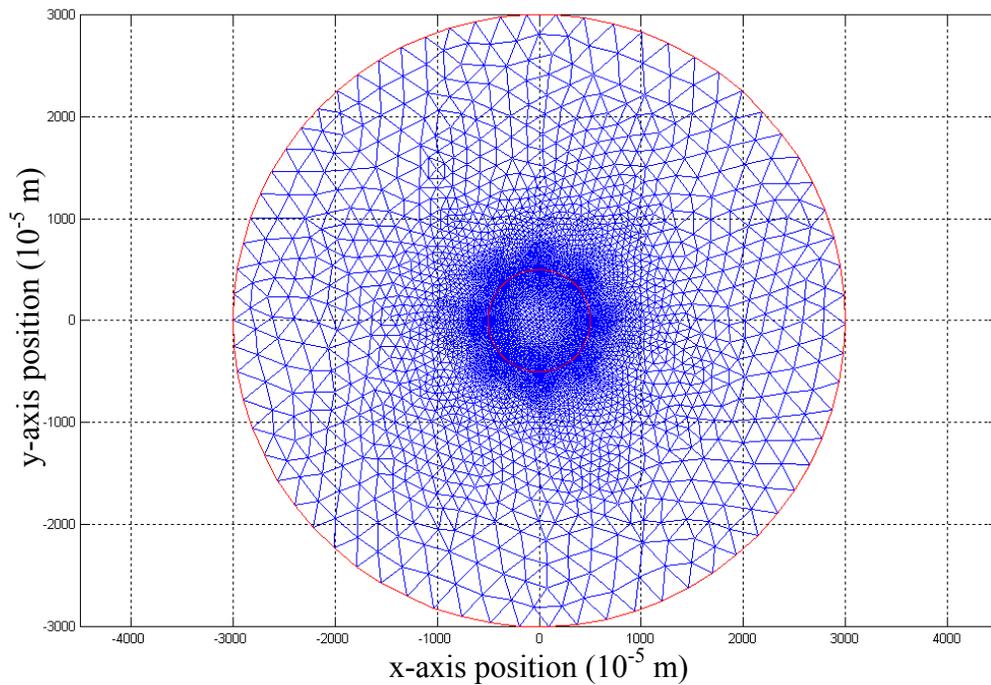
The project included three different approaches to modeling the corrosion magnetic field. The first technique made use of finite element analysis to calculate the magnetic field, but was not well suited to the goals of the project. The second technique, based on the Biot-Savart Law, proved to be the most practical and physically realizable model. The third technique modified the geometry of the currents in the second technique and also applied Biot-Savart Law. The following model descriptions take a detailed look at how each model accomplished the random current assignment, magnetic field calculation, and simulation over time.

#### ***3.3.1 Finite element analysis***

The initial corrosion model used finite element analysis (FEA) to calculate the magnetic field surrounding a corroding steel bar. A random current distribution was generated in a thin outer layer of the bar and then FEA was performed to calculate the magnetic flux density. The following sections detail the model set up, FEA calculations, and the results of the model.

### 3.3.1.1 FEA model explanation

The FEA model has a different geometry from the other two corrosion models. The geometry was defined using MATLAB's partial differential equation toolbox. Figure 3.1 shows the model geometry and the mesh created by MATLAB. In order to simplify the three-dimensional corrosion problem, the model uses a two-dimensional cross section of a corroding bar.



**Figure 3.1: Circular FEA Model and Mesh**

Finite element analysis is a common way to estimate the solution of partial differential equations. To perform FEA, the model must be broken into many small pieces called a mesh [12]. The mesh used in the corrosion model is shown in Figure 3.1. From this mesh, a shape function matrix,  $\bar{S}$ , is calculated which gives a weight to each mesh vertex based on the areas of the elements the vertex touches and its distance to other vertices.

The shape function matrix accumulates as each element of the mesh adds its contribution to its three vertices [12].

The current density matrix is generated from a defined current density given for the model. For any element where current is greater than zero, the current density is divided equally among each of the three vertices. The current density contributed by each element is accumulated in the current density matrix  $\bar{J}$ .

Using  $\bar{S}$  and  $\bar{J}$ , the vector potential matrix  $\bar{A}$  may now be calculated according to Ampere's law,

$$\bar{\nabla} \times \bar{H} = \bar{J} \quad (3.1)$$

where  $\bar{\nabla}$  is curl and  $\bar{H}$  is the magnetic field vector. Because the problem has only two dimensions,  $\bar{J}$  can be assumed to flow only in the  $z$ -direction, but depends only on  $x$  and  $y$ . This means that  $\bar{H}$  is defined in the  $x$  and  $y$  directions only.

The magnetic vector potential,  $\bar{A}$ , indicates the voltage inducing capability of the magnetic field and is defined by

$$\nabla \times \bar{A} = \bar{B} \quad (3.2)$$

where  $\bar{B}$  is the magnetic flux density. With some substitution, Ampere's law now becomes

$$-\nabla \cdot \frac{1}{\mu} \nabla \times \bar{A} = \bar{J} \quad (3.3)$$

where  $\mu$  is the effective magnetic permeability.

The shape function matrix is designed such that

$$\overline{S} \overline{A} = \overline{J}, \quad (3.4)$$

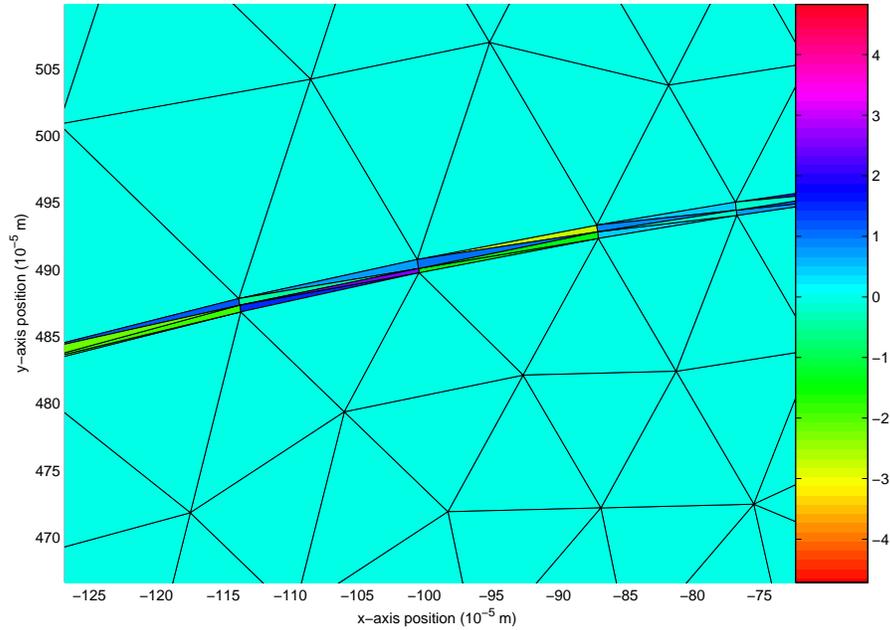
which can now be solved for  $\overline{A}$ . The  $\overline{A}$  matrix is then used to solve for  $\overline{B}$  which gives the magnetic flux density for each element in the mesh [12].

### 3.3.1.2 FEA model implementation

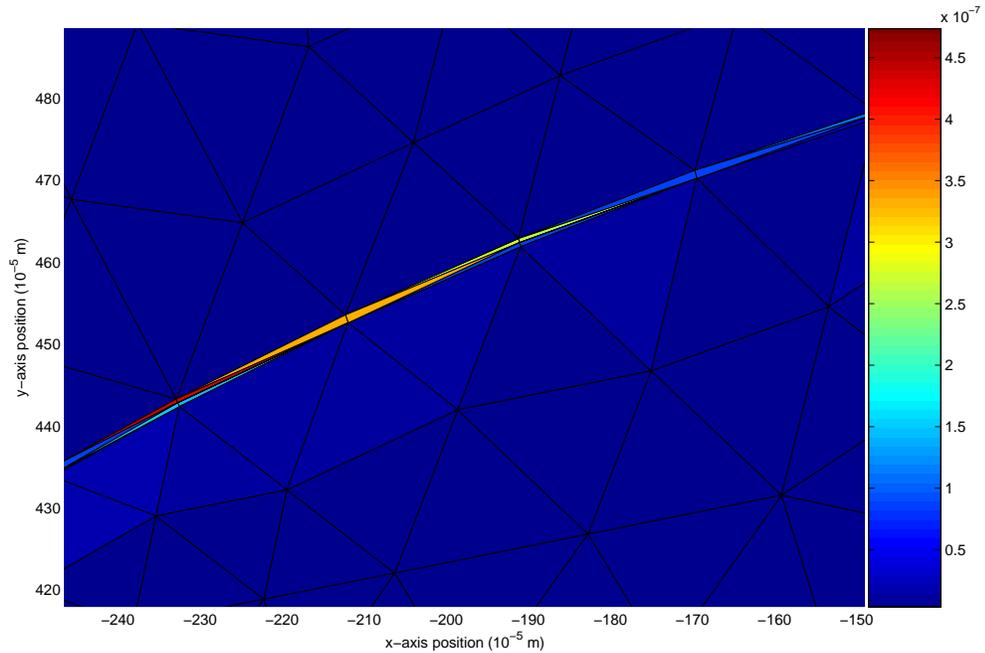
Applying the FEA technique to corrosion began with generating a current distribution. The steel bar cross section splits into two regions: one represents the interior of the bar where little corrosion activity takes place, and the other represents the thin outer layer of the bar that corrodes. Corrosion can only occur where the steel is in contact with the environment. The exterior layer is modeled as 10  $\mu\text{m}$  thick. MATLAB meshes these two regions as well as a large bounding region, as seen in Figure 3.1. Using the mesh, the model assigns each element in the thin corroding layer a random current density based on a normal distribution centered at an average value for rapid corrosion. The current density values are used to generate the  $\overline{J}$  matrix.

The shape function matrix is only calculated one time, because the geometry of the model does not change between time steps. The current density matrix, however, is recalculated at each time step when the current distribution is regenerated. When the currents regenerate, the normal distribution also re-centers around the value from the previous time-step. Figure 3.2 illustrates the thin corroding layer with current density, and Figure 3.3 shows the magnetic field calculated in one time step. Figure 3.4 shows the magnetic

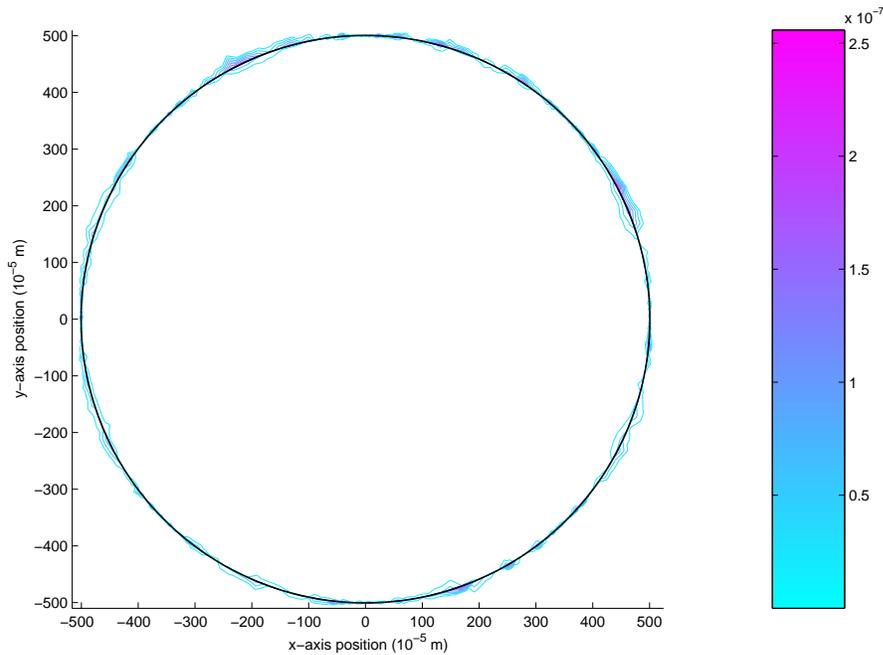
field surrounding the bar as a contour plot, making it easy to see the variation and irregularity.



