

MAGNETIC SIGNATURE ESTIMATION USING NEURAL NETWORKS

A Thesis
Submitted to
the Temple University Graduate Board

In Partial Fulfillment
Of the Requirements for the Degree
MASTER OF SCIENCE
of ELECTRICAL ENGINEERING

By
Matthew J. Bosack
August, 2012

Thesis Committee:

Saroj Biswas, Thesis Advisor, Electrical Engineering, Temple University

Dennis Silage, Committee Member, Electrical Engineering, Temple University

Frank Higgins, Committee Member, Electrical Engineering, Temple University

Frank Ferrese, Committee Member, Dept. of Naval Surface Warfare, Philadelphia

ABSTRACT

Ferrous objects in earth's magnetic field cause distortion in the surrounding ambient field. This distortion is a function of the object's material properties and geometry, and is known as the magnetic signature. As a precursor to first principle modeling of the phenomenon and a proof of concept, the goal of this research is to predict off-board magnetic signatures from on-board sensor data using a neural network. This allows magnetic signature analysis in applications where direct field measurements are inaccessible. Simulated magnetic environments are generated using MATLAB's Partial Differential Equation toolbox for a 2D geometry, specifically for a rectangular shell. The resulting data sets are used to train and validate the neural network, which is configured in two layers with ten neurons. Sensor data from within the shell is used as network inputs, and the off-board field values are used as targets. The neural network is trained using the Levenberg-Marquardt algorithm and the back propagation method by comparing the estimated off-board magnetic field intensity to the true value. This research also investigates sensitivity, scalability, and implementation issues of the neural network for signature estimation in a practical environment.

ACKNOWLEDGEMENTS

It was with a great deal of effort that this thesis has been completed, and in no small part to the many people who provided assistance along the way. Sincere and grateful thanks are due to my thesis adviser, Dr. Saroj Biswas, who has been an excellent guide and mentor throughout my time at Temple University.

Thanks also to my thesis committee, Dr. Frank Higgins, Temple University, and Dr. Dennis Silage, Temple University, for their time and effort. Your expertise has been invaluable. Frank Ferrese, of Naval Surface Warfare Philadelphia, was instrumental to the project's funding, and his guidance and contributions throughout enabled its success.

I'd like to acknowledge Dr. Axel Kohlmeyer, of Temple's High Performance Cluster, for helping me get acclimated and tolerating near crashes of the system.

I am extremely grateful to the Office of Naval Research, George Stimak, and Anthony Seman, who provided considerable financial support to this work and my schooling. I also have much appreciation for the Naval Surface Warfare, Carderock Division, specifically Christopher Burgy and Dr. Brian Glover, who provided many concepts and ideas.

Lastly, an enormous thanks to my entire family, and especially Mom, whose enduring support provided the extra encouragement that was essential to make it through.

TABLE OF CONTENTS

ABSTRACT	i
ACKNOWLEDGEMENTS	ii
LIST OF FIGURES	v
LIST OF TABLES	vii
1 INTRODUCTION	1
1.1 Motivation	2
1.2 Objectives and Research Outline	3
1.3 Outline of the Thesis	4
2 A BRIEF REVIEW OF MAGNETIC FIELDS AND NEURAL NETWORKS	5
2.1 Magnetic Fields	5
2.1.1 Mathematical Operators	7
2.1.2 Magnetization of Structures	8
2.2 Neural Networks	12
2.2.1 Regression Analysis	14
2.2.2 Least Mean Square Algorithm	15
2.3 Magnetic Signature Estimation	16
3 EXPERIMENTATION	18
3.1 Experiment Design	18
3.1.1 Theory of Helmholtz	19
3.2 Procedure and Results	20
4 MODELING MAGNETIC ENVIRONMENTS IN 2-D	24
4.1 Finite Element Method and the PDETool	26

4.2	The Shell Structure	29
4.3	Boundary Conditions and the Magnetic Field Intensity	30
5	NEURAL NETWORK ESTIMATION OF MAGNETIC FIELDS	34
5.1	Topology and Configuration	34
5.2	Data Collection	37
5.3	Neural Network Estimation	39
5.4	Back Propagation in Multi Layer Networks	40
5.4.1	Training	42
5.4.2	The HPC Super Cluster	43
6	SIMULATION RESULTS	45
6.1	Simulation	45
6.2	Upscaling and Parameter Sensitivity	50
6.2.1	Model Scaling	50
6.2.2	Variations in Material Properties	52
6.2.3	Environmental Sensitivity and Noise	53
7	CONCLUSION	57
7.1	Summary	57
7.2	Suggestions for Future Research	58
	BIBLIOGRAPHY	60

LIST OF FIGURES

2.1	(a) Random moment without an incident \mathbf{B} field. (b) Magnetization after applied \mathbf{B} field. [16]	10
2.2	Some properties of magnetic materials and their classifications.	11
2.3	Hysteresis curve for ferromagnetic materials. [16]	12
2.4	Neural Network architecture with weights, neurons, and activation function.	13
2.5	Regression analysis approximates an output by multiplying the regressor times a parameter and sums.	14
3.1	Honeywell magnetic flux sensor for field measurements.	19
3.2	Constructed coil with I-beam section for signature creation.	22
3.3	Real time data for magnetic signature disruption of a field.	23
4.1	Map of Earth's Magnetic Field [12].	25
4.2	Environmental geometry prior to magnetization.	27
4.3	Discretized domain for solving with FEM.	28
4.4	Resulting \mathbf{B} field of shell with noise in the ambient environment.	30
4.5	The boundary conditions establishing the magnetic vector potential.	31
5.1	Overview of the network architecture for magnetic signature estimation.	35
5.2	Preprocessing techniques for normalizing neural network input data.	36
5.3	Transfer functions used to regulate layer outputs.	37
5.4	The theorized location of input and output sensor nodes about the shell.	38
5.5	A block diagram of the system from modeling to network training.	38
5.6	A multilayer network configuration.	40
6.1	MSE during network training.	46
6.2	Error histogram for scaled up network training.	47
6.3	Linear regression plot of network.	48
6.4	Estimation of magnetic signature from simulated environment.	49

6.5	3D Visualization of simulated object and magnetic signature.	49
6.6	Preprocessing techniques for normalizing neural network input data. .	54
6.7	Geometric representation and first principle signature of a sphere. . .	55
6.8	PDETool Magnetic Signature of the sphere from figure 6.7.	56

LIST OF TABLES

3.1 Helmholtz Coil Parameters.	20
--	----

CHAPTER 1

INTRODUCTION

It has been a long standing goal of scientists and engineers to accurately predict magnetic signatures. A magnetic signature arises from the interaction between an ambient magnetic field and a ferrous object, and is unique to the objects size, shape, and material. Accurately obtaining magnetic signatures through measurement or estimation allows detection and classification of objects, as well as opportunities for signature neutralization. But despite continued efforts and advances in magnetic signature research, certain aspects of the phenomenon remain unsolved. The relationship between magnetic materials and their environment are complex, and governed by many dependent variables. The objective of this research is to provide a methodology to indirectly estimate the magnetic signatures of objects in environments not readily accessible to sensors.

Current methods for measuring object signatures include passive sensor arrays in target environments, and signature reduction is facilitated through compensative current loops [5, 7]. While effective to an extent, typical measurement systems require direct sensor placement in fixed locations, and compensative maintenance requires equipment down time. Furthermore, first principle attempts to characterize magnetization and signatures in variational environments remain problematic [8]. Through the use of a neural network developed from specific model based designs,

magnetic signatures can be accurately estimated using internal object measurements. This mitigates the need for environmental access, provides continuous signature awareness, and can be developed for use in active compensation systems.

The development of neural networks predate the magnitudes of today's computing power, but their objective remains the same: approximate unknown systems. The 'black box' of magnetic signature induction is yet to be fully understood, and neural networks can serve to predict its current effects, and be used in future model validation. Potential also exists for their use in active control and degaussing of signatures. Generalized approximations of signatures has always been included in many application design phases, and the technique used here offers rapid prototyping capability. This thesis investigates a neural network approach and develops a technique for modeling and estimating magnetic signatures of ferrite structures.

1.1 Motivation

A good deal of work over the last two decades pertains to computation of external magnetic fields, magnetic detection of bodies using neural networks, and magnetic classification of vehicles. Inductive loop detectors have long been used in both military and civilian applications for detecting the presence of magnetic bodies. Harbor beds employ side by side loops and use subtraction methods to minimize noise in submarine detection. Transportation departments embed sensors in roadways to determine statistical tracking and flow of traffic patterns [14]. Nevertheless, precise determination of the magnetic footprint of ferrite objects remains an unresolved problem.

Applications of neural networks are regularly used in pattern recognition, and find use with magnetic signatures. Recorded magnetic signatures can be run through a network to determine size, distance, and geometries of objects. The general outcomes

of such works tend to convey that the neural network approach for estimating magnetic bodies has seen a success, and that modeling magnetic fields with the finite element method provides good simulation data. However, these approaches typically employ data from observational fields [10, 9]. The problem presented in this thesis requires the computation of inverse field sources. In particular, this research pertains to accurate determination of magnetic fields in areas where measurement data is unavailable, and estimations are carried out based on sensor data from on-board locations.

1.2 Objectives and Research Outline

The goal of this thesis is to generate a generic model for magnetic signatures, and use it to develop an inverse estimation technique for the magnetic flux density outside the structure. The method considers a two phase approach: simulated interaction between an incident magnetic environment and a ferrite object, followed by neural network training and validation. The simulation provides control over the magnetic incident fields and object properties for data generation, which is used to train the neural network. A framework is provided which validates the simulations likeliness to physical systems, including normalized parameters for model scalability. Experimentation is also discussed, and a test system is constructed for use in magnetic signature observation, as well as future work for model validation.

Successful implementation of a neural network for magnetic signature estimation first requires the composition of an exhaustive data set in which to train the network. This data set is built within the MATLAB PDETool through use of a generalized model developed in previous research in [7]. Maxwell's equations are used to solve for the magnetic vector potential of the domain. From this model, input and target vectors of the magnetic potential are chosen to represent known sensor data from

within the object, and the desired output to be approximated. A neural network object is configured in MATLAB to first train its weight vectors on the input/output pair, and then is subjected to unknown sets of magnetic data to assess its performance.

1.3 Outline of the Thesis

Estimating a magnetic signature is a problem covering a vast domain. Between the sources of magnetization, effects of hysteresis, and other physical properties giving influence over an object's magnetic properties, classification can become quite cumbersome. The main topics of this thesis are presented as follows: Chapter 2 provides introductory material on magnetic fields and neural networks. The experimental design is covered in Chapter 3, which physically defines magnetic signatures. Chapter 4 introduces the shell structure, modeling, and the Finite Element Method. Chapter 5 provides the technique's on data collection and neural network configuration. Chapter 6 presents the results, with Chapter 7 providing concluding remarks.