**Figure 3.2: FEA Model Current Density Output in A/m<sup>2</sup>**



**Figure 3.3: FEA Model Magnetic Field Output in Am<sup>-1</sup>**



**Figure 3.4: FEA Model Magnetic Field Contour Plot in  $\text{Am}^{-1}$**

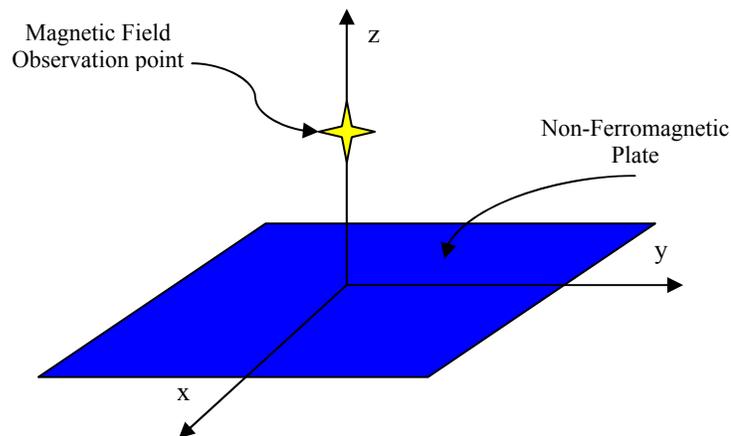
Despite the success of the FEA model in calculating the magnetic field, the modeling technique was abandoned because it was difficult to construct a physical experiment that replicated the simulation. In reality, the corrosion currents flow on the surface of the steel, not only in the  $z$ -direction as assumed in the model. The FEA technique also used extra computation time calculating the magnetic field at every point in the model geometry. The GMR sensors can only monitor one point in space at a time, which makes calculating this extra information add little value. The models based on the Biot-Savart law have a simple physical interpretation and only calculate the magnetic field at a single point, which saves computational effort.

### 3.3.2 Nonferromagnetic plate model using Biot-Savart law

The nonferromagnetic plate model developed from a need to simplify the geometry and create a model that executes faster than the FEA model. The FEA model determined magnetic field for a slice of a cylindrical piece of iron, while this model calculates the magnetic field for currents in a corroding metal sheet. The problem was further simplified by assuming that the corroding material was nonferromagnetic and therefore has the same permeability as the air surrounding it. This assumption allowed the use of the Biot-Savart law directly without modification for highly permeable materials.

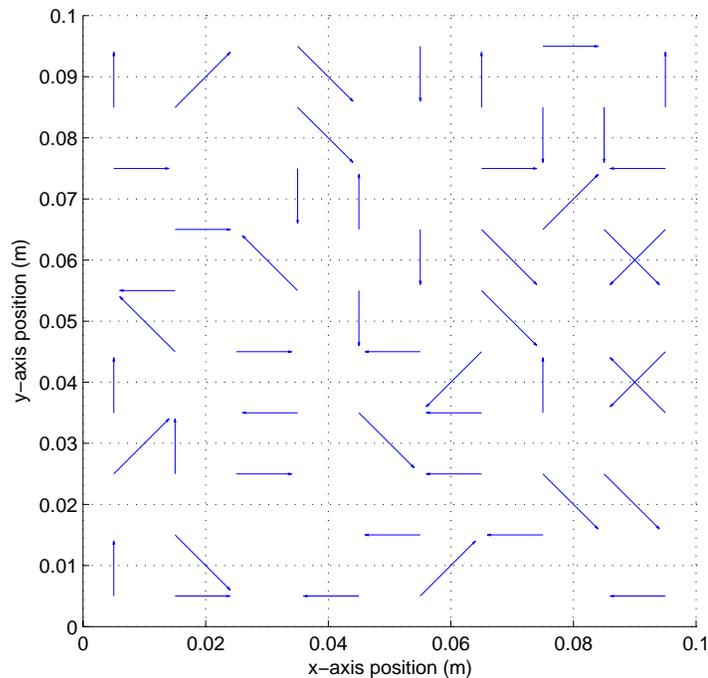
#### 3.3.2.1 Plate model explanation

The plate model has a simple geometry that is easily replicated in experiment. The currents flow in a plate defined in the  $xy$ -plane and the observation point is placed directly above that plane. The location of the observation point may be changed to reflect the location of the sensor in an experiment. Figure 3.5 illustrates the model's geometry.



**Figure 3.5: Plate Model Geometry**

A grid divides the nonferromagnetic plate into sections which are randomly selected as either an anode or a cathode site. The simulation randomly selects 20% of the sections to be anodes. For every anode a neighboring section is chosen as its cathode. Each section may be either an anode or a cathode, but not both. Cathode sections may act as a cathode for multiple anodes, but an anode section has only one cathode. Figure 3.6 shows the plate divided into 100 sections with currents assigned to the anodes and cathodes. The spacing of the grid that divides the plate into sections affects the magnetic field calculation. The grid spacing shown in Figure 3.6 is too large for effective field calculation, but the larger size allows the image to easily show the concept.



**Figure 3.6: Sample Current Distribution**

For each anode-cathode pair, a normally distributed random current magnitude is assigned. The normal distribution has a median that is equivalent to  $500 \mu\text{A}/\text{cm}^2$ , which is a typical current density for rapid corrosion. The normal distribution causes the current

to have minor fluctuations but maintain this average current density on the plate. The simulation then applies the Biot-Savart law for each current to obtain the magnetic field magnitude at the observation point.

The Biot-Savart law determines the magnetic field at a point in space due to a filamentary current. The differential form is given by

$$\overline{dH}_2 = \frac{I_1 \overline{d\ell}_1 \times \hat{a}_{R_{12}}}{4\pi R_{12}^2} \quad (3.5)$$

where  $\overline{dH}_2$  is the magnetic field due to the filamentary current of magnitude  $I_1$  and differential length  $\overline{d\ell}_1$ . The length between the current,  $R_{12}^2$ , and its unit vector  $\hat{a}_{R_{12}}$  are illustrated in Figure 3.7, where a graphical representation of the basis to find  $\overline{dH}_2$  due to a finite current segment is shown [13-14].

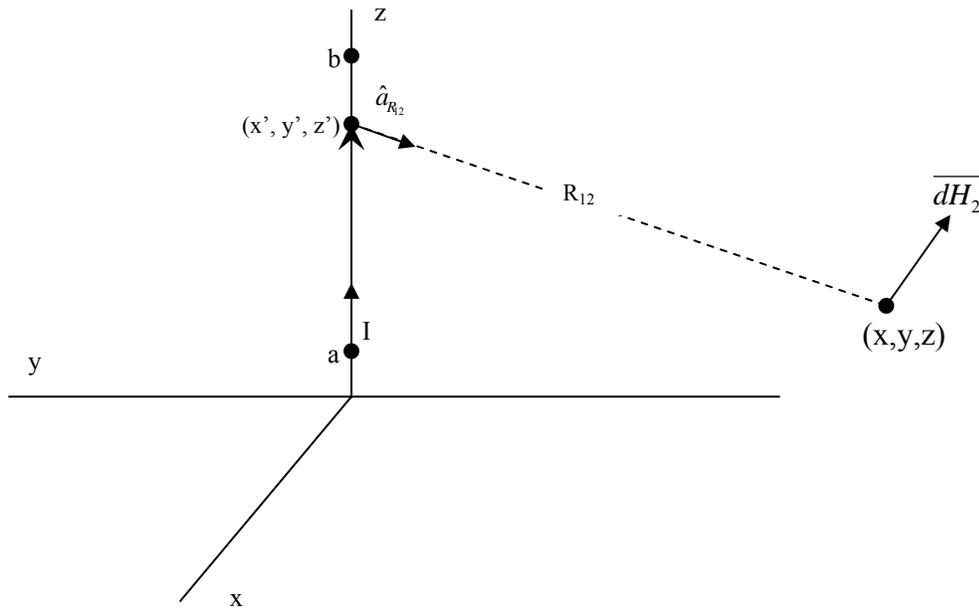


Figure 3.7: Biot-Savart Law for Finite Current Segment [13]

When the differential form is integrated from point a to point b as shown in Figure 3.7, the law becomes

$$\overline{H}(x, y, z) = \int_a^b \frac{I \overline{d\ell} \times \hat{a}_R(x', y', z', x, y, z)}{4\pi R^2(x', y', z', x, y, z)}. \quad (3.6)$$

In this equation,  $x'$ ,  $y'$ , and  $z'$  represent the coordinates of the differential current length, and  $x$ ,  $y$ , and  $z$  represent the coordinates of the observation point. This integral form is used to calculate the magnetic field at the observation point for each of the current segments placed on the nonferromagnetic plate [13].

### 3.3.2.2 Implementation and testing

The simulation software, implemented in MATLAB, first defines the geometry of the model including the spacing of the grid and the size of the plate. The program then randomly selects 20% of the grid sections as anode regions. The program finds a suitable cathode for each anode, insuring that the selected cathode is not also an anode region. A random current magnitude is assigned to the anode-cathode pair using a normal distribution. For each current, the simulation then calculates the magnetic field contribution at the observation point by numerically integrating the Biot-Savart law. Once all contributions are known, the magnetic field from each current is summed as a vector to calculate the total magnetic field. The simulation repeats all of these steps for each time step, except model geometry definition. The code that implements this simulation is located in Appendix B.

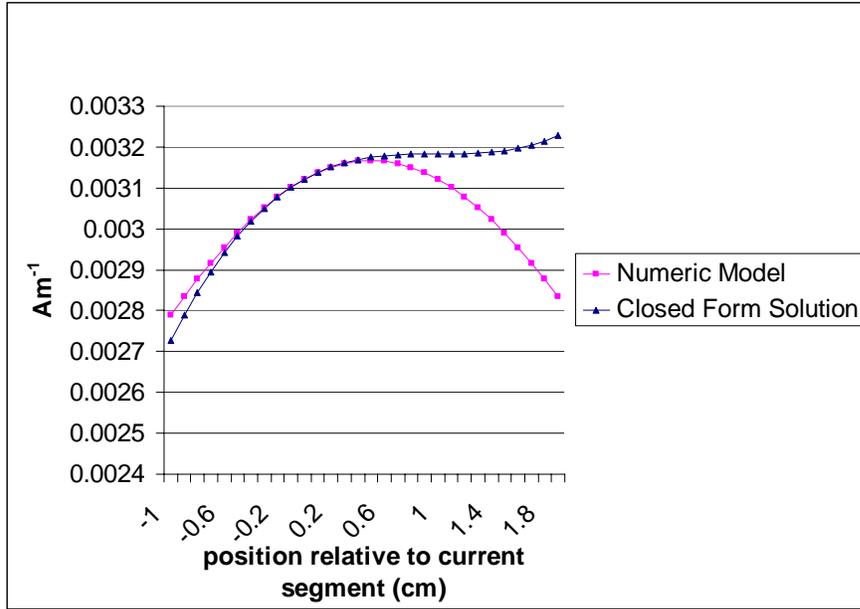
The model's accuracy was checked and tuned with several tests. The original simulation results were compared with a closed form solution of the Biot-Savart law. For a finite current segment, the magnetic field at an observation point may be calculated using the closed form expression

$$\vec{H} = \frac{\hat{\phi}I}{4\pi R} \left\{ \frac{b-z}{\left[R^2 + (b-z)^2\right]^{\frac{1}{2}}} - \frac{a-z}{\left[R^2 + (a-z)^2\right]^{\frac{1}{2}}} \right\}. \quad (3.7)$$

The variables here come from Figure 3.7, with the exception of  $\hat{\phi}$  which is the unit vector in the  $\phi$  direction of a cylindrical coordinates system.

A second simulation was developed using this form of Biot-Savart to compare with the plate model simulation. The closed form gives a valid solution only when the observation point falls between the points a and b and on the same plane as the current segment. In both simulations, the observation point was slowly moved from slightly beyond point a to slightly beyond point b while remaining in the same yz-plane as the current segment. The graph in Figure 3.8 summarizes the results.

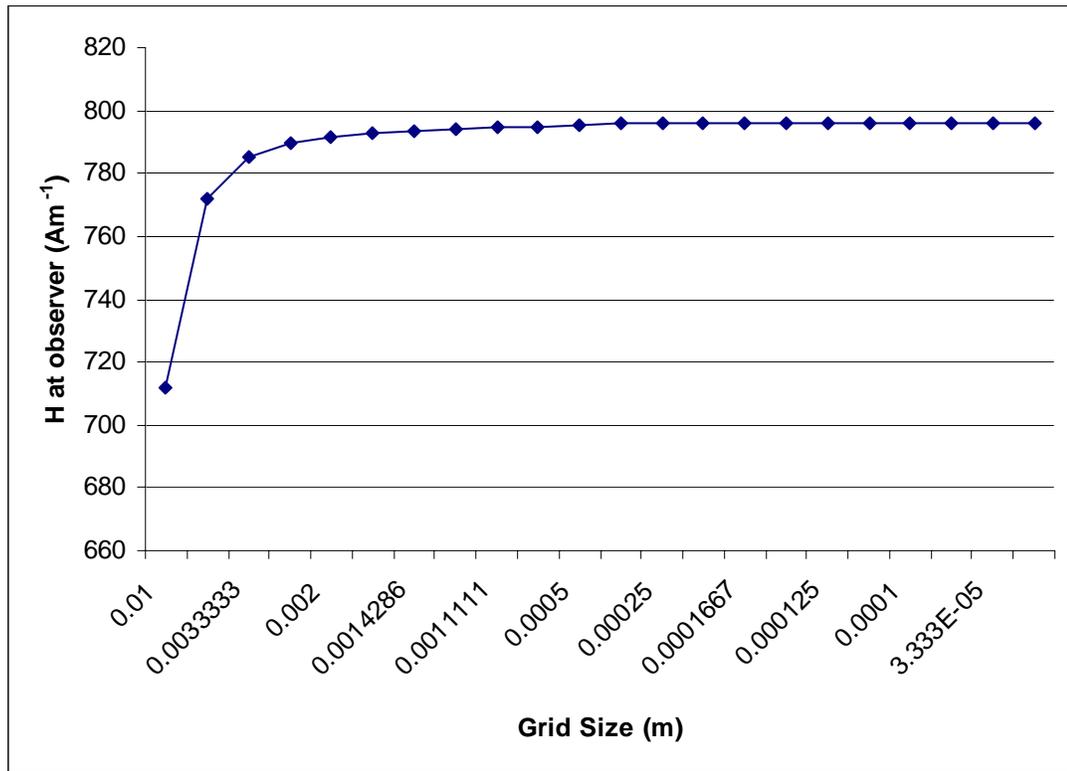
The graph indicates that the numeric integration in the simulation achieves nearly the same results as the closed form solution. However, beyond the ends of the current segment, the simulation produces more accurate results than the closed form solution. The closed form solution makes simplifying assumptions to arrive at a simple expression which cause the results of the solution to be invalid beyond the ends of the current segment.



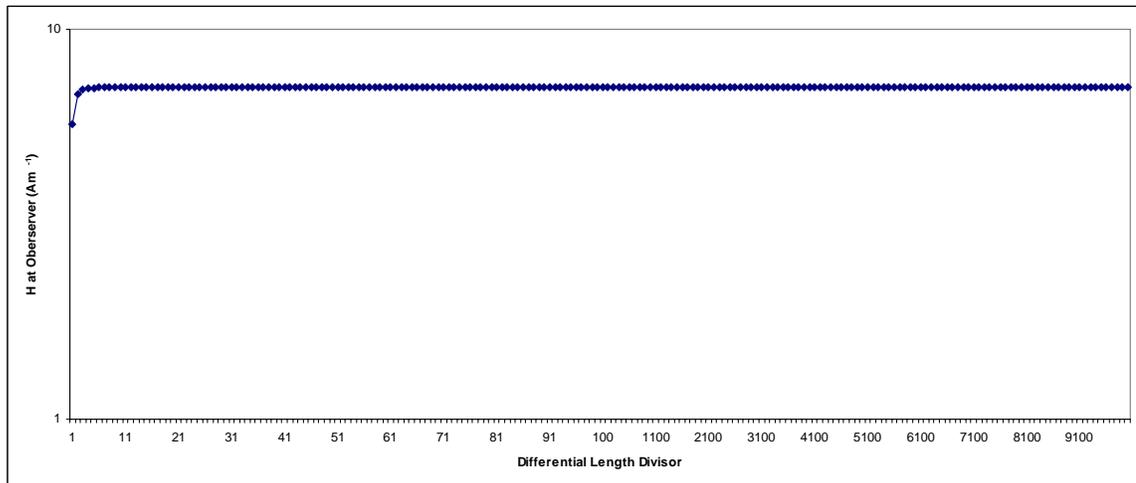
**Figure 3.8: Plate Model versus Closed Form Solution**

Two further tests were conducted to determine the appropriate parameters for the size of the grid and the differential current path length. The first test held the locations of the current segment and observer constant while scaling the current magnitude and varying the size of the grid. Figure 3.9 shows the results. As the grid size decreases, the calculated magnetic field at the observer stabilizes. As a result of this test, the simulations used in the analysis use a grid size of 0.002, which makes a 50 x 50 grid for the 10-cm square piece of aluminum.

The final test effectively varied the differential length used for the numeric integration while holding all other variables constant. The test changed differential length by gradually increasing the factor used to divide the total length of the current segment. Figure 3.10 shows the results of the test. The differential length can be relatively large and still maintain computational accuracy. The simulations performed for comparison with experimental results used a differential length scalar of 20.



**Figure 3.9: Magnetic Field versus Grid Size**



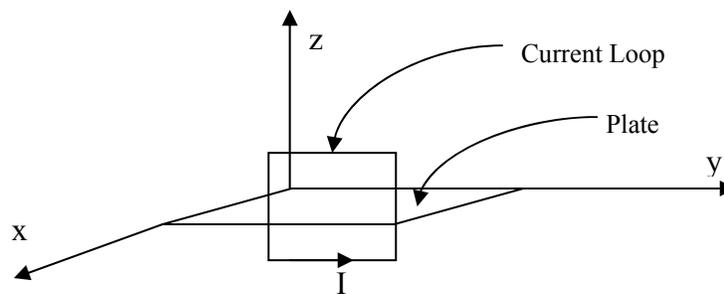
**Figure 3.10: Magnetic Field versus Differential Current Length**

### 3.3.3 Current loop model

The current loop model shares many characteristics with the plate model. Similar assumptions lead to a simple geometry and the use of aluminum removes the need to simulate different permeability. An experiment developed for comparison to the current loop model could also be compared to the plate model. The models differ only in how the current distribution is defined.

#### 3.3.3.1 Current loop model explanation

The current loop model geometry is the same as the plate model. A nonferromagnetic plate is placed in the  $xy$ -plane and an observation point is located above the plate with  $z$ -axis displacement. The difference between the models arises from the definition of the currents flowing in the plate. In the plate model, the currents all flow on the surface from an anode to a cathode. In the current loop model, sections of the grid contain small rectangular current loops instead of current filaments. These current loops are normal to the aluminum plate and have a magnitude randomly determined based on a normal distribution as in the plate model. Figure 3.11 illustrates how a single element in the plate and its current are arranged.



**Figure 3.11: Current Loop in Plate Element**

### 3.3.3.2 Implementation

For each time step, the simulation selects 20% of the plate elements to contain a current loop as in the plate model. The loop is 90% of the length of the plate element in the  $y$ -direction, and  $2\ \mu\text{m}$  thick in the  $z$ -direction. The length and width were chosen so that the length of the current is similar to the plate model, and the  $z$ -direction thickness was chosen to reflect the depth of corrosion in steel. The magnetic field at the observation point is obtained by first rotating the coordinate system  $90^\circ$  around the  $y$ -axis and then applying the solution technique outlined in the paper by Misakian [15]. The solution technique uses the locations of the corners of the rectangular loop to solve for the magnetic flux density at a give observation point. Figure 3.12 shows the basic setup of the rectangular loop problem as presented in the paper.