CHAPTER 2

A BRIEF REVIEW OF MAGNETIC FIELDS AND NEURAL NETWORKS

2.1 Magnetic Fields

Electromagnetic fields and waves arising from time-varying currents are governed by the set of equations developed by James Clerk Maxwell in the mid 19th century. Maxwell's equations give rise to the intrinsic relations between the magnetic field intensity, \mathbf{H} , the electric field intensity, \mathbf{E} , and the surface current density, \mathbf{J} . Using these equations, in conjunction with the appropriate boundary conditions, a regional solution can be obtained for studying the magnetic field of a ferromagnetic object in the immediate environment. Maxwell's equations [11] in differential form are as follows:

$$\nabla \cdot \mathbf{D} = \rho_v \quad (2.1)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (2.2)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad (2.3)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \quad (2.4)$$

where \mathbf{B} is the magnetic flux density, \mathbf{D} is the electric flux density, and ρ_v is the free charge density.

Magnetic fields result from the current generated by electron spin in atoms. Our study is concerned with the interaction of the earth's ambient field (produced by Earth's rotating liquid metal core) with ferrite structures. It is known that the average magnetic flux density of the earth's field is approximately 50,000 nT [13]. In free space, the relation between the magnetic flux density \mathbf{B} and the magnetic field intensity \mathbf{H} is given by:

$$\mathbf{B} = \mu_0 \mathbf{H}$$

where μ_0 is the permeability of free space, $\mu_0 = 4\pi \times 10^{-7} \frac{H}{m}$.

Magnetic flux through a surface can be computed using

$$\psi = \int_s \mathbf{B} \cdot d\mathbf{S}$$

where ψ is the magnetic flux crossing the surface. Note however that our exploration takes place across multiple mediums, which employs magnetic boundary conditions. Magnetic field lines don't have any divergence, i.e., magnetic flux lines crossing any closed boundary is zero. This is the magnetic equivalent of Gauss's law, such that

$$\oint \mathbf{B} \cdot d\mathbf{S} = 0 \tag{2.5}$$

A current carrying conductor induces a local magnetic field which is characterized by Ampere's law:

$$\oint \mathbf{H} \cdot d\mathbf{L} = I \tag{2.6}$$

When magnetic flux crosses the boundary between two magnetic media, the direction of the flux path may change depending on the magnetic permeabilities of the two media. In particular, it is known that the normal component of flux

density remains constant in the two media, while the tangential components of the field intensities remain constant at the boundary (assuming that there is no surface current). Using these conditions, it can be verified that

$$\frac{\tan \Theta_1}{\tan \Theta_2} = \frac{\mu_1}{\mu_2}$$

where Θ_1 and Θ_2 are the angle of incidence at the boundary in the two media.

Magnetic field properties are better analyzed using magnetic vector potential, \mathbf{A} , which is defined by

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{2.7}$$

This equation in conjunction with (2.1) can be further simplified to obtain

$$\nabla^2 \mathbf{A} = \mu_0 \mathbf{J} \tag{2.8}$$

This is known as the Poisson's equation, which, if set to equal zero, is a special case known as the Laplace's equation. The general solution as a function of r , the spatial identifier, to (2.8) is [16]

$$\mathbf{A}(r) = \frac{\mu_0}{4\pi} \int \frac{\mathbf{J}(r)}{|\mathbf{r} - \mathbf{r}'|} dv \tag{2.9}$$

The solutions to these equations, and their presentation to modeling systems, will be covered over the next two chapters.

2.1.1 Mathematical Operators

This thesis deals with magnetic objects with 2D geometry. Thus, the equations and operations used to define the behavior of magnetic fields will be described primarily in the Cartesian coordinate system. The operators found throughout the work are described below.

In Cartesian coordinates, an arbitrary vector is represented by

$$\mathbf{A} = A_x \mathbf{a}_x + A_y \mathbf{a}_y + A_z \mathbf{a}_z$$

The gradient of a scalar field V results in a vector field pointing in the direction of greatest rate of change in space and magnitude.

$$\nabla V = \frac{\partial V}{\partial x} \mathbf{a}_x + \frac{\partial V}{\partial y} \mathbf{a}_y + \frac{\partial V}{\partial z} \mathbf{a}_z$$

The divergence of a vector is the outward flux of a unit volume shrinking around a point:

$$\nabla \cdot \mathbf{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z}$$

Curl is the maximal rotational magnitude per unit area, normal to the direction of the area:

$$\nabla \times \mathbf{A} = \left[\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z} \right] \mathbf{a}_x + \left[\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x} \right] \mathbf{a}_y + \left[\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y} \right] \mathbf{a}_z$$

The Laplacian operator defines the divergence of the gradient:

$$\nabla^2 A = \frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} + \frac{\partial^2 A}{\partial z^2}$$

2.1.2 Magnetization of Structures

First principle modeling of magnetic signatures can be carried out using simple geometries, which yield useful information pertaining to signature depth and possible detection ranges. Clearly, practical ferrite structures used in commercial and defense applications are of more complex geometries and will produce magnetic signatures that may be more difficult to analyze; in addition there will be significant variations

in the signatures due to the presence of machines and electrical circuitry inside the structure. Nevertheless, simple geometries provide enough details to determine whether a signature analysis method will be effective or not. This research is based on a structure represented by a 2D rectangle, thus the primary analysis for its signature will be presented in rectangular coordinates.

The process of magnetization occurs at the atomic level, with electrons that orbit a central nucleus and spin on their own axis. The generating principle here is that external magnetic fields applied to a ferrite object produce a force that tends to rotate or align the object along the field lines. Considering an orbiting electron, its path is a current loop, which generates the magnetic dipole moment, \mathbf{m} given by product of the current and the area of the loop. In the presence of an external magnetic field \mathbf{B} , the magnetic dipole experiences a torque, \mathbf{T} ,

$$\mathbf{T} = \mathbf{m} \times \mathbf{B} \quad (2.10)$$

$$\mathbf{m} = IS\mathbf{a}_n \quad (2.11)$$

with I as the current (A), S being the area (m^2), and \mathbf{a}_n is the unit normal to the plane of the loop. The total magnetization \mathbf{M} of an object is the net sum, or orientation of the total magnetic dipole moment from individual contributions. For N atoms in a volume Δv , with the k^{th} atom moment \mathbf{m}_k ,

$$\mathbf{M} = \lim_{\Delta v \rightarrow 0} \frac{\sum_{k=1}^N \mathbf{m}_k}{\Delta v} \quad (2.12)$$

Without an external magnetic field being applied to an object, the magnetic moment will be random and sum to $\mathbf{M} = 0$. With an incident \mathbf{B} field influencing the moment, they will align and the object will be said to be magnetized. The magnetic moment can also give rise to the magnetic vector potential.

$$\mathbf{A} = \frac{\mu_o \mathbf{m} \times \mathbf{a}_r}{4\pi r^2}$$

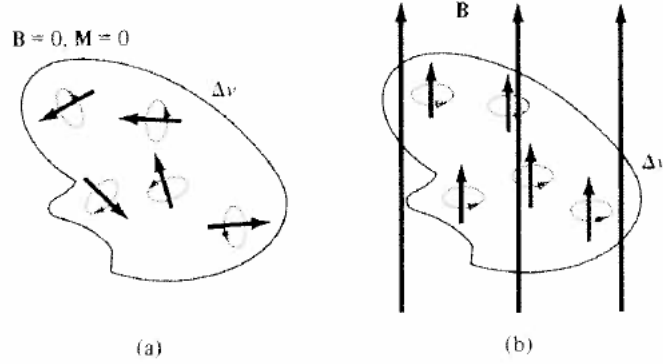


Figure 2.1: (a) Random moment without an incident \mathbf{B} field. (b) Magnetization after applied \mathbf{B} field. [16]

The relationships which hold amongst \mathbf{B} , \mathbf{H} , and \mathbf{M} , are given as follows:

$$\mathbf{B} = \mu_o(\mathbf{H} + \mathbf{M}) \quad (2.13)$$

$$\mathbf{M} = \chi_m \mathbf{H} \quad (2.14)$$

$$\mathbf{B} = \mu_o \mu_r \mathbf{H} \quad (2.15)$$

$$\mu_r = 1 + \chi_m = \frac{\mu}{\mu_o} \quad (2.16)$$

where χ_m is a dimensionless quantity, known as the magnetic susceptibility of a material, and $\mu = \mu_o \mu_r$ is the magnetic permeability of material in Henrys per meter, with μ_o as the permeability of free space, and μ_r is another dimensionless quantity representing the permeability of the given material. Note however that equation (2.13) holds only in linear values of μ defining the properties of magnetic materials. The classification of materials is depicted below:

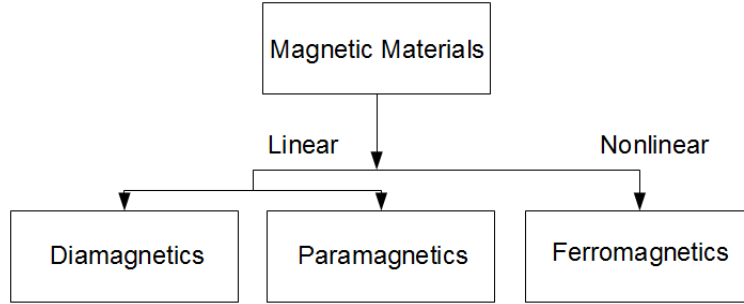


Figure 2.2: Some properties of magnetic materials and their classifications.

Diamagnetic materials have limited susceptibility to fields, with $\mu_r \leq 1$, whereas paramagnetic materials are slightly more influenced, as a result, $\mu_r > 1$. Ferromagnetic materials, denoted as having nonlinear effects within fields, are classified with $\mu_r \gg 1$. A major phenomenon amongst ferromagnetic materials is the hysteresis of the magnetization curve, or B-H curve. It conveys the relationship between the three magnetization parameters. Assuming an unmagnetized start point at the origin, as the magnetization \mathbf{H} increases, the magnetic flux density \mathbf{B} approaches a saturation point following a nonlinear path. However, as the incident field is decreased, the flux density of the material does not return to the origin with the field intensity. This lag is known as hysteresis, and is cyclic in nature as shown in the figure below. The retained magnetization in materials is highly undesirable in many applications. While empirical data is available to predict hysteresis, along with a few approximation models, its behavior is still relatively unknown under certain conditions. The problem of magnetization and ferromagnetic hysteresis are largely responsible for magnetic signatures, and their parameters must be accounted for in order for successful estimation and reduction of signatures.

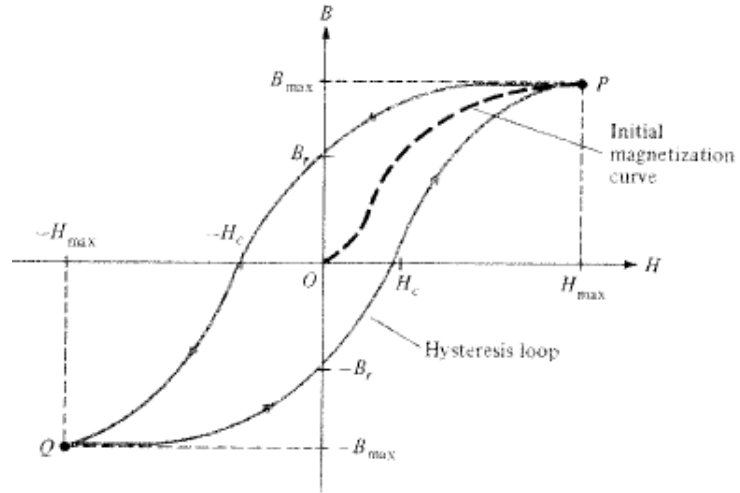


Figure 2.3: Hysteresis curve for ferromagnetic materials. [16]

2.2 Neural Networks

Artificial neural networks began in 1958 with Rosenblatt's Perceptron, evolved with Widrow and Hoff's Least Mean Squares algorithm, and has continued to serve as a computationally efficient method for function approximations.