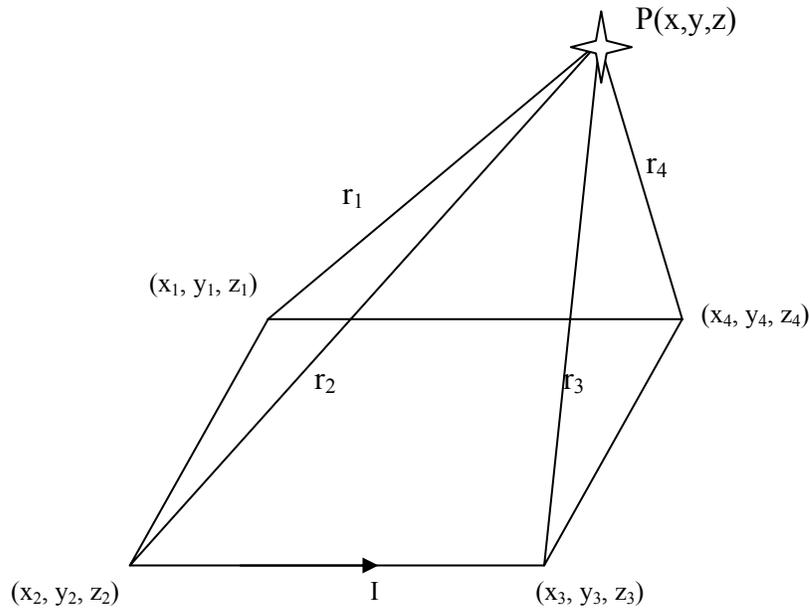


Figure 3.12: Rectangular Loop Problem [15]

The components of the magnetic flux  $\bar{B}$  at point P can then be solved using

$$B_z = \frac{\mu_o I_1}{4\pi} \sum_{\alpha=1}^4 \left[ \frac{(-1)^\alpha (y - y_\alpha)}{r_\alpha [r_\alpha + (-1)^{\alpha+1} (x - x_\alpha)]} - \frac{x - x_\alpha}{r_\alpha (r_\alpha + (y - y_\alpha))} \right], \quad (3.8)$$

$$B_x = \frac{\mu_o I_1}{4\pi} \sum_{\alpha=1}^4 \left[ \frac{(-1)^\alpha (z - z_\alpha)}{r_\alpha [r_\alpha + (y - y_\alpha)]} \right], \quad (3.9)$$

and

$$B_y = \frac{\mu_o I_1}{4\pi} \sum_{\alpha=1}^4 \left[ \frac{(-1)^\alpha (z - z_\alpha)}{r_\alpha [r_\alpha + (-1)^{\alpha+1} (x - x_\alpha)]} \right]. \quad (3.10)$$

In these expressions,  $\mu_o$  is the permeability of free space,  $I_1$  is the current in the loop, and  $r_\alpha$  is the distance from a corner to the observation point, as seen in Figure 3.12. The symbols  $x_\alpha$ ,  $y_\alpha$ , and  $z_\alpha$  denote the coordinates of the corners of the current loop, while  $x$ ,  $y$ , and  $z$  denote the coordinates of the observation point.

The axis rotation places the current loop in the  $xy$ -plane which allows the application of the rectangular loops solution. The paper [15] is not presented for a general case, but with axis rotation, the results can be suitably adapted. The equations developed by Misakian are applied to each loop on the plate and the magnetic field at the observation point is the sum of the individual results. The code that implements this simulation is located in Appendix C.

## **4. RESULTS**

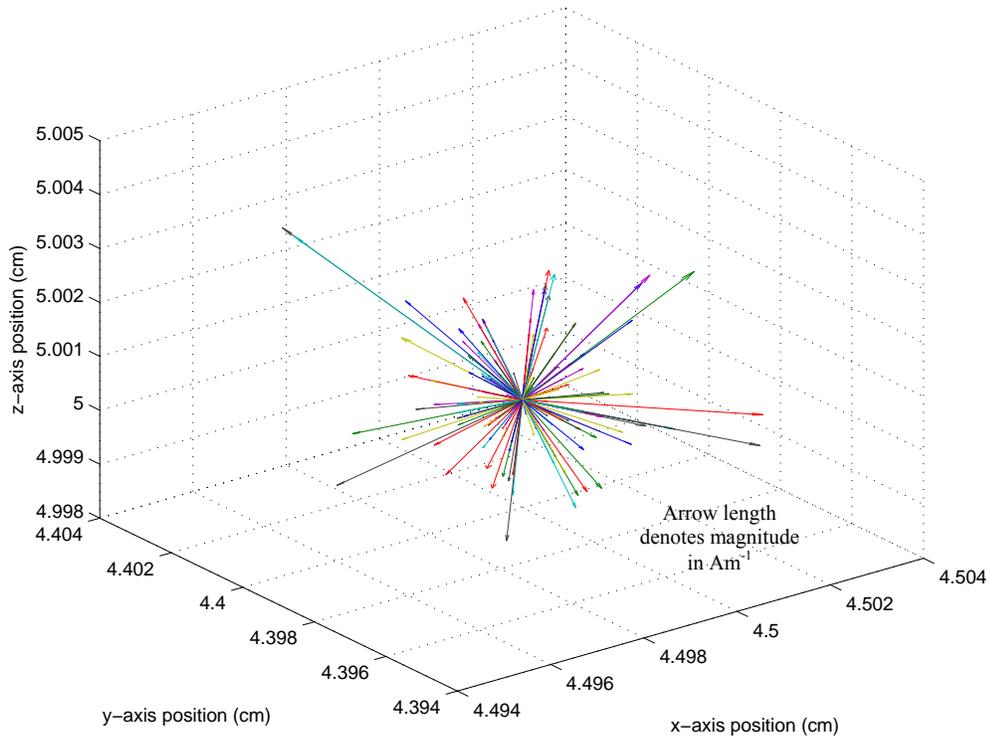
The simulations were primarily performed using the plate model because it has the most intuitive physical interpretation. The simulations from the plate model produce results which resemble the magnetic field that was generally anticipated. The following sections present the results of simulations conducted with the plate model and the current loop model. Postprocessed results, aimed at finding a trend in the magnetic field, are also discussed.

### **4.1 Simulation Results**

Various simulations were conducted with both the plate and current loop models. The simulations ranged in length from 200 time points up to 100,000 time points. As mentioned before, at each time point, the current distribution is regenerated and the magnetic field at the observation point is recalculated. Each current on the plate contributes to the magnetic field at the observation point. Figure 4.1 shows a sample of what the various magnetic field contributions look like at the observation point using the plate model. The individual components clearly point in random directions and have random magnitude as would be anticipated from the currents placed on the plate. In the figure, the observation point is located roughly in the middle of the plate and one centimeter from the surface.

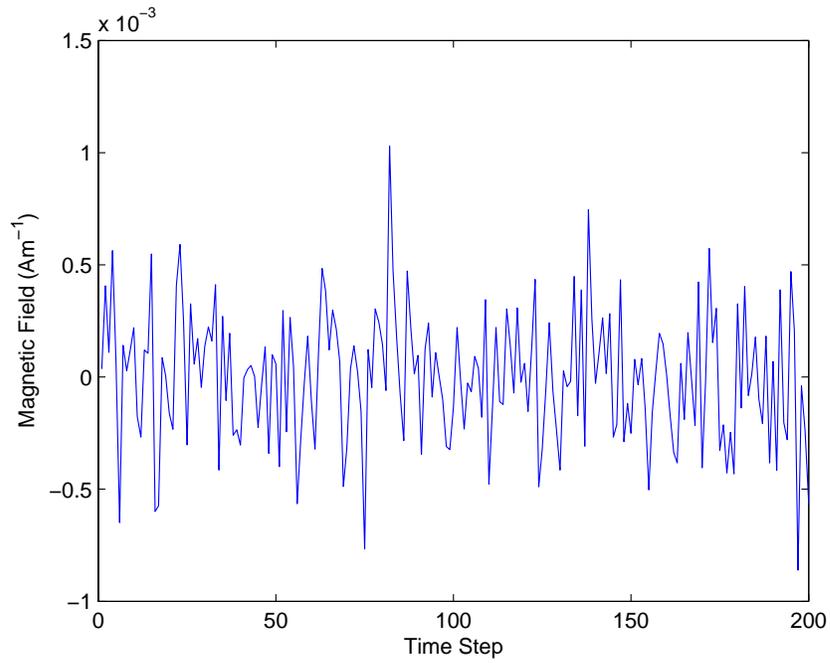
Once these components are calculated, they are summed as vectors and the total magnetic field for the time step is recorded. This is repeated for multiple time steps to produce the

magnetic field in the simulation time. The simulation time is relative and indicates only the number of time steps included in the simulation.

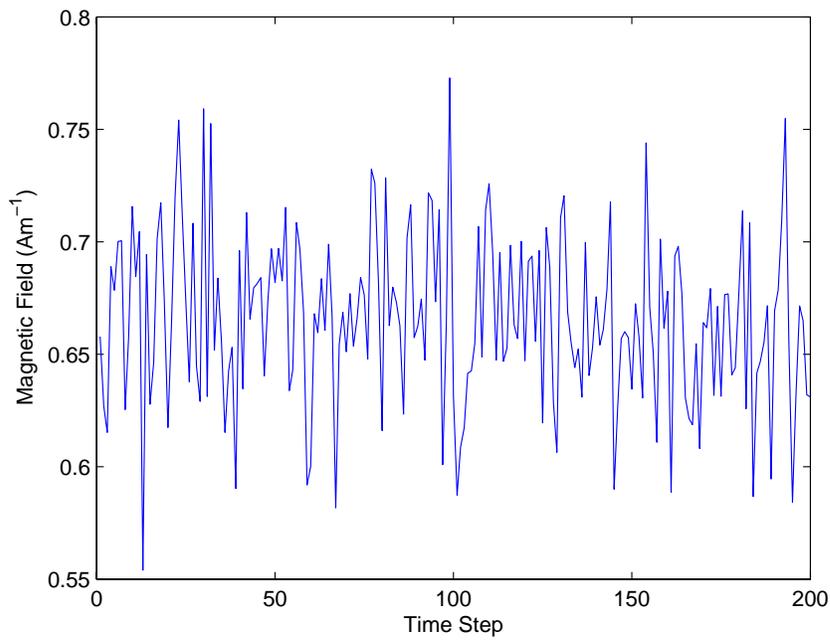


**Figure 4.1: Magnetic Field Contributions from a Plate Model Current Distribution**

The actual time between each time step is scaled in postprocessing. Figure 4.2 shows the normal, or  $z$ , component for a 200 time step simulation using the plate model. The GMR sensors detect principally the normal component of the magnetic field, and so analysis is performed using that component. Figure 4.3 shows a similar simulation using the current loop model. The primary differences between the two sets of results are the offset in the normal component and the increased amplitude seen in the current loop model simulation. The offset and amplitude increase occur because all the currents travel in the  $y$ -direction in the current loop model.



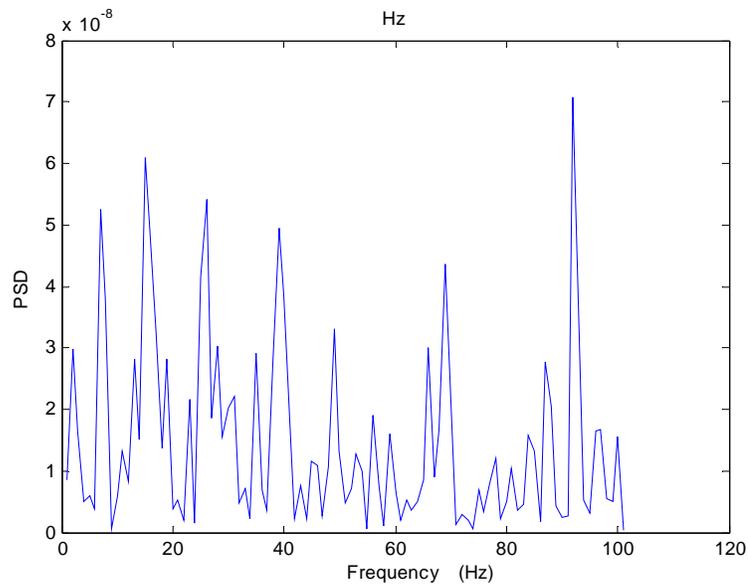
**Figure 4.2: 200-Point Simulation Data using the Plate Model**



**Figure 4.3: 200-Point Simulation Data using the Loop Model**

## 4.2 Postprocessing Analysis

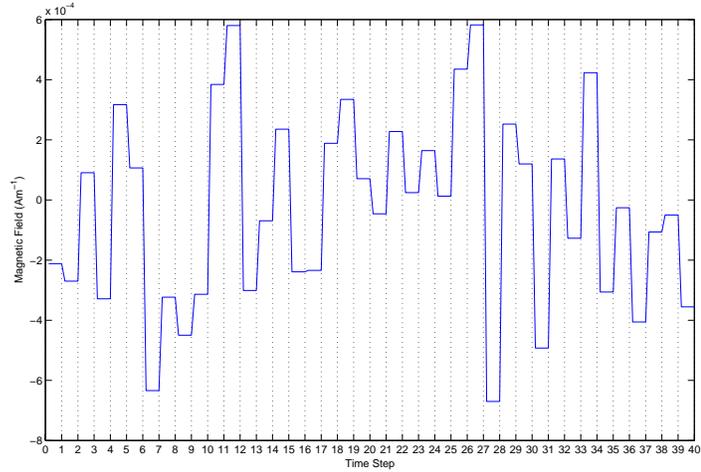
The simulation signals were first analyzed using fast Fourier transform. More specifically, the signals were transformed to power spectral density (PSD) plots which allow the power in each frequency to be seen. The PSDs produced with the simulation data used the MATLAB “pwelch” function which implements Welch’s algorithm. Welch’s algorithm divides the signal data into overlapping window regions using user input parameters that determine the window shape and the amount of overlapping between the windows. If the signal cannot be divided evenly, it is truncated so that it may divide evenly into the specified segments. Each segment then undergoes a fast Fourier transform with a user-specified number of points. The FFTs are converted to a set of periodograms which estimate the power spectral density of the signal. The set is then averaged to give the PSD of the signal and scaled to the user-provided sampling frequency [16-18]. Figure 4.4 shows the PSD of the 200-point simulation presented in Figure 4.2.



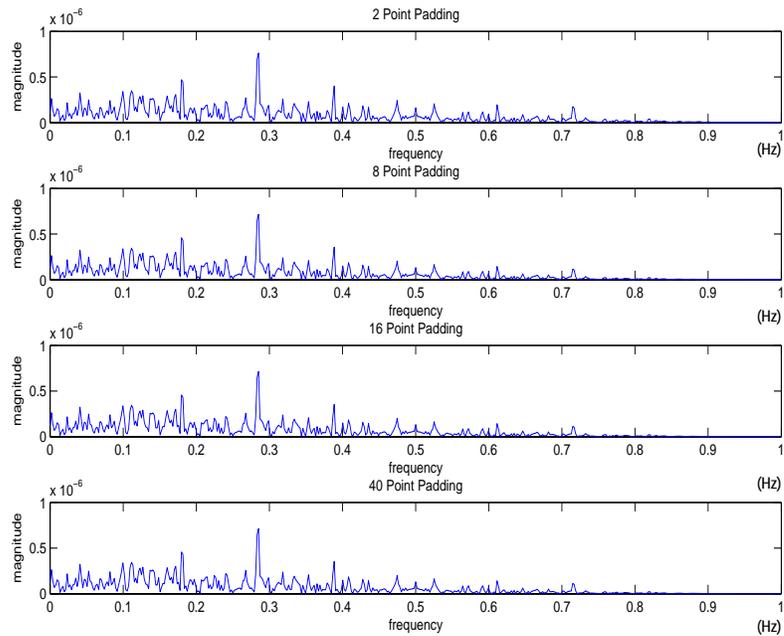
**Figure 4.4: 200-Point Simulation PSD using the Plate Model**

Because the time steps in the simulation are relative, the data is padded to simulate oversampling. Essentially, between each calculated time step, points of the same value are inserted into the signal. This signal padding is also called sample and hold interpolation. This padding does not affect the low frequency information, but does add some high frequency content due to the square corners of the waveform. It also artificially expands the bandwidth of the PSD. Figure 4.5 illustrates what the sample and hold interpolation does to the time domain signal. This particular signal is padded with five replicated values between each simulated data point. The signal exhibits flattened areas due to the padding points. The padding is added to reflect the effects of sampling at a higher rate than the signal changes. In reality, the GMR sensors will sample the magnetic field much faster than it varies, so repeated values are expected in the data.

Figure 4.6 shows the same 1024 time step signal padded differently while the PSD remains the same. The frequency axes in these plots are adjusted so that the bandwidth they have in common is displayed. Without this adjustment, the frequency axis extends to one half the sampling frequency according to the Nyquist rate.



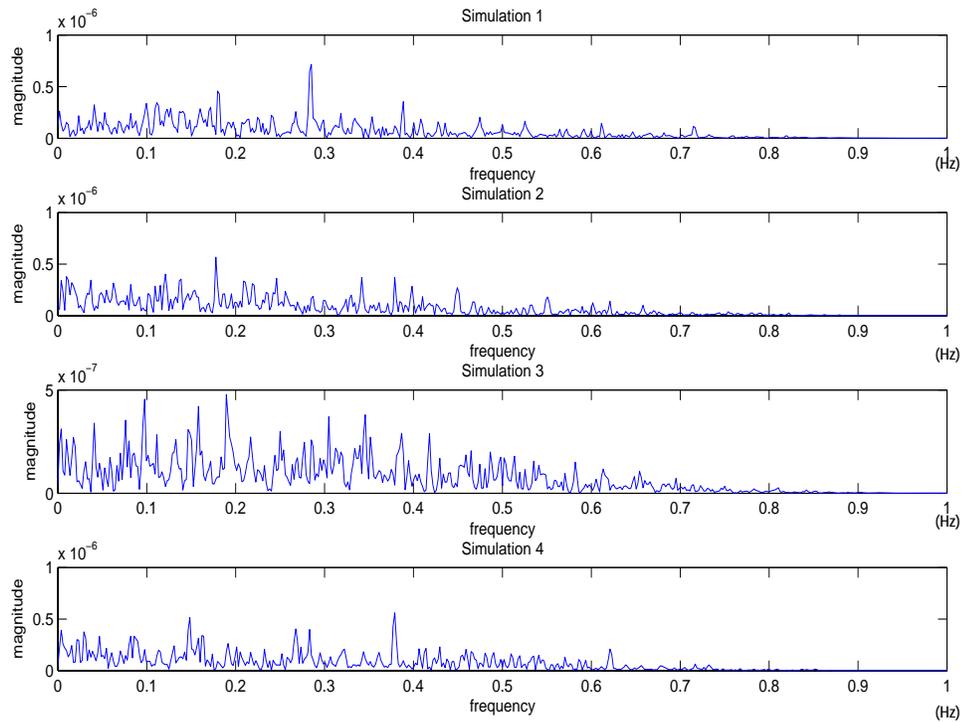
**Figure 4.5: Five Point Padded Signal using the Plate Model**



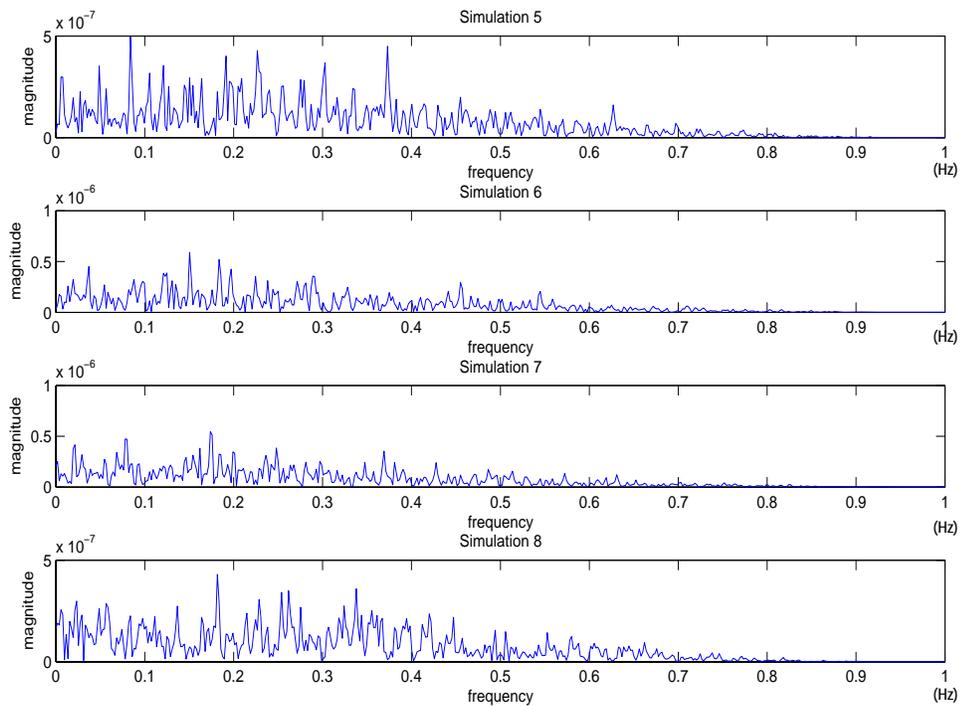
**Figure 4.6: PSDs of Padded Signals from the Plate Model**

Figures 4.7 and 4.8 contain four simulation PSDs each. These four independent simulations are each 1024 time steps long and are padded with 10 points between each time step value. This gives the signals a sampling rate of 10 Hz and the PSD a bandwidth of 5 Hz. The PSDs are truncated in the figures to show detail between 0 and 1 Hz. These

PSDs reveal that there is very little correspondence between the simulations. It was originally thought that a spike would be visible at the frequency at which the currents were changed in the simulation. However, upon closer inspection, this frequency artifact is not visible because the magnetic field generated in the simulation uses normally distributed noise.

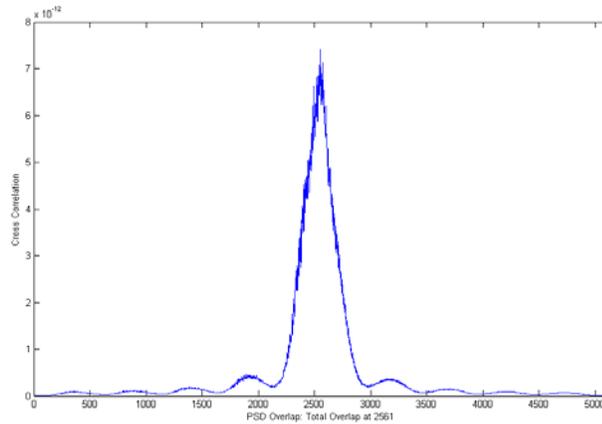


**Figure 4.7: 1024-Point Simulation 1 through 4 PSDs using the Plate Model**



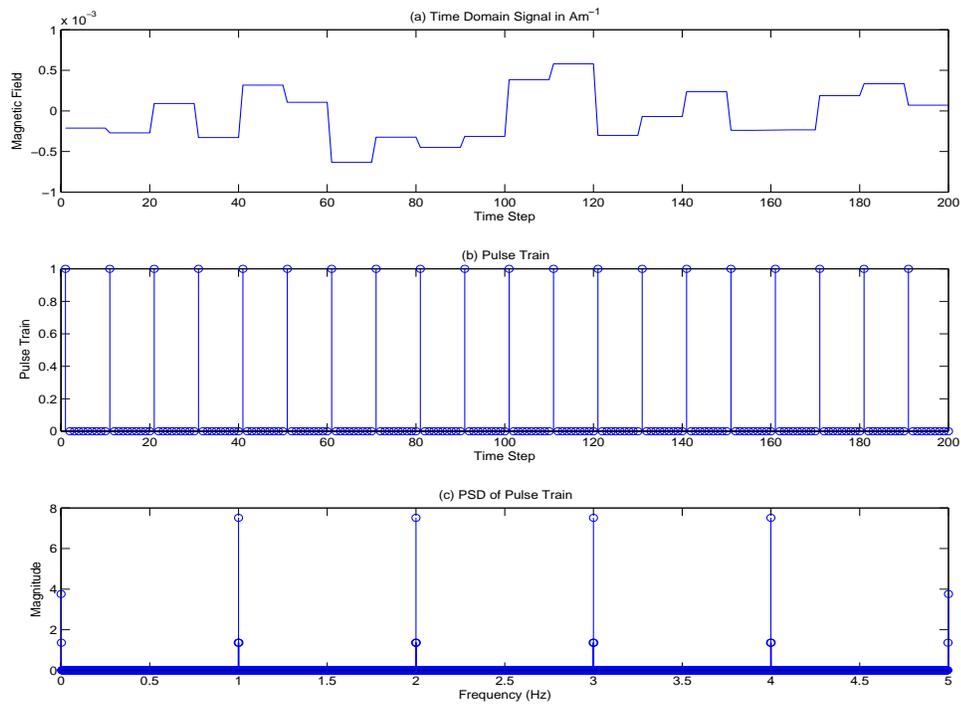
**Figure 4.8: 1024-Point Simulation 5 through 8 PSDs using the Plate Model**