The artificial neural network is an interconnection of simple processing units, each capable of carrying simple-to-perform tasks, usually organized in multiple layers. The concept is derived from the behavior of the brain, whose sensory inputs and processing capabilities manage its complex system quite gracefully. The typical model of a neural network is shown in Figure 2.4.

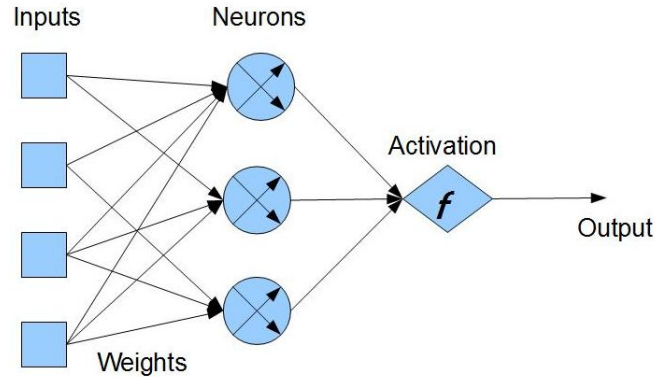


Figure 2.4: Neural Network architecture with weights, neurons, and activation function.

The above figure shows a two layer neural network: the input, neurons of a hidden layer, and an output layer. The number of layers and the number of neurons in each layer are usually unknown, and are determined through simulation to suit a given problem. Each layer contains a fixed set of neurons, or nodes, which can be described by an appropriate activation function and a bias. Three types of activation function are commonly used in the literature: sigmoid function, radial basis function, and linear function [6]. Neurons between two consecutive layers are connected through appropriate weights. The input layer accepts sensory data from the external system for processing. Thus each neuron of the hidden layer receives a weighted sum of output of neurons in the input layer. The neurons of the hidden layer processes this weighted sum through its activation function and bias, which is then applied to the output layer. The interconnection weights and the bias are determined so that the output of the network matches with the output of a system that the neural network is expected to represent. This process of adjusting the interconnection weights and the bias is known as the training of the neural network.

2.2.1 Regression Analysis

Regression analysis is a form of function approximation, in which a set of independent variables are used to illicit the response of a dependent variable. The response is determined by a set of regressors which are the independent variables, operated on by a parameter vector to formulate the dependence. Capable of modeling both linear and nonlinear relationships, regression analysis is well suited for data estimation, and lays the foundation for regularization theory and the least mean square (LMS) algorithm. The linear regression model consists of a known environmental variable, \mathbf{x} , defined by

$$\mathbf{x} = [x_1, x_2, \dots, x_N]^T$$

which will parameterize the output as

$$y_i = \sum_{j=1}^N w_j x_j. \quad (2.17)$$

where \mathbf{w} is the same dimension as \mathbf{x} . Figure 2.5 portrays this process:

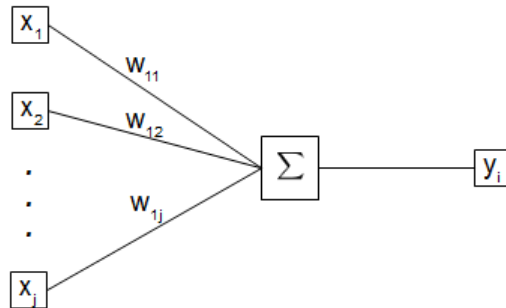


Figure 2.5: Regression analysis approximates an output by multiplying the regressor times a parameter and sums.

Regression analysis works well in statistically defined environments, and readily lends itself to LMS, which improves upon approximation by iteratively updating the parameter vector through minimizing the error.

2.2.2 Least Mean Square Algorithm

The LMS Algorithm developed by Widrow and Hoff is a gradient based iterative method which updates weight vectors along negative gradient directions. The presentation here prepares the theory for the magnetic signature estimator and its implementation as a multilayer neural network discussed in Chapter 5. Reusing Figure 2.4 allows us to visualize a single layer of neurons with a linear activation function and no bias, and in particular, interconnection of the i -th neuron along with all interconnecting weights. The i -th neuron is expected to approximate the target output d_i . The objective is to minimize the error between the target output d_i and the neuron output y_i from equation (2.17), which is defined as

$$\begin{aligned} e_i &= y_i - d_i \\ &= \sum_{j=1}^N w_{ij} x_j - d_i \end{aligned} \tag{2.18}$$

Thus for the network of M neurons, the total error is

$$\begin{aligned} E &= \frac{1}{2} \sum_{i=1}^M e_i^2 \\ &= \frac{1}{2} \sum_{i=1}^M \left(\sum_{j=1}^N w_{ij} x_j - d_i \right)^2 \end{aligned} \tag{2.19}$$

For minimization of the error with respect to the interconnection weight w_{mn} , we find the gradient

$$\frac{\partial E}{\partial w_{mn}} = \left(\sum_{j=1}^N w_{mj} x_j - d_m \right) x_n \tag{2.20}$$

Driving the cost function to zero by optimizing the weight vector obviously achieves a perfect estimate. This is usually accomplished using an iterative method

in which the weights are according to the correction rule

$$w_{mn}(k+1) = w_{mn}(k) - \eta e_m(k) x_n(k) \quad (2.21)$$

where η is the training rate, and k is the iteration counter. The idea behind this iterative method is to reduce the total error in successive iterations, i.e.,

$$E(k+1) \leq E(k).$$

The simplicity and efficiency of this algorithm has served to make it the benchmark of adaptive filters and neural networks. The multilayer neural estimation in Chapter 5 will rely heavily on this work, where a more detailed discussion of the cost function, learning rate parameter, and neural networks will take place.

2.3 Magnetic Signature Estimation

Magnetic bodies are subject to exploitation from their field effects, in applications like those discussed in [7, 14, 9]. First order mathematical and simple physical models of magnetic signatures have been previously developed. Typical early modeling processes use spherical or prolate spheroidal geometries for inquires into material usage and general signature responses. Vertical and horizontal signatures are usually computed separately in 2D models, and allow for parameters such as incident field magnitude, hull thickness, beam length, and permeability. In addition to these first order models, semi-empirical models are created using full scale or physical scale measurement data. The forward-model approach is used to compute magnetic signatures on any line or surface outside the hull. Inverse-models are used [8] to manipulate input sensor data from a ship and compute magnetic source strengths.

A few techniques have been carried out to solve the inverse field problem. The work from [17] uses computation of point charge distributions on circular cylinders of arbitrary sources with Coulomb's law to determine any external magnetic field at any point in space. Fourier series expansion of Green's function is then used to represent the magnetic source on the circular boundary, then moves outward with toroidal expansion. Similar to this work, reference [10] solves the inverse magnetic field problem through use of the finite element method and a two layer neural network. Both techniques were used to model electrical motors. This research continues in the same direction for estimation of magnetic fields using neural network.

CHAPTER 3

EXPERIMENTATION

Magnetic fields and forces are an interesting phenomenon in that we can view their effects without their source. The experimental work here was explored to qualify magnetic signatures as physical entities. The end result gives the ability to explore magnetic signatures with a more tangible understanding, and also paves way for future work to utilize the facilities and procedures to carry out more in depth testing. Experimentation was facilitated through construction of a Helmholtz Coil, and readily available laboratory equipment and sensors at the Temple University Magnetics Laboratory. Although the testing was not done on shapes representing those of the models in the later chapters, it does capture signatures and validate the exploration of signature modeling and estimation.

3.1 Experiment Design

A number of constraints exist on the development of a magnetic testing facility. The most important is the magnetic field strength, which is desired to be that of Earth's, $50,000nT = 0.5G$. Hence a coil must be designed to obtain an evenly distributed $0.5G$. The next consideration falls to the available sensor, the Honeywell HMC-2003. This is a three axis magnetic flux sensor, which can measure between $\pm 2G$ in steps

of $40nT$. Further, from the chip specifications below, the Y-axis spans $20mm$, thus with a $20mm$ spacing between each, and 7 sensors required to match simulation, a uniform field is required over $240mm$.

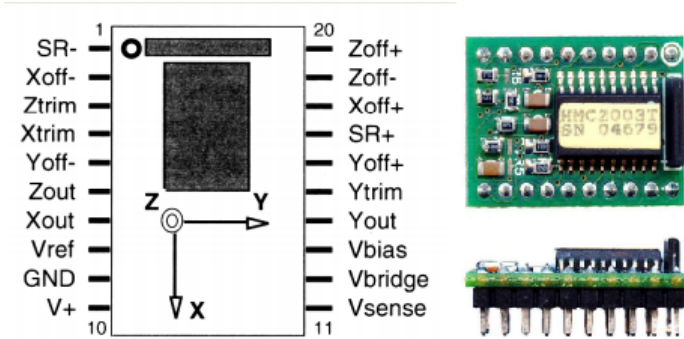


Figure 3.1: Honeywell magnetic flux sensor for field measurements.

It would be well to define the field over a larger area, thus, from the theory of Helmholtz, a $1m$ coil diameter produces a $50cm$ uniform field. The final constraint relates to the first, in that a power supply be available to maintain currents high enough to create the field.

3.1.1 Theory of Helmholtz

Helmholtz coils, named for their creator Hermann Von Helmholtz, are a two identical wire coils spaced one radius apart, functioning to create a uniform magnetic field between them. The current flowing through the coils in the same direction creates a proportional field magnitude, in conjunction with the *right-hand rule*. With N being the number of coil turns, I the current in Amps, R a coil radius in meters, and H the magnetic field intensity,

$$H = 0.715 \frac{NI}{R}$$

The magnetic flux density is obtained by

Table 3.1: Helmholtz Coil Parameters.

Current I	5.86A
Voltage V	14.66 V
Coil Radius R	50 cm
Coil Turns N	19
Coil Length (2)	119.38 m
Coil Resistance (18 AWG)	2.50 Ω
Maximum Field Strength	200,000 nT

$$B = \mu H = 0.899\mu \frac{NI}{R}$$

The *Center Factor*, $CF = 0.715$, is obtained from [1]. From the constraints in 3.1, we can determine how many coil turns are needed to maximally produce $2G$ using a current controlled power supply rated for $200W$. This will lead to the length of coil, total resistance, and required voltage. Table 3.1 lists the desired parameters for the coil operation.

3.2 Procedure and Results

Facilitating the creation and control of the magnetic field requires a good deal of laboratory equipment. The driving waveform is supplied by the Agilent 3648A DC power supply, which is controlled via the National Instruments General Purpose Interface Bus (GPIB). Magnetic field measurements taken at the axis are utilized both for feedback control and data acquisition for the neural network testing. As mentioned, these sensors are from Honeywell, the HMC-2003. A sensor array utilizing seven of the available tri-axis sensors is configured for measurements. The sensors output voltage ranges between $0.5 - 2.5$ volts, representing $\pm 2G$. The voltages are read and transduced by a LabVIEW data acquisition system, namely the NI-9205 voltage input module, supported by the cDAQ-9188 compact DAQ chassis. Each sensor provides four readings, being X, Y, Z, and a reference voltage. The analog

output module, NI-9263, produces voltage signals between the ranges of $\pm 10V$, which will be used to drive currents through the coils. However, an additional power supply is needed, provided by the Kepco Company BOP 200W L model. The power supply is utilized for two reasons, one being the LabVIEW system can only output 1mA, but more importantly the Kepco supply circuitry is specially designed to handle inductive loads, i.e. the helmholtz coil. The Kepco supply accepts the voltage signal from the LabVIEW system as a programmable input, and outputs the required power. Thus the LabVIEW voltage signal controls the output of the BOP unit, driving the coil field.

The coil itself was constructed from three sheets of plywood, where a circular core with a 1 meter diameter and $\frac{15}{16}$ in. thickness was cut for wire wrapping, with a 1.1 meter diameter outer guard. The guard also served as a support base, while slats were used to fix the coil distance at 50 cm. Each coil was wrapped in 19 turns of 18 AWG, which can support up to 16 A [18]. Each coil also had two terminal jacks, where one set served as an input/ground configuration and the other created the connection between the two coils.



Figure 3.2: Constructed coil with I-beam section for signature creation.

The control of the coil and data acquisition was performed in LabVIEW, with the average sensor magnitude providing feedback for setting the field strength. The sensor array as seen in the picture lies in the center of the coil. The I-beam cross section passes through the coil 40cm above the sensors, creating the magnetic signature as seen in figure 3.3. The x-axis *samples* is in reference to time, however there is some distortion to the signature as it does not pass at a uniform speed (manually).

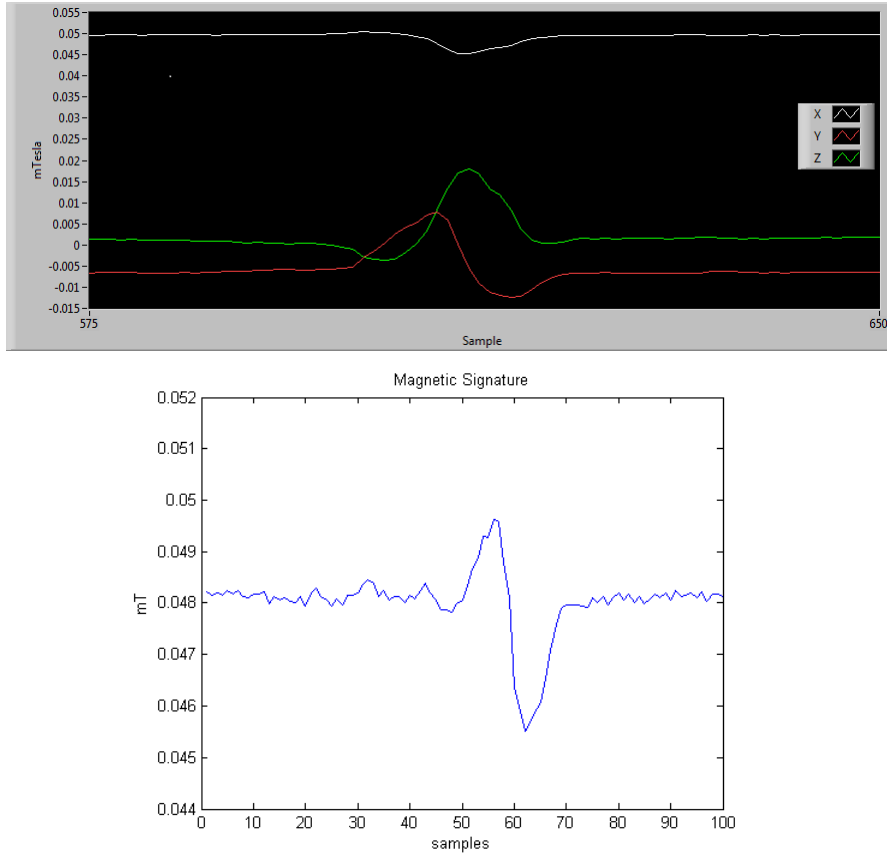


Figure 3.3: Real time data for magnetic signature disruption of a field.

Thus we can see the distortion caused by ferrous objects passing through the field, in the dimensional and magnitude plot. The orientation of the sensor here has the x-axis parallel with that of the coils field, with the z-axis measuring magnitudes above and below and the y-axis from the sides. From the experimental testing set up here future work will be well suited to undertake magnetic signature silencing. With a physical representation of magnetic signatures obtained, we can proceed with modeling and estimation.

CHAPTER 4

MODELING MAGNETIC ENVIRONMENTS IN 2-D

Without data for magnetic signatures of structures being readily available, simulations are run to produce environmental conditions for training data. The current technique models a simple rectangular shell, which includes simulated noise blocks in a 2D environment. These noise blocks may be considered as representation of various electrical machines and other ferrite objects within the shell that may affect the magnetic signature.

The simulated environment provides an ambient magnetic field intensity strictly in the positive x direction. From the United States Geological Survey (USGS) [13], the average magnetic intensity of the Earth is about $50,000nT$, and as seen in Figure 4.1, ranges between $47,250 - 52,250nT$ in the US region.

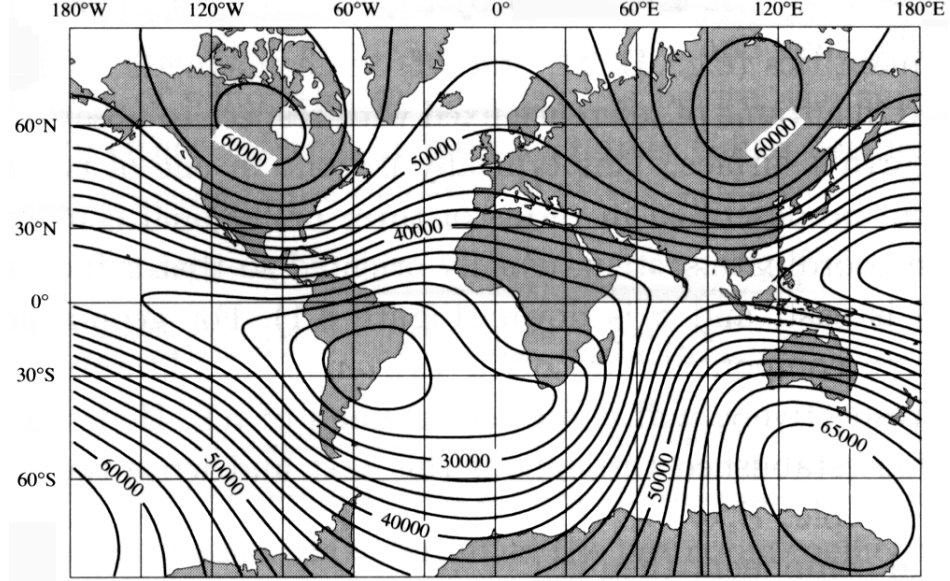


Figure 4.1: Map of Earth's Magnetic Field [12].

Magnetic field distribution is better investigated in terms of magnetic vector potential \mathbf{A} which is defined as

$$\mathbf{B} = \nabla \times \mathbf{A} \quad (4.1)$$

along with appropriate boundary conditions. Then recalling the use of Maxwell's equations (2.1), we know static magnetic fields satisfy

$$\begin{aligned} \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{H} &= \mathbf{J} \end{aligned} \quad (4.2)$$

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

Simplifying the above equations, we obtain

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} \right) = \mathbf{J} \quad (4.3)$$

For the two dimensional geometry, we assume that there is only z -component of

\mathbf{A} -field exists which leads to \mathbf{B} -field in the xy -plane so that

$$\begin{aligned}\mathbf{A} &= (0, 0, A) \\ \mathbf{J} &= (0, 0, J)\end{aligned}\tag{4.4}$$

Substituting the above equations in (4.2), we obtain

$$-\nabla \cdot \left(\frac{1}{\mu} \nabla A \right) = J\tag{4.5}$$

where μ is the magnetic permeability of the region, and J is the current density. We also set $J = 0$ since there is no applied current to the structure. The boundary condition configuration and solution to the system here is detailed in the following section.

4.1 Finite Element Method and the PDETool

MATLAB provides a finite element method toolbox for solving systems of partial differential equations (PDE) which employs the finite element method (FEM) for numerical solution of boundary value problems governed by PDE's. Specifically we use the Matlab toolbox PDETool [1] to study the magnetic field distribution of the 2D geometry as shown in Figure 4.2. The toolbox provides a number of methods for drawing and editing geometry and environments, including editing magnetic properties, boundary conditions, generating FEM meshes, and solving for nonlinear systems.

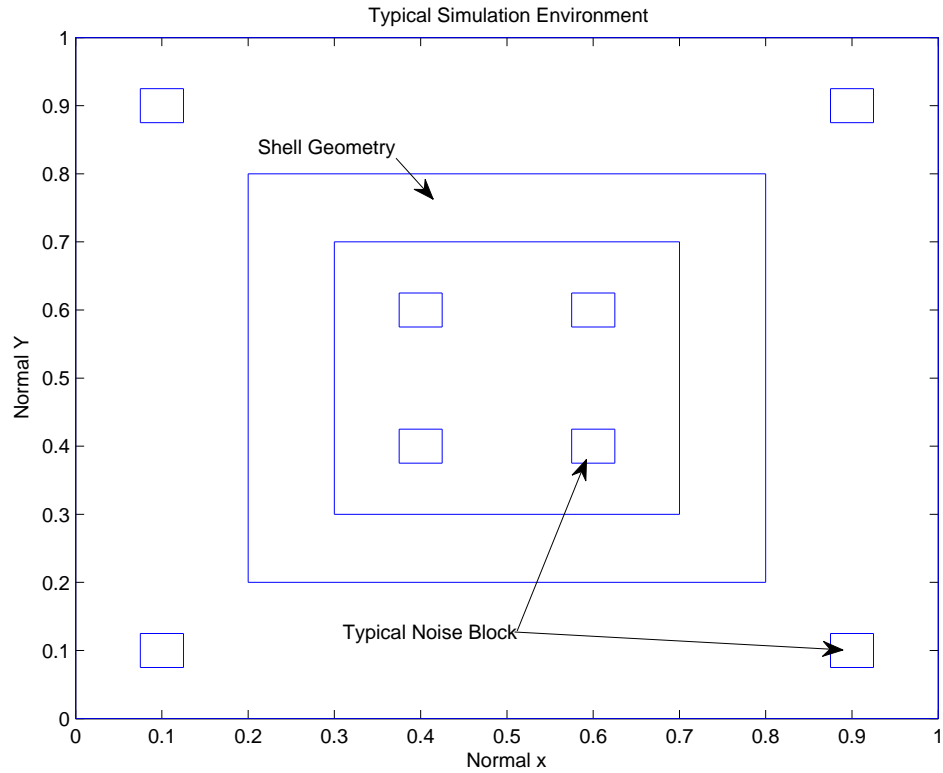


Figure 4.2: Environmental geometry prior to magnetization.