Correlating the PSDs reveals that the results are similar, but bear little in common with one another, other than a general low pass shape. Figure 4.9 shows the correlation between the PSD of two 1024 time step simulations with 10-point padding. The correlation plot is generated by gradually sliding the two PSDs past each other. As the PSDs slide, the overlap between them increases until they reach the maximum overlap in the middle of the plot. Past this midpoint, the overlap decreases as the slide continues. At each position, the similarity between the two PSDs is calculated and plotted to form Figure 4.9. The correlation plot takes the shape of a sinc function due to high frequency FFT effects in the PSD. The increased correlation as the PSDs totally overlap occurs because the PSD have a similar shape. However, the correlation results indicate that the two simulations have no real common characteristics [19].



**Figure 4.9: PSD Correlation**

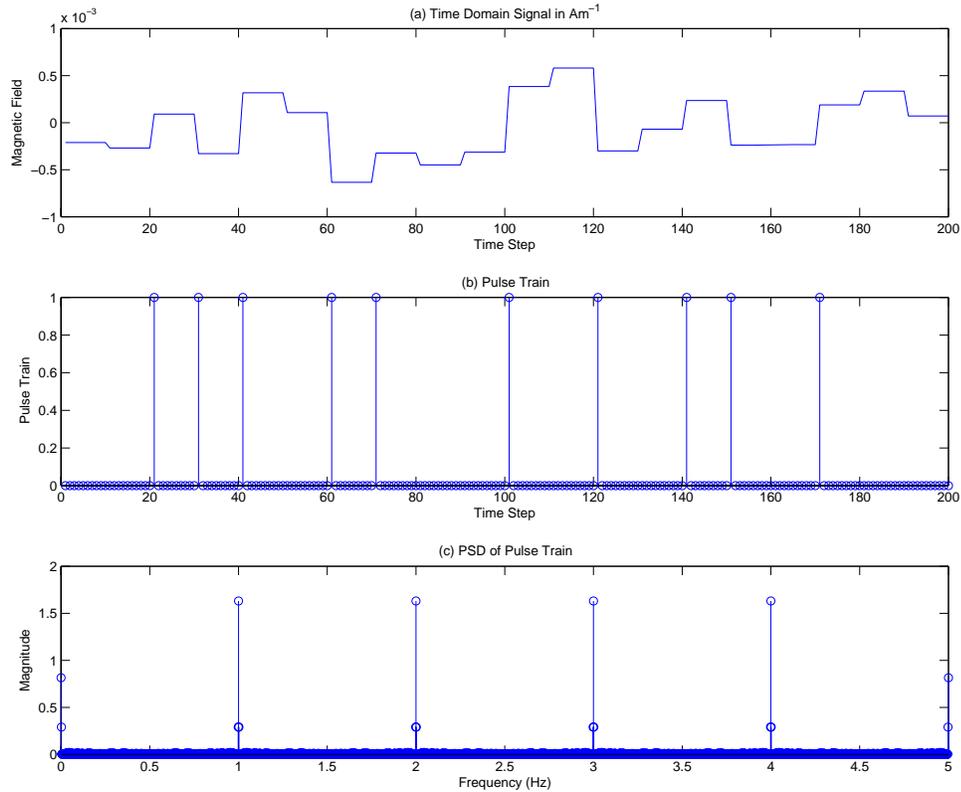
Another possible approach to extracting meaningful information from magnetic field measurements involves detecting only the change in magnetic field. If the sensors were to create a pulse when they recorded a change in magnetic field based on some threshold value, the generated pulse train would likely have clear frequency characteristics. To illustrate this process, the data from a 1024 time step simulation and 10 point padding was converted to a pulse train of ones and zeros. A pulse with a height of one was generated each time the magnetic field changed, which leads to a pulse train at 1 Hz in this example. The original signal and the associated pulse train are displayed in Figure 4.10(a) and Figure 4.10(b) respectively. The time scale is zoomed in so that detail is visible. The PSD of this signal in Figure 4.10(c) shows a spike at 1 Hz and also at every harmonic. These unwanted harmonics can be filtered from the desired information.



**Figure 4.10 Pulse Train PSD from a 1024-Point Plate Model Simulation**

However, this example is somewhat unrealistic. In practice the sensors will act based on a threshold value for magnetic field change. To better examine this technique, a threshold was placed at the standard deviation of the magnetic field data. When the change between two samples exceeded this value, the program generated a pulse, just as the sensors will do. The standard deviation was chosen as the threshold so that a pulse would be generated at every significant change in the magnetic field. The results of this analysis are presented in Figure 4.11. The original time domain waveform in Figure 4.11(a) is the same as the previous example. Figure 4.11(b) shows the new pulse train based on a threshold value. The new PSD of the pulse train is in Figure 4.11(c). The frequency characteristics of the two pulse trains are nearly identical, which indicates that using a threshold value is a viable means of extracting meaningful information from the

magnetic field data. The challenge in implementing this method comes from the selection of the threshold value. More investigation will be needed to determine what threshold value is viable under various sensing circumstances.



**Figure 4.11: Pulse Train Generated with Standard Deviation Threshold Value**

Detecting corrosion by recording change in the magnetic field may have more potential than simply listening to magnetic field noise. As corrosion begins, progresses, and ends, the characteristics of the induced currents change [7]. At the onset of corrosion, many fast transients are observed, but these transients gradually fade and are replaced by DC trends [7]. These transients could be the key to detecting corrosion in its early stages when the current changes rapidly as the passivating layer breaks down. Later, slower transient behavior could be used to indicate corrosion already in progress.

### **4.3 Further Analysis**

The analysis presented here needs to be expanded. The use of advanced signal processing techniques, such as wavelet analysis, may be able to reveal more trends and commonalities between the magnetic field simulations. The data would also benefit from comparison to experimental data. This experiment would involve measuring a corroding aluminum plate with the GMR sensors and looking for correspondence between the simulated and collected data. The difficulty in extracting meaningful trends from the data arises from its generation as noise. If the currents are determined by a more advanced simulation that takes into account the concentration of corroding agents, the results may have a more noticeable trend.

## **5. CONCLUSIONS**

The models developed provide a method of estimating the magnetic field generated by corrosion currents. The behavior of the magnetic field will be instrumental in determining how to measure and detect the presence of corrosion using GMR magnetic sensors. The simulations data did not directly yield information on the frequency at which corrosion currents change, nor did the analysis extract interesting information. However, the simulations did lead to a possible sensing technique for the magnetometers. The GMR sensors can be used to detect changes in the magnetic field, rather than only listening to its magnitude. Applying the GMR sensors in this manner may allow the system to assess more effectively the state of corrosion within a reinforced structure. That capability makes the GMR system a powerful tool in protecting infrastructure and ensuring safety.

## REFERENCES

- [1] J. Popovics, G. Gallo, M. Shelton, and P. Chapman, "A magnetic sensing approach to characterize corrosion in reinforced concrete," in *Proceedings of SPIE – The International Society for Optical Engineering*, 2007, p. 65291A.
- [2] Y. Ma, J. Wikswo, M. Samuleviciene, K. Leinartas, and E. Juzeliunas, "Magnetic fields induced by electrochemical reactions: Aluminum alloy corrosion sensing by SQUID magnetometry on a macroscopic scale," *Journal of Physical Chemistry B*, vol. 106, pp. 12549-12555, December 2002.
- [3] J. Popovics, G. Gallo, M. Johnson, and P. Chapman, "Passive and active corrosion sensing for metals using magnetic sensors," in *Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical and Aerospace Systems 2007 Conference*, 2008.
- [4] S. Kranc and A. Sagüés, "Detailed modeling of corrosion macrocells on steel reinforcing in concrete," *Corrosion Science*, vol. 43, pp. 1355-1372, July 2001.
- [5] P. Ghods, O. Isgor, and M. Pour-Ghaz, "A practical method of calculating the corrosion rate of uniformly de-passivated reinforcing bars in concrete," *Materials and Corrosion*, vol. 58, pp. 265-272, April 2007.
- [6] J. Warkus, M. Raupach, and J. Gulikers, "Numerical modeling of corrosion – Theoretical backgrounds," *Materials and Corrosion*, vol. 57, pp. 614-617, August 2006.
- [7] B. Zhao, J. Li, R. Hu, R. Du, and C. Lin, "Study on the corrosion behavior of reinforcing steel in cement mortar by electrochemical noise measurements," *Electrochimica Acta*, vol. 52, pp. 3976-3984, March 2007.
- [8] D. A. Jones, *Principles and Prevention of Corrosion*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1996.
- [9] "Deterioration of construction materials," class notes for CEE 500D, Department of Civil Engineering, University of Illinois at Urbana-Champaign, Fall 2006.
- [10] Concrete Technology, "Corrosion of embedded materials," February 2008. [Online]. Available: [http://www.cement.org/tech/cct\\_dur\\_corrosion.asp](http://www.cement.org/tech/cct_dur_corrosion.asp).
- [11] J. Popovics, G. Gallo, P. Chapman, and M. Shelton, "Basis and application of GMR sensors for reinforced concrete elements," presented at World Forum on Smart Materials and Smart Structures Technology, Chongqing, China, May 2007.

- [12] “Analysis techniques for large scale electrical systems,” class notes for ECE 598 GG, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Fall 2006.
- [13] S. V. Marshall and G. G. Skitek, *Electromagnetic Concepts and Applications*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 1990.
- [14] W. H. Hayt and J. A. Buck, *Engineering Electromagnetics*, 6th ed. New York, NY: McGraw Hill, 2001.
- [15] M. Misakian, “Equations for the magnetic field produced by one or more rectangular loops of wire in the same plane,” *Journal of Research of the National Institute of Standards and Technology*,” vol. 105, pp. 557-564, July-August 2000.
- [16] M. Hayes, *Statistical Digital Signal Processing and Modeling*. New York, NY: John Wiley & Sons, 1996.
- [17] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*. Upper Saddle River, NJ: Prentice-Hall, 1997.
- [18] P. D. Welch, “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Trans. Audio Electroacoustics*, vol. AU-15, pp. 70-73, June 1967.
- [19] D. Bertsekas and J. Tsitsiklis, *Introduction of Probability*. Belmont, MA: Athena Scientific, 2002.