The objective of meshing is to discretize the domain into the finite elements that the system solve for. The toolbox uses the Delaunay Traingulation method to increase mesh densities around areas of rapidly changing value, and ensures none of the discretized points are included within the domains of any other triangle. The overall mesh density is based on the geometry size, and can be increased by inserting new mesh points at the center of existing triangles. A initialized mesh is shown in Figure 4.3, while the mesh used for actual simulation is generated at higher densities.

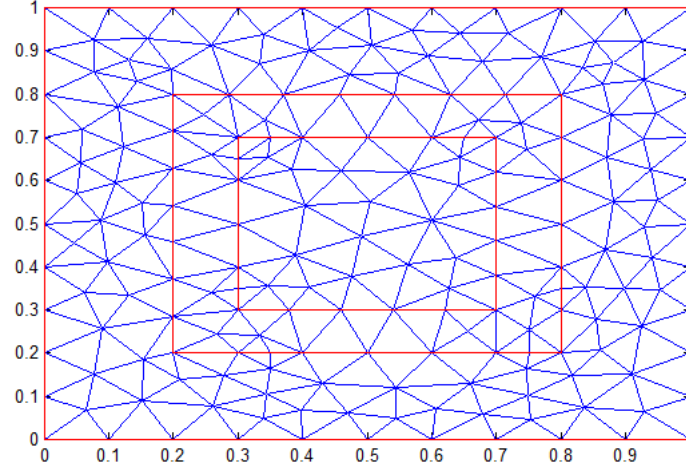


Figure 4.3: Discretized domain for solving with FEM.

Using the finite element method [15], the magnetic vector potential, \mathbf{A} , is solved for each point in the domain. In simulations, the problem uses a mixed set of Dirichlet and Neumann boundary conditions, respectively described as

$$h A = r \tag{4.6}$$

$$n \left(\frac{1}{\mu} \nabla A \right) + q A = g \tag{4.7}$$

with h as a weight, r as the boundary magnetic potential, q is a weight, and g being the magnetic field value. To generate the desired value of the magnetic field, the magnetic potential at the top boundary is set as 2×10^{-4} with that for the bottom boundary taken as the reference; this will be justified in 4.3. The Neumann boundary, which is designed to represent the tangential portion of the normal vector to the boundary, has $q = 0, g = 0$ for both right and left boundaries. Thus an ambient field, which is same as the earth's ambient field, is generated in the positive x-direction across the domain.

We may now proceed to compute the complete solution of the magnetic vector potential in the region for \mathbf{A} , and with equation (4.1) on a 2D domain, can find the

\mathbf{B} and \mathbf{H} fields from

$$\mathbf{B} = \left(\frac{\partial A}{\partial y}, -\frac{\partial A}{\partial x}, 0 \right) \quad (4.8)$$

$$\mathbf{H} = \frac{1}{\mu} \mathbf{B}. \quad (4.9)$$

4.2 The Shell Structure

The work of Holmes [8] models the shell geometry as a prolate spheroid. To investigate the accuracy of the neural network technique, and simplify calculations, this research will be using a rectangular shell. The model in Figure 4.2 depicts a square environment of length and width equal to 1, with a centered hollow rectangle with uniform shell thickness. The environment and shell are normalized in the simulation and can be scaled up to represent larger systems. However, the scaling increases uniformly so the shell thickness must be properly adjusted to account for any distortions that would occur in upscaling. The magnetic permeability of the area surrounding the shell is meant to mimic sea water, $\mu_r = 1$. The physical region of the shell itself is based on a parameter set for $\mu_r = 75$, and the hull interior is air, $\mu_r = 1$. These permeabilities provided the magnetic properties of the environment and the structure; recall that $\mu = \mu_0 \mu_r$, where μ_r is the relative permeability, $\mu_0 = 4\pi \times 10^{-7}$ is the permeability of free space. The subscript r is dropped for simplicity in the rest of the paper, leaving relative magnetic permeability as μ .

A few additional elements are set within the hull, which are referenced here as noise blocks. The noise blocks serve multiple purposes; the first being that the current simulation is set up only to create a magnetostatic field. Thus there would be no variation in the ambient field. The noise blocks are also introduced to behave as magnetic noise, much like that to be experienced in an environment with operating

machinery, circuitry, or other magnetic bodies. The magnetic noise is simulated by randomly varying μ over every magnetic field solution to introduce at most $20dB$ of noise. The positions are set to remain within the shell in non-overlapping configurations. From the methods described above, and the geometry proposed in this section, a single iteration of the simulation is depicted below in Figure 4.4.

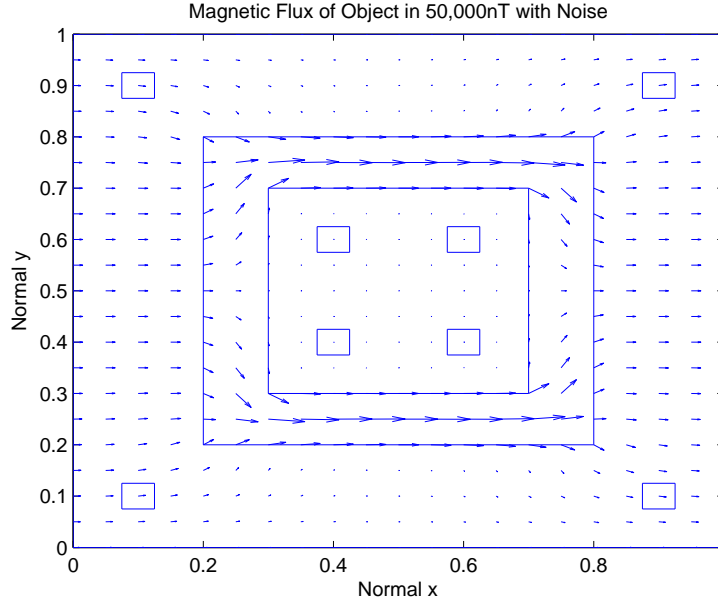


Figure 4.4: Resulting \mathbf{B} field of shell with noise in the ambient environment.

4.3 Boundary Conditions and the Magnetic Field Intensity

MATLAB's PDETool provides an excellent mechanism for studying magnetic fields. However, because our incident field is generated using the boundary conditions, and is representative of a larger field, it must be verified that the simulated field magnitudes equal that of a earth's uniform field magnitude. For this purpose, boundary conditions as shown in Figure 4.5 were used, where A_0 must be set appropriately so as to generate the earth's ambient field in the absence of the structure.

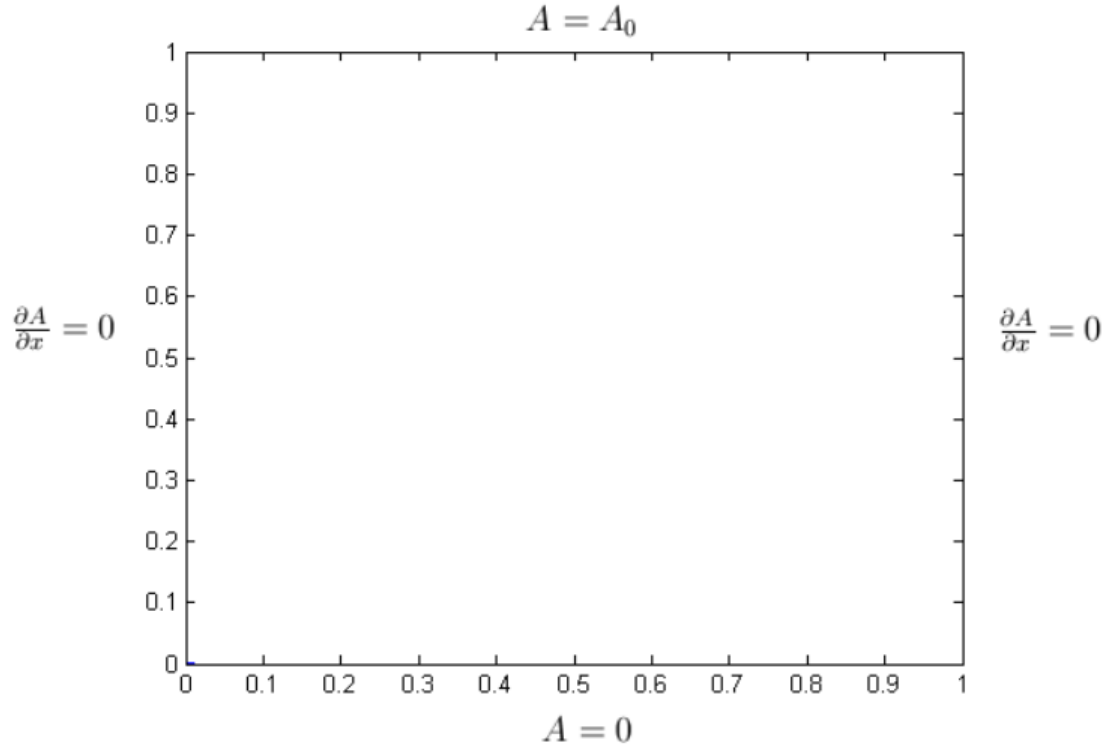


Figure 4.5: The boundary conditions establishing the magnetic vector potential.

For regions free from any ferrite objects, Equation Equation (4.5) simplifies to

$$\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} = 0.$$

Then defining the magnetic vector potential as the product of two functions,

$$A(x, y) = X(x)Y(y),$$

the above equation leads to

$$\ddot{X}Y + XY'' = 0$$

so that

$$-\frac{\ddot{X}}{X} = \frac{Y''}{Y} = C$$

where C is a constant. Here \ddot{X} represents the second derivative of X with respect to

x , and Y'' is the second derivative of Y with respect to y .

A linear variation of the magnetic vector potential A from $A(0) = 0$ to $A(1) = A_0$ can be obtained if the constant $C = 0$. For the vertical y axis, we obtain the valid solution from

$$Y(y) = k_2 + k_1y$$

Using the above boundary conditions, one obtains $k_2 = 0$. This gives

$$Y(y) = k_1y$$

For the solution for $X(x)$, we have

$$\ddot{X} + Cx = 0, \quad X(x) = k_4x + k_3$$

Using the boundary condition

$$\frac{\partial A}{\partial x}(0) = 0 = k_4$$

yielding the magnetic vector potential as

$$A(x, y) = X(x)Y(y) = k_3k_1y = k_5y$$

For a domain with the height h ,

$$A(x, h) = k_5h = A_0 \quad \text{so that} \quad k_5 = \frac{A_0}{h}$$

Thus for an environment free from ferrite objects,

$$A(x, y) = \frac{A_0}{h}y.$$

The magnetic flux density is then easily calculated from

$$\mathbf{B} = \left(\frac{\partial A}{\partial y}, -\frac{\partial A}{\partial x} \right) = \left(\frac{A_0}{h}, 0 \right).$$

So when establishing the boundary conditions, the boundary magnetic vector potential can be set as the desired magnetic flux density magnitude. Clearly this ambient field is of constant magnitude in the x -direction. In the presence of a ferrite structure, the same boundary conditions will be used, and as expected the magnetic field within the domain will be distorted. Such variations can be calculated using the finite element simulation described earlier. Field data obtained from such simulations will be used in the next chapter for training the neural network.

CHAPTER 5

NEURAL NETWORK ESTIMATION OF MAGNETIC FIELDS

The network configuration steps taken in this chapter follow a methodology described in the Neural Network Toolbox User's Guide. The process is relatively simplistic considering the large amounts of connectivity between neurons and layers, and the associated vectors and matrices that go with them. The following sections will expand upon the topics introduced in Section 2.2, and covers inputs, outputs, hidden layers, activation functions, and training. A large number of network configurations were tested on the simulated data using the High Performance Computing Cluster at Temple University, but the two layer ten neuron network will be the main focus here. The results of this network are presented in chapter 6.

5.1 Topology and Configuration

Within the MATLAB Neural Network Toolbox, neural network configuration is highly customizable. Network objects are created that allow users to define any number of inputs, neurons, layers, and outputs. Further, each network object contains subobjects that control how the network will be trained, using what functions, interconnects between layers, and a vast number of other options to create a network

suitable to a user's requirements.

The designed neural network consisted of a (7x1) input vector, each input individually connected to the 10 neurons of the hidden layer. The hidden layer connects to the output layer, designed to match the targets of the (21x1) output vector. These dimensions were chosen based on the total number of grid point available within the shell (7), and the total length of the external environment (21). Weights operate on the input during the hidden and output layers, while the hidden layer provides a log-sigmoidal activation function, and the output layer smooths the data with a linear activation function. Figure 5.1 offers a graphical overview of the layers, activations, and sizes.

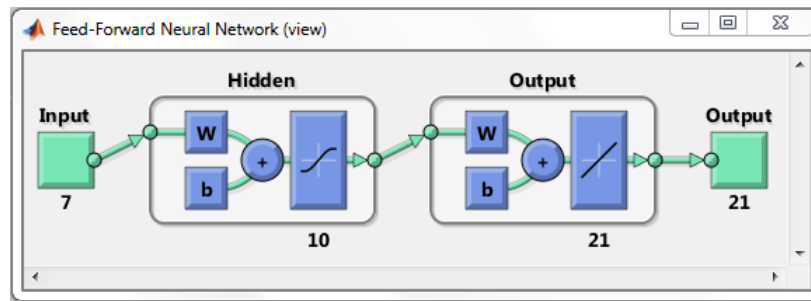


Figure 5.1: Overview of the network architecture for magnetic signature estimation.

The input vector to the system also undergoes some pre-processing functions to assist in optimization of the network training. The technique for this normalizes the mean value over the entire training set to be close to zero. The main advantage of preprocessing in our case is that all of our input data will have a positive magnitude, restricting its motion towards the error surface as underdamped, which should be avoided. The input should also undergo decorrelation and covariance equalization, which also accelerates backpropagation training.

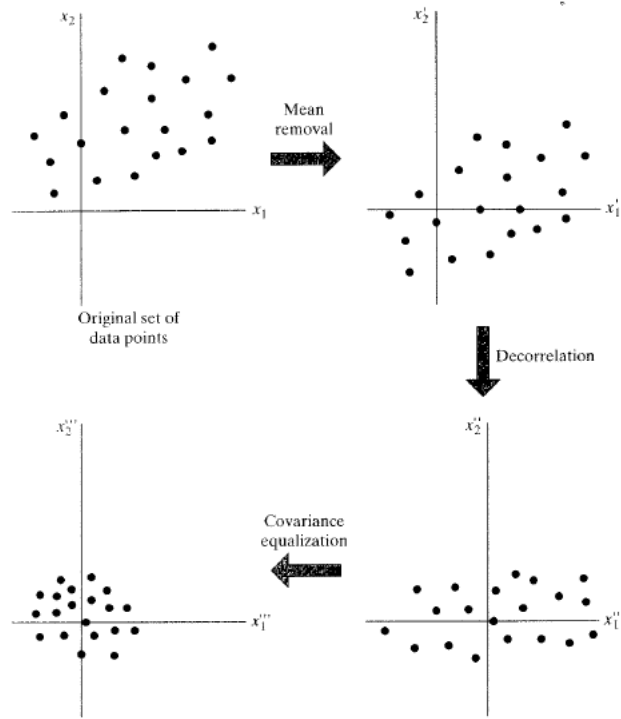


Figure 5.2: Preprocessing techniques for normalizing neural network input data.

At each layer, the weights are randomly initialized at values less than 1. The bias input, a value used to implement an offset, is set at zero throughout the entire process. The bias can be used to offset the input from the limiting value of the activation function. The activation transfer function of the hidden layer uses the logsig function, which accepts inputs with ranges from negative to positive infinity. The purpose of using the activation function is to restrict the output between 0 and 1, and allows the network to learn the nonlinear relationship occurring between \mathbf{B} and \mathbf{H} . The output layer uses the linear activation function, common in pattern recognition problems, to assist in correlating the input and output. The only requirement for these functions is that they are continuously differentiable [3].

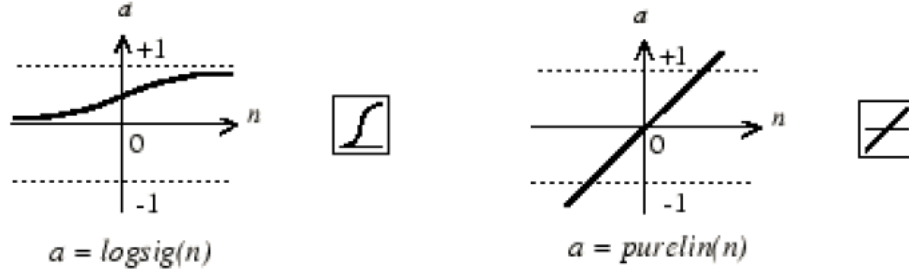


Figure 5.3: Transfer functions used to regulate layer outputs.

5.2 Data Collection

The neural network training will be performed as a batch process, where all of the data will be presented at once to be used for updating the neuron weights. The initial configuration to be studied in this research attempts to estimate the magnetic field at seven points vertically centered outside the top of the shell, based on twenty-one horizontally spaced points outside the shell. The objective is to create a network that can be used to approximate magnetic fields in the range of $47,250 - 52,250nT$ in noise up to $20dB$. However, the errors between simulation data in such a small range provide a poor training set. Thus the environment is actually simulated in an incident field of $5T$. When the network is implemented in a physical system, the data collected will be in the expected range, but must be scaled up to present to the network. The resulting output can be scaled back down to relate back to the original magnetic field magnitudes. Neural networks are inherently bad at predicting outputs from inputs which the network had not been previously trained over. Thus in order to ensure good functionality, a large amount of input/output pairs must be presented to the network. The training data set was created by varying noise fields over each iteration. With a large range of noise and environment values, network training sets were collected over various sizes in the given field range, up to ($i = 75,000$). Figure 5.4 represents the approximate orientation of the sensor point locations in the environment.

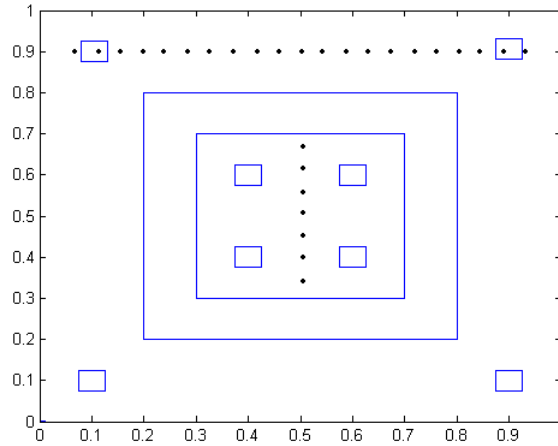


Figure 5.4: The theorized location of input and output sensor nodes about the shell.

After each iteration of the simulation, the PDEtool generates the magnetic field at each triangular mesh point within the discretized domain, which is then interpolated onto a rectangular grid, with a size specified by the user. Each \mathbf{B} field is stored in a 3D matrix. For each iteration, input and target vectors are also created by collating \mathbf{B} -field data corresponding to the location of sensor nodes outside the shell, and the nodes at which the \mathbf{B} -field needs to be estimated within the shell. Corresponding to the total number of iterations, the input and the target data sets are of dimension $7 \times i$ and $21 \times i$. The data set is stored until ready to be called upon by the neural network training procedure. The block diagram below showcases the process.

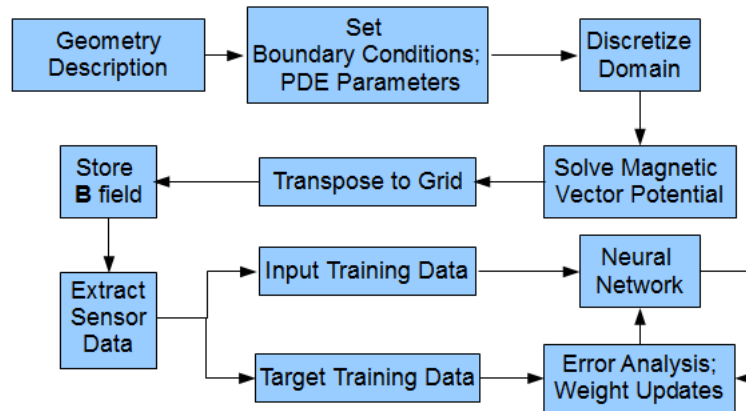


Figure 5.5: A block diagram of the system from modeling to network training.

5.3 Neural Network Estimation

Multilayer networks approximate nonlinear functions using hidden processing layers and nonlinear activation functions. High degrees of connectivity between neuron layers also contribute to improvements over the single model perceptron and LMS algorithm in Section 2.2.1. The neural network operates on the same principle of processing data and updating parameters as in an adaptive filter. The Back-Propagation Algorithm is typically employed for multilayer networks, and is used in this work. This begins by moving data forward through the network of neurons until reaching the output, where the error between the desired and approximate signal is computed. Then the back-propagation begins, where the error is used to update the connection weights of the network so as to reduce the error. Typically a cost function is defined as the sum of square of all errors between the network output and the intended target vector. This cost function serves as the basis for the error correction rule, and this method is repeated until certain stopping criteria are met. The primary steps from start to finish for a neural network training process are as follows:

1. Collect Data
2. Configure and Initialize Network
3. Forward Data Pass and compute Error Signal
4. Backward Data Pass and update network weights
5. Repeat 3 & 4 until stopping criteria is satisfied
6. Implement

5.4 Back Propagation in Multi Layer Networks

Now the task at hand is to extend the linear estimation method to solve for non-linear outputs using multi-layer neural networks and activation functions. The technique here is the back propagation algorithm, in which the weight update rules are driven by the error contribution of each neuron. The partial derivative of the output error will be computed with respect to the adaptive weights, and the vector will update in accordance to each weights error contribution. This section will also introduce layer activation functions, and techniques for normalizing inputs and outputs. The configuration of the network depends on the input vector size, output vector size, and the number of layers. The number of neurons in a layer will specify the size of the weight matrix, and the number of outputs from that layer.