## APPENDIX A: FEA MODEL CODE

### A.1 Finite Element Model Main Program

```
clear all
clc
tic
%-----VARIABLE DECLARATIONS-----%
uR=5000;
num_sim = 1;
[p,t,e] = mesh_data();           %Model Mesh Data Variables: p,t,e.
simLength = 1;                  %simLength: The number of samples
                                %generated by the simulation.
Bsims = zeros(simLength,num_sim);%Vector of Flux Density data generated
                                %in the multiple simulations.
[S,b,c,areas] = calculateS_once(p,t,e,uR);
                                %S, b, c, and areas
                                %S: Shape Function Matrix generated
                                %by function.
                                %areas: Array containing the areas of
                                %each triangle element.

lp = length(p);
deltaT = 1;                      %deltaT: Determines the size of the
                                %simulation time step.
stdDev = 1;                      %stdDev: Standard deviation used to
                                %make current distribution
                                %Sequence from 1 to length(t)
k = 1:1:length(t);
indexmask = (t(4,:) == 2);
cRegion = k(indexmask);         %cRegion: List of triangle elements
                                %in the corroding region.

for num_sim = 1:1:num_sim
    CurrentMat = zeros(length(t),2); %CurrentMat: Contains current
    %information per triangle in the mesh. Each Column contains the
    %data for one time step. Initialized with first set of current
    %values
    CurrentMat(cRegion,2) = stdDev*(randn(1,length(cRegion)));
                                %nomral distributions with mean 0
    CurrentMat(cRegion,2) = CurrentMat(cRegion,2) -...
        sum(CurrentMat(:,2))/length(cRegion);
                                %Balance Current.

    J = sparse(lp,1);
    J = J + accumarray( t(1,:)', CurrentMat(:,2).*areas(:)./3);
    J = J + accumarray( t(2,:)', CurrentMat(:,2).*areas(:)./3);
    J = J + accumarray( t(3,:)', CurrentMat(:,2).*areas(:)./3);
    %Time Step: Counts the number of time steps taken in the
    %simulation.
    % To change the size of the time step change the deltaT.
    timeStep = 1;
%-----END VARIABLE DECLARATIONS-----%
for timeStep = 1:1:(simLength/deltaT) %simulation loop
    [B] = FEA(p,t,e,CurrentMat(:,2),uR,S,J,b,c,areas);
    Bsims(timeStep,num_sim) = B(255);
    %Generates new current based on normal distribution with mean
    %at previous timeStep and the same standard deviation as
    %before.
    CurrentMat(:,1) = CurrentMat(:,2);
```

```

        CurrentMat(cRegion,2) = CurrentMat(cRegion,1) +...
            (stdDev*randn(length(cRegion),1));
        CurrentMat(cRegion,2) = CurrentMat(cRegion,2) -...
            sum(CurrentMat(:,2))/length(cRegion);
    J = sparse(length(p),1);
    J = J + accumarray( t(1,:) , CurrentMat(:,2).*areas(:)./3);
    J = J + accumarray( t(2,:) , CurrentMat(:,2).*areas(:)./3);
    J = J + accumarray( t(3,:) , CurrentMat(:,2).*areas(:)./3);
end % Simulation Loop
end
time = 1:deltaT:deltaT*simLength;
figure(1)
plot(time,Bsims)
figure(2)
pdeplot(p,e,t,'xydata',CurrentMat(:,2),'xystyle','flat',...
        'colormap','hsv','mesh','on')
axis equal tight
title('current density plot')
figure(3)
pdeplot(p,e,t,'xydata',B,'xystyle','flat','colormap','jet','mesh','on')
axis equal tight
figure(4)
pdeplot(p,e,t,'xydata',B,'xystyle','off','contour','on','mesh','on')
axis equal tight
toc

```

## A.2 Shape Function Matrix Function

```

%Calculate S one time Function.
function [S,b,c,areas] = calculateS_once(p,t,e,uR)
S = sparse(length(p),length(p));
%J = sparse(length(p),1);
b = zeros(length(t),3);
c = zeros(length(t),3);
areas = zeros(length(t),1);
u0 = pi*4e-7;
xmin = min(p(1,:))/10000;
xmax = max(p(1,:))/10000;
ymin = min(p(2,:))/10000;
ymax = max(p(2,:))/10000;
for tri = 1:length(t)
    %get Triangle X and Y coordinates
    xi = p(1,t(1,tri))/10000;
    yi = p(2,t(1,tri))/10000;
    xj = p(1,t(2,tri))/10000;
    yj = p(2,t(2,tri))/10000;
    xk = p(1,t(3,tri))/10000;
    yk = p(2,t(3,tri))/10000;
    %Calculate b and c terms.
    bi = yj-yk;
    ci = xk-xj;
    bj = yk-yi;
    cj = xi-xk;
    bk = yi-yj;
    ck = xj-xi;

```

```

b(tri,:) = [bi,bj,bk];
c(tri,:) = [ci,cj,ck];
if t(4,tri)==2 || t(4,tri)==3
    u = uR*u0;
else
    u = u0;
end
%Calculate Triangle Area
areas(tri) = abs((xj*yk-xk*yj)+(xi*yj-xj*yi)+(xk*yi-xi*yk))/2;
K = 1/(4*u*areas(tri));
%S entries for Row I
S(t(1,tri),t(1,tri)) = S(t(1,tri),t(1,tri)) + K*(bi^2 + ci^2);
S(t(1,tri),t(2,tri)) = S(t(1,tri),t(2,tri)) + K*(bi*bj + ci*cj);
S(t(1,tri),t(3,tri)) = S(t(1,tri),t(3,tri)) + K*(bi*bk + ci*ck);
%S entries for Row J
S(t(2,tri),t(2,tri)) = S(t(2,tri),t(2,tri)) + K*(bj^2 + cj^2);
S(t(2,tri),t(1,tri)) = S(t(2,tri),t(1,tri)) + K*(bi*bj + ci*cj);
S(t(2,tri),t(3,tri)) = S(t(2,tri),t(3,tri)) + K*(bj*bk + cj*ck);
%S entries for Row K
S(t(3,tri),t(3,tri)) = S(t(3,tri),t(3,tri)) + K*(bk^2 + ck^2);
S(t(3,tri),t(1,tri)) = S(t(3,tri),t(1,tri)) + K*(bi*bk + ci*ck);
S(t(3,tri),t(2,tri)) = S(t(3,tri),t(2,tri)) + K*(bj*bk + cj*ck);
%Add boundary conditions
if xi==xmin || xi==xmax || yi==ymin || yi==ymax
    S(t(1,tri),:) = 0;
    S(:,t(1,tri)) = 0;
    S(t(1,tri),t(1,tri)) = 1;
elseif xj==xmin || xj==xmax || yj==ymin || yj==ymax
    S(t(2,tri),:) = 0;
    S(:,t(2,tri)) = 0;
    S(t(2,tri),t(2,tri)) = 1;
elseif xk==xmin || xk==xmax || yk==ymin || yk==ymax
    S(t(3,tri),:) = 0;
    S(:,t(3,tri)) = 0;
    S(t(3,tri),t(3,tri)) = 1;
end
end
end

```

### A.3 Finite Element Analysis Function

```

%Finite Element Analysis
%This program will perform a magnetostatic 2-D FEA
function [B] = FEA(p,t,e,currents,uR,S,J,b,c,areas)
%Variable identifications:
% S - shape function matrix (very large (pxp), sparse)
% J - Current density vector of length (p)
% A - Vector of vector potentials at each node, length (p)
% Be - Vector of flux densities
% areas - Vector of areas for each triangle element
% b - Vector of vertice differences in y-direction
% c - Vector of vertice differences in x-direction

%Perform LU on S, forward/backward substitution, and solve for A
A = S\J;

```

```

%Solve for the flux density, Be, for each element then solve for
%the energy level at each element, we, & then the global energy, W
Be = zeros(length(b),2);
u0 = pi*4e-7;
W = 0;
for tri = 1:length(b)
    if t(4,tri)==2 || t(4,tri)==3
        u = uR*u0;
    else
        u = u0;
    end
    K = 1/(2*areas(tri));
    Ae = A(t([1 2 3],tri))';
    ce = c(tri,:)' ;
    be = b(tri,:)' ;
    Be(tri,1) = K*Ae*ce;
    Be(tri,2) = K*Ae*be;
end
B = sqrt(Be(:,1).^2+Be(:,2).^2);

```

## APPENDIX B: PLATE MODEL CODE

### B.1 Plate Model Main Program

```
clear all
clc
stdDev = 1E-6; %standard deviation 1 uA
counter = 1;
simTime = 1024;
scale = 20;
%Define Plate Size
l = 50;
w = 1;
units = 0.1/l;
plate = zeros(1,w,5); %1 = counter
                        %2 = x coordinate of center
                        %3 = y coordinate of center
                        %4 = Used as Anode
                        %5 = Used as Cathode

active = 0.20*l*w;
c_density = 500*10^-6; %uA/cm^2
current = c_density*100; %plate has area of 100cm^2
median = current/active; %median for normal distribution
for i = 1:1:l
    for j = 1:1:w
        plate(i,j,1) = counter;
        counter = counter + 1;

        plate(i,j,2) = (.5+(j-1))*units;
        plate(i,j,3) = (.5+(i-1))*units;
    end
end
rTail = zeros(1,active);
cTail = zeros(1,active);
rTip = zeros(1,active);
cTip = zeros(1,active);
H_Obs = [units*4.5,units*4.4,.01]; %Define Field Observation Point
H = zeros(3,1);
Hstep = zeros(3,1);
dH = zeros(3,1);
for time = 1:1:simTime
    plate(:, :, 4:5) = 0;
    for j = 1:1:active
        %-----Pick the Tail and Tip of Current-----%
        tail_good = 0;
        tip_good = 0;
        tail_bad = 0;
        while tail_good == 0 || tip_good == 0
            if tail_good == 0
                tail = randint(1,1,l*w)+1;
                [rTail(j),cTail(j)] = find(plate(:, :, 1) == tail);
            end
            [tip,tail_bad]=getTip_sim(tail,w,l,plate,rTail(j),cTail(j));
            [rTip(j),cTip(j)] = find(plate(:, :, 1) == tip);
        end
    end
end
```

```

    if plate(rTail(j),cTail(j),4) == 0 &&...
        plate(rTail(j),cTail(j),5) == 0 && tail_bad == 0
            plate(rTail(j),cTail(j),4) = 1;
            tail_good = 1;
        end
    if plate(rTip(j),cTip(j),4) == 0 && tail_good == 1
        plate(rTip(j),cTip(j),5) = 1;
        tip_good = 1;
    end
end
%-----%
L = [plate(rTip(j),cTip(j),2)-plate(rTail(j),cTail(j),2);
    plate(rTip(j),cTip(j),3)-plate(rTail(j),cTail(j),3);
    0];% currents are in the x-y plane
Lmag = sqrt(L(1)^2 + L(2)^2 + L(3)^2);
dl = Lmag/scale;
direct = L./scale;
aI = L/Lmag;
Imag = abs(stdDev.*randn(1)) + median;
for i = 1:1:Lmag/dl
    xp = plate(rTail(j),cTail(j),2) + i*direct(1);
    yp = plate(rTail(j),cTail(j),3) + i*direct(2);
    zp = 0;
    R12 = [H_Obs(1)-xp, H_Obs(2)-yp, H_Obs(3)-zp];
    Rmag = sqrt(sum( R12.^2));
    a12 = (R12./Rmag)';
    dH(:,i) = cross( Imag*dl*aI, a12)/ (4*pi*Rmag^2) ;
end
Hstep(:,j) = [sum(dH(1,:)); sum(dH(2,:)); sum(dH(3,:))];
end
H(:,time) = [sum(Hstep(1,:)); sum(Hstep(2,:)); sum(Hstep(3,:))];
end