Fundamentally any number of inputs can approximate any number of outputs, as long as at least one hidden layer is included between them to provide the required number of weights for transformation. For sequential hidden layers, the input to any layer of neurons is taken form of the previous layer's output. The interconnection weights are initialized randomly.

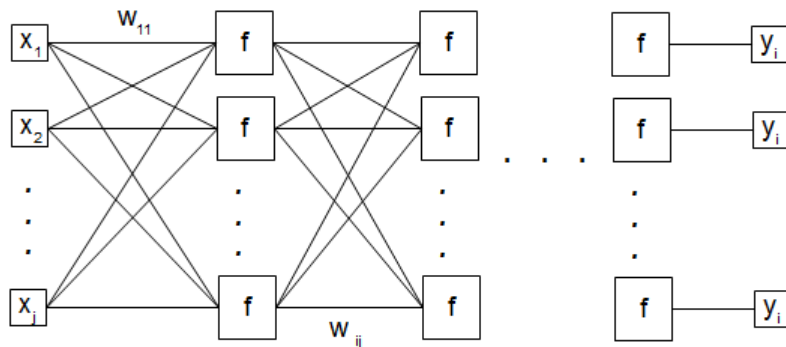


Figure 5.6: A multilayer network configuration.

Various layers of the neural network are numbered as $\ell = 0, 1, 2, \dots, L$, where the input layer is denoted with $\ell = 0$ and the output layer by $\ell = L$. No activation

functions are used in the input layer. The matrix W_ℓ denotes the interconnection between the layers ℓ and $\ell - 1$. There are seven steps in the back propagation algorithm:

1. Initialize the input layer.

$$y_0 = x$$

2. Forward Activation: for $\ell = 1, 2, \dots, L$, compute the output of each layer using

$$y_\ell = f_\ell(W_\ell y_{\ell-1} + b_\ell)$$

where b_ℓ is the bias of the neuron, and f_ℓ is the neural activation function.

3. Calculate the error in the output layer:

$$\delta_L = T - y_L$$

where y_L is the output of the output layer and T is the target output.

4. Back propagate the error to hidden layers: for $\ell = L - 1, L - 2, \dots, 1$,

$$\delta_\ell = W_{\ell+1}^T \delta_{\ell+1} f'_\ell(W_\ell y_{\ell-1} + b_\ell)$$

where f' is the derivative of the activation function.

5. Calculate gradient for weight update

$$\Delta W = \delta_\ell y_{\ell-1}^T$$

6. Update weights

$$W(k+1) = W(k) - \gamma \Delta W(k)$$

where γ is the learning rate, and k is the iteration counter.

7. Repeat the steps until stopping criteria is satisfied

The activation function for the hidden layers uses the logistic-sigmoid function

$$f_{\ell}(u) = \frac{1}{1 + e^{-u}} \quad (5.1)$$

with its first derivative as

$$f'_{\ell}(u) = \frac{e^{-u}}{1 + e^{-u^2}}. \quad (5.2)$$

5.4.1 Training

The training method used in this research is based on the Levenberg-Marquardt minimization algorithm. There are two ways a neural network can be trained: incremental and batch. The first method entails the use of one data point at a time, while the later uses the entire data set simultaneously. This research uses the batch mode. which presents the entire training sample to the network before the weights are updated. Again, the idea is to optimize network performance based on the gradient of network with respect to the weights. Network performance is judged on the mean square error (m.s.e.)

$$m.s.e. = \frac{1}{N} \sum_i^N (t_i - a_i)^2. \quad (5.3)$$

where t_i is the training data and a_i is the output of the neural network, and N is the number of data points used for training.

The training process begins by calling the *train* function, sending it as arguments the input vector, output vector, and the network object. Obviously the objective of

training is to reduce the m.s.e. to zero, which is unlikely to achieve in practice. Thus, *stopping criteria* are introduced to the network object to cease training. Two stopping criteria are used during our training process: if the number of epochs reaches 1000, or the m.s.e. reaches 1×10^{-5} . In addition to this criteria, the HPC Cluster implements a processing wall time to not allow jobs to hog resources, which is currently restricted to 12 hours.

5.4.2 *The HPC Super Cluster*

The computational intensity required for generating the simulation data and training the neural network is extensive. Even using simple geometries on small domains, Finite Element Method (FEM) meshing can result in vast matrices, and neural network training continuously updates large matrices of weight vectors. Further, the high volume of data samples to properly train a network places more strain on field simulations. The High Performance Cluster (HPC) on Temple's main campus provides an excellent resource to handle most of this research's computational needs.

The majority of resources spent thus far in our research was on the devel cluster. Devel has 4 nodes, each with 2 CPU's and 4 cores clocked at 2.66 GHz each. The total available memory is 24 GBytes, with a 120 GByte scratch disk. Comparatively, the workstation uses an Intel i5 processor, with 2 cores at 2.4Ghz and 4 GB of RAM. The cluster was instrumental in determining among different architectures for best network performance in our work. A crucial feature provided by the cluster, not available from our lab workstations, is the high level of parallelism.

The magnetic simulation data is generated from a MATLAB partial differential equation package, PDETool. While the base environmental parameters remain the same, material properties continuously change to provide a rich sample training set. This allows the sets to be generated independently, where a 50,000 sample set

simulated on a lab workstation must be run sequentially, the HPC can run 8 separate scripts creating 6,250 samples each. MATLAB on the HPC is currently configured to only run on one processor, thus neural network training can not yet be optimized.

CHAPTER 6

SIMULATION RESULTS

The preceding work was carried out to simulate the behavior of a ferrite shell in a magnetic environment, using the results to train a neural network to predict future behavior. While simplistic in nature, a successful proof of concept would yield a powerful tool for future research in the modeling, estimation, and neutralization of magnetic signatures. This chapter compiles the simulation results of neural network training and signature estimation, and also discusses questions of scaling and sensitivity.

6.1 Simulation

Training data sets were accumulated using modeled magnetic vector potentials of magnitudes (upper boundary condition) between $A = 4.8$ and $A = 5.2$ in $20dB$ ambient noise. The increase of magnitude was used to induce larger errors than would have been seen using data on the nano-Tesla scale. The range was chosen to represent magnetic field values regionally found across the United States, as well as the average magnetic value of Earth. The step between each set was created large enough not to create additional burdens in accumulating large data sets; the largest set collected 15,000 iterations at each interval for a total of 75,000 samples. The neural network

used a 7×1 input vector to approximate a 21×1 output vector with a 10 node hidden layer. After training, a new data set of 1,000 random samples anywhere in the continuous range was collected and tested on the network. The figures below represent the network training and final performance values from the simulated data. The training record plot, shown in figure 6.1, tracks the mean squared error during the training, and points out the minimum value achieved. Here, performance function measures the normalized mean squared error, obtained at the 177th epoch, achieving a performance of 1.73. The epochs are the total number of presentations of data to the network for training. With no extreme differences between the Train, Validation, and Test curves, the network architecture can be implemented for use in magnetic signature estimating.

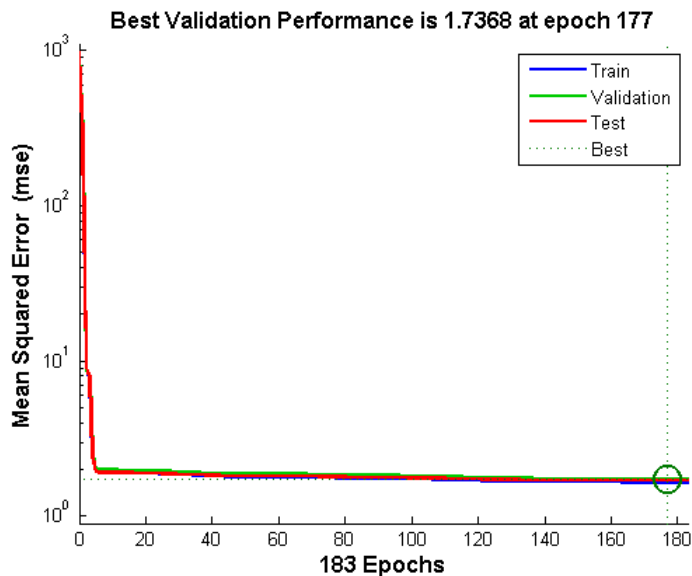


Figure 6.1: MSE during network training.

The histogram below is an easy way of visualizing the amount of errors that occur between the network output and the target data, over every sample. With the overwhelming majority of errors occurring in the two bins located closest to zero error, it further indicates the network architecture will be successful in estimating off board signatures.

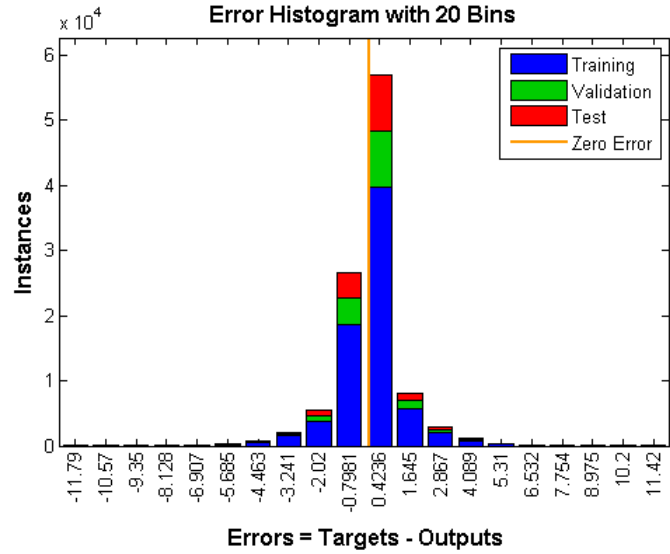


Figure 6.2: Error histogram for scaled up network training.

Finally, the regression plots from figure 6.3, we have a representation of how well the network output relates to the target data. The R value denotes how close a fit we have, where $R = 1$ is a perfect relationship. It is also seen that there are no severe outliers in the data. While this would most likely not be true in a physical data set, in most cases outliers can be ignored, or at least are heavily compensated for by the other neuron weight values.