```

## B.2 Anode-Cathode Selection Function

```

function [tip, tail_bad] = getTip_sim(tail,w,l,plate,rTail,cTail)
tip = 1;
%Check for corner cases...t
if tail == 1
    if plate(rTail,cTail+1,4)==1 && plate(rTail+1,cTail,4)==1 &&...
        plate(rTail+1,cTail+1,4)==1
            tail_bad = 1;
    else
        tip = randint(1,1,3)+1;
        tail_bad = 0;
        if tip == 1
            tip = tail+1;
        elseif tip == 2
            tip = tail + w+1;
        elseif tip == 3
            tip = tail+w ;
        end
    end
elseif tail == w
    if plate(rTail,cTail-1,4)==1 && plate(rTail+1,cTail-1,4)==1 &&...
        plate(rTail+1,cTail,4)==1

```

```

        tail_bad = 1;
    else
        tip = randint(1,1,3)+1;
        tail_bad = 0;
        if tip == 1
            tip = w-1;
        elseif tip == 2
            tip = w+(w-1);
        elseif tip == 3
            tip = w+w;
        end
    end
elseif tail == w*(l-1)+1
    if plate(rTail-1,cTail,4)==1 && plate(rTail-1,cTail+1,4)==1 &&...
        plate(rTail,cTail+1,4)==1
            tail_bad = 1;
    else
        tip = randint(1,1,3)+1;
        tail_bad = 0;
        if tip == 1
            tip = w*(l-1)+1 - w;
        elseif tip == 2
            tip = w*(l-1)+1 - (w-1);
        elseif tip == 3
            tip = w*(l-1)+1 +1;
        end
    end
elseif tail == w*l
    if plate(rTail-1,cTail,4)==1 && plate(rTail-1,cTail-1,4)==1 &&...
        plate(rTail,cTail-1,4)==1
            tail_bad = 1;
    else
        tip = randint(1,1,3)+1;
        tail_bad = 0;
        if tip == 1
            tip = w*l - w;
        elseif tip == 2
            tip = w*l - (w+1);
        elseif tip == 3
            tip = w*l - 1;
        end
    end
end
%Check for side cases....
elseif mod(tail,w)== 1
    if plate(rTail-1,cTail,4)==1 && plate(rTail-1,cTail+1,4)==1 &&...
        plate(rTail,cTail+1,4)==1 && plate(rTail+1,cTail+1,4)==1 &&...
            plate(rTail+1,cTail,4)==1
                tail_bad = 1;
    else
        tip = randint(1,1,5)+1;
        tail_bad = 0;
        if tip == 1
            tip = tail - w;
        elseif tip == 2
            tip = tail - (w-1);
        elseif tip == 3
            tip = tail + 1;
        end
    end
end

```

```

        elseif tip == 4
            tip = tail + (w+1);
        elseif tip == 5
            tip = tail + w;
        end
    end
end
elseif mod(tail,w) == 0
    if plate(rTail-1,cTail,4)==1 && plate(rTail-1,cTail-1,4)==1 &&...
        plate(rTail,cTail-1,4)==1 && plate(rTail+1,cTail-1,4)==1 &&...
        plate(rTail+1,cTail,4)==1
        tail_bad = 1;
    else
        tip = randint(1,1,5)+1;
        tail_bad = 0;
        if tip == 1
            tip = tail - w;
        elseif tip == 2
            tip = tail - (w+1);
        elseif tip == 3
            tip = tail - 1;
        elseif tip == 4
            tip = tail + (w-1);
        elseif tip == 5
            tip = tail + w;
        end
    end
end
elseif tail >= 2 && tail <= w-1
    if plate(rTail,cTail-1,4)==1 && plate(rTail+1,cTail-1,4)==1 &&...
        plate(rTail+1,cTail,4)==1 && plate(rTail+1,cTail+1,4)==1 &&...
        plate(rTail,cTail+1,4)==1
        tail_bad = 1;
    else
        tip = randint(1,1,5)+1;
        tail_bad = 0;
        if tip == 1
            tip = tail - 1;
        elseif tip == 2
            tip = tail + (w-1);
        elseif tip == 3
            tip = tail + w;
        elseif tip == 4
            tip = tail + (w+1);
        elseif tip == 5
            tip = tail + 1;
        end
    end
end
elseif tail >= w*(1-1)+2 && tail <= 1*w-1
    if plate(rTail,cTail-1,4)==1 && plate(rTail-1,cTail-1,4)==1 &&...
        plate(rTail-1,cTail,4)==1 && plate(rTail-1,cTail+1,4)==1 &&...
        plate(rTail,cTail+1,4)==1
        tail_bad = 1;
    else
        tip = randint(1,1,5)+1;
        tail_bad = 0;
        if tip == 1
            tip = tail - 1;
        elseif tip == 2

```

```

        tip = tail - (w+1);
elseif tip == 3
    tip = tail - w;
elseif tip == 4
    tip = tail - (w-1);
elseif tip == 5
    tip = tail + 1;
end
end
%General Case
else
    if plate(rTail-1,cTail-1,4)==1 && plate(rTail-1,cTail,4)==1 &&...
        plate(rTail-1,cTail+1,4)==1 && plate(rTail,cTail+1,4)==1 &&...
        plate(rTail+1,cTail+1,4)==1 && plate(rTail+1,cTail,4)==1 &&...
        plate(rTail+1,cTail-1,4)==1 && plate(rTail,cTail-1,4)==1
            tail_bad = 1;
    else
        tip = randint(1,1,8)+1;
        tail_bad = 0;
        if tip == 1
            tip = tail - (w+1);
        elseif tip == 2
            tip = tail - w;
        elseif tip == 3
            tip = tail - (w-1);
        elseif tip == 4
            tip = tail - 1;
        elseif tip == 5
            tip = tail + 1;
        elseif tip == 6
            tip = tail + (w-1);
        elseif tip == 7
            tip = tail + w;
        elseif tip == 8
            tip = tail + (w+1);
        end
    end
end
end
end

```

## APPENDIX C: CURRENT LOOP MODEL CODE

### C.1 Current Loop Model Main Program

```
clear all
clc
stdDev = 1E-6; %standard deviation 1 uA
counter = 1;
simTime = 200;
scale = 20;
%Define Plate Size
l = 50;
w = 1;
units = 0.1/l;
plate = zeros(1,w,4); %1 = counter
                        %2 = x coordinate of center
                        %3 = y coordinate of center
                        %4 = Used as Anode

active = 0.20*l*w;
c_density = 500*10^-6; %uA/cm^2
current = c_density*100; %plate has area of 100cm^2
median = current/active; %median for normal distribution
zDepth = 1E-6;
xform = [ 0 0 1; 0 1 0; -1 0 0];
%Define Field Observation Point
H_Obs = [units*5,units*5,.01];
H = zeros(3,1);
Hstep = zeros(3,1);
dH = zeros(3,1);
for i = 1:1:l
    for j = 1:1:w
        plate(i,j,1) = counter;
        counter = counter + 1;
        plate(i,j,2) = (.5+(j-1))*units;
        plate(i,j,3) = (.5+(i-1))*units;
    end
end
for time = 1:1:simTime
    plate(:, :, 4) = 0;
    for j = 1:1:active
        good = 0;
        while good == 0
            element = randint(1,1,l*w)+1;
            [rTail(j),cTail(j)] = find(plate(:, :, 1) == element);
            if plate(rTail(j),cTail(j),4) == 0
                good = 1;
                plate(rTail(j),cTail(j),4) = 1;
            end
        end
        l1 = xform*[plate(rTail(j),cTail(j),2),...
                    plate(rTail(j),cTail(j),3)-(.9*units/2),zDepth]';
        l2 = xform*[plate(rTail(j),cTail(j),2),...
                    plate(rTail(j),cTail(j),3)-(.9*units/2),-zDepth]';
        l3 = xform*[plate(rTail(j),cTail(j),2),...
                    plate(rTail(j),cTail(j),3)+(.9*units/2),-zDepth]';
    end
end
```

```

    l4 = xform*[plate(rTail(j),cTail(j),2),...
                plate(rTail(j),cTail(j),3)+(.9*units/2),zDepth)];
    P = xform*H_Obs';
    I = abs(stdDev.*randn(1)) + median;
    Hstep(:,j) = inv(xform)*loopField(l1,l2,l3,l4,P,I)';
end
H(:,time) = [sum(Hstep(3,:)); sum(Hstep(2,:)); sum(Hstep(1,:))];
end

```

## C.2 Magnetic Field Calculation Function

```

function [H]= loopField(l1,l2,l3,l4,P,I)
u0 = 4*pi*10^-7;
Bx = zeros(3,1);
By = zeros(3,1);
Bz = zeros(3,1);
r(1) = sqrt( (P(1)-l1(1))^2 + (P(2)-l1(2))^2 + (P(3)-l1(3))^2 );
r(2) = sqrt( (P(1)-l2(1))^2 + (P(2)-l2(2))^2 + (P(3)-l2(3))^2 );
r(3) = sqrt( (P(1)-l3(1))^2 + (P(2)-l3(2))^2 + (P(3)-l3(3))^2 );
r(4) = sqrt( (P(1)-l4(1))^2 + (P(2)-l4(2))^2 + (P(3)-l4(3))^2 );
a1 = abs((l1(1) - l2(1)) /2);
b1 = abs((l1(2) - l4(2)) /2);
C(1) = P(1) - l1(1);
C(2) = P(1) - l2(1);
C(3) = P(1) - l3(1);
C(4) = P(1) - l4(1);
d(1) = P(2) - l1(2);
d(2) = P(2) - l2(2);
d(3) = P(2) - l3(2);
d(4) = P(2) - l4(2);
e(1) = P(3) - l1(3);
e(2) = P(3) - l2(3);
e(3) = P(3) - l3(3);
e(4) = P(3) - l4(3);
for i = 1:1:length(C)
    Bx(i) = ((-1)^(i+1)*e(1))/(r(i)*(r(i)+d(i)));
    By(i) = ((-1)^(i+1)*e(i))/(r(i)*(r(i) + (-1)^(i+1)*C(i)));
    Bz(i) = ((-1)^i*d(i))/(r(i)*(r(i)+(-1)^(i+1)*C(i))...
            - C(i)/(r(i)*(r(i)+d(i)));
end
H(1) = (I/(4*pi)) * sum (Bx);
H(2) = (I/(4*pi)) * sum (By);
H(3) = (I/(4*pi)) * sum (Bz);

```