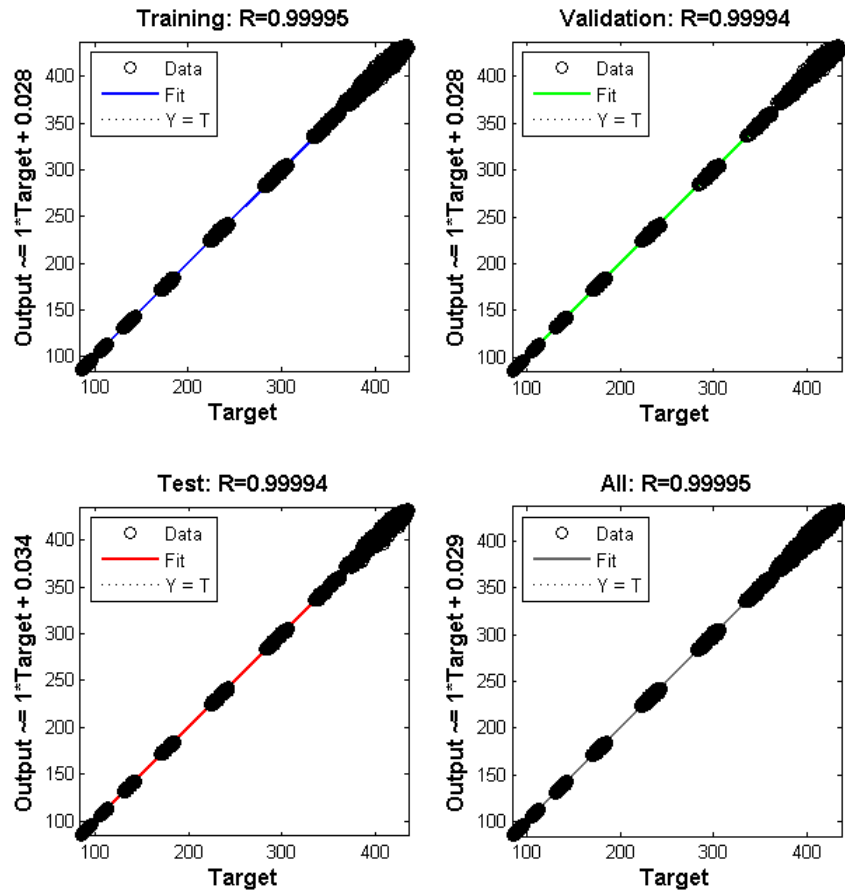


Figure 6.3: Linear regression plot of network.

From the validation set, the maximum error obtained was 2.5%, which corresponds to a magnitude difference of $1,012nT$. This difference is difficult to distinguish in the plot, and while still detectable by sensors, it is not a large deviation from normal environmental changes.

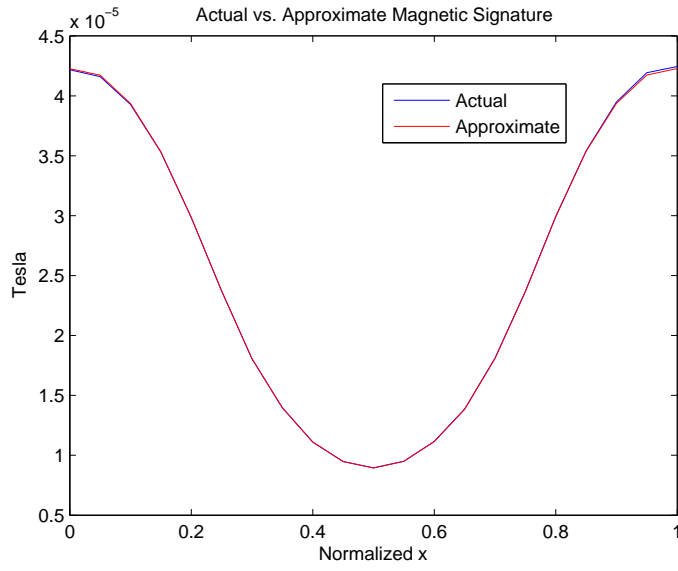


Figure 6.4: Estimation of magnetic signature from simulated environment.

The 3-dimensional plot of the gradient is offered for a comparative view of the flux being estimated, which corresponds to the line at $Y = 2$. However, the estimation is 2-dimensional.

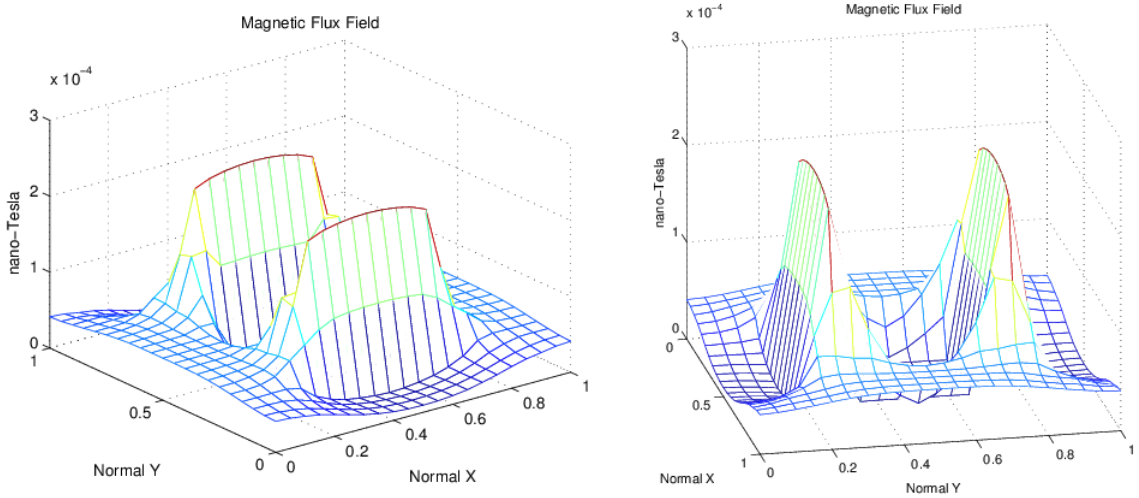


Figure 6.5: 3D Visualization of simulated object and magnetic signature.

6.2 Upscaling and Parameter Sensitivity

This section reviews the questions of scalability of the neural network model and sensitivity to variations in the environment on signature estimation. For a realistic practical implementation, it will be beneficial if the neural network signature estimator developed for one structure can be ported to another structure of a different size without having to go through the process of retraining. It is known that ambient magnetic field is not uniform on the surface of the earth. Thus it is necessary to understand the effects of variations of the earth's ambient field on signature estimation.

6.2.1 Model Scaling

This sections investigates the effects of changes in the size of the structure on signature estimation. It is assumed that a neural network has been trained for signature estimation using magnetic field data of a given structure. Then the fundamental question is whether the same neural network can be used for a different structure. This section provides an answer to this question based on the first principle.

Recalling Section 4.2 describing the working environment with 2-D geometry, the magnetic vector potential is represented by

$$-\nabla \cdot \left(\frac{1}{\mu} \nabla A \right) = J$$

which is equivalent to

$$-\frac{\partial}{\partial x} \left(\frac{1}{\mu} \frac{\partial A}{\partial x} \right) - \frac{\partial}{\partial y} \left(\frac{1}{\mu} \frac{\partial A}{\partial y} \right) = J \tag{6.1}$$

Let's also assume that the environment has a square geometry with

$$0 \leq x \leq a, \quad 0 \leq y \leq a.$$

Clearly the vector magnetic potential in the region can be computed using equation (6.1). This also implies that the neural network model of the environment will also correspond to equation (6.1).

Now suppose the vector magnetic potential need to be computed for a different environment within the domain

$$0 \leq x \leq b, \quad 0 \leq y \leq b.$$

Define new independent space variables as

$$\hat{x} = \frac{a}{b}x, \quad \hat{y} = \frac{a}{b}y \tag{6.2}$$

so that

$$0 \leq \hat{x} \leq a \quad 0 \leq \hat{y} \leq a.$$

Now the vector magnetic potential A can be written in terms of the normalized \hat{x} and \hat{y} coordinates using the basic relations

$$\frac{\partial A}{\partial x} = \frac{\partial A}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial x} = \frac{b}{a} \frac{\partial A}{\partial \hat{x}}$$

and

$$\frac{\partial}{\partial x} \left(\frac{1}{\mu} \frac{\partial A}{\partial x} \right) = \frac{b^2}{a^2} \frac{\partial}{\partial \hat{x}} \left(\frac{1}{\mu} \frac{\partial A}{\partial \hat{x}} \right)$$

Thus equation (6.1) applied to the new geometry simplifies to

$$-\frac{\partial}{\partial \hat{x}}\left(\frac{1}{\mu} \frac{\partial A}{\partial \hat{x}}\right) - \frac{\partial}{\partial \hat{y}}\left(\frac{1}{\mu} \frac{\partial A}{\partial \hat{y}}\right) = \frac{a^2}{b^2} J \quad (6.3)$$

Clearly the models (6.1) and (6.4) are same except for the right hand side. Note also that for estimation of magnetic signatures, we assume that the current J is zero so that the two models are fundamentally same and will produce the exact same results. Thus once the estimations are obtained, they must be interpreted using the coordinate transformation given by equation (6.2). This clearly demonstrates that the neural network trained for one structure can be ported to a different structure as long as spatial coordinates are scaled appropriately for interpretation of the results.

6.2.2 Variations in Material Properties

This section reviews the portability of neural estimator for structures constructed of material with a different magnetic permeability. Consider equation (6.4) for two different structures constructed of two different magnetic materials which will be denoted using the subscripts 1 and 2. This gives

$$-\frac{\partial}{\partial x}\left(\frac{1}{\mu_1} \frac{\partial A_1}{\partial x}\right) - \frac{\partial}{\partial y}\left(\frac{1}{\mu_1} \frac{\partial A_1}{\partial y}\right) = J \quad (6.4)$$

and

$$-\frac{\partial}{\partial x}\left(\frac{1}{\mu_2} \frac{\partial A_2}{\partial x}\right) - \frac{\partial}{\partial y}\left(\frac{1}{\mu_2} \frac{\partial A_2}{\partial y}\right) = J \quad (6.5)$$

Then it can be verified that the vector magnetic potential of the two systems are related by

$$A_2 = \frac{\mu_2}{\mu_1} A_1 \quad (6.6)$$

This shows that the corresponding magnetic flux densities are related by

$$B_2 = \frac{\mu_2}{\mu_1} B_1 \quad (6.7)$$

Thus the neural estimator trained for one structure can be used for another structure if the flux densities are multiplied by the ratio of their permeabilities.

6.2.3 Environmental Sensitivity and Noise

The integrity of sensor measurements is a required point of emphasis for successful estimation. Determining the effects of two environmental factors outside of the magnetic incident field are key here: field sensitivity within the shell, and magnetic noise. Sensitivity within the shell is a function of the shell itself; its permeability μ and thickness determine how much flux penetrate to the sensor location. As discussed in the previous section, its thickness is scalable, which leaves the sensitivity as a function of μ . It is important to note that the measure of sensitivity does not rely on the proportion between the internal and external field values, but on the effect that an external field change exerts on the internal. The overall objective is to be able to detect small changes in the external field. If large fluctuations in the external field derive small changes inside, the system will not work. From the linear solution in Section 4.3, knowing Laplace's equation is linear, and using linear μ values, the implication is that the field solution to the environment will be linear, at least in the unsaturated region of magnetization. It is reasonable to assume that the shell structure will be magnetically unsaturated since the earth's ambient field is weak. A simple simulated verification can be carried out in two iterations using two different magnetic vector potentials. The first uses the average magnitude, $50,000nT$, with the second uniformly increasing this value by $20dB$. The magnetic field results of these simulations, while monitoring a point within and a point outside the shell, reveal that

a $20dB$ change outside correlates to a $20dB$ inside. Recall the boundary conditions from section 4.3. For $A_o = 50,000nT$, a uniform field is generated. With the object placed in the field, we observe two points as shown in the figure. Here, $P1 = 5.6 \times 10^{-5}$ and $P2 = 3.968 \times 10^{-5}$.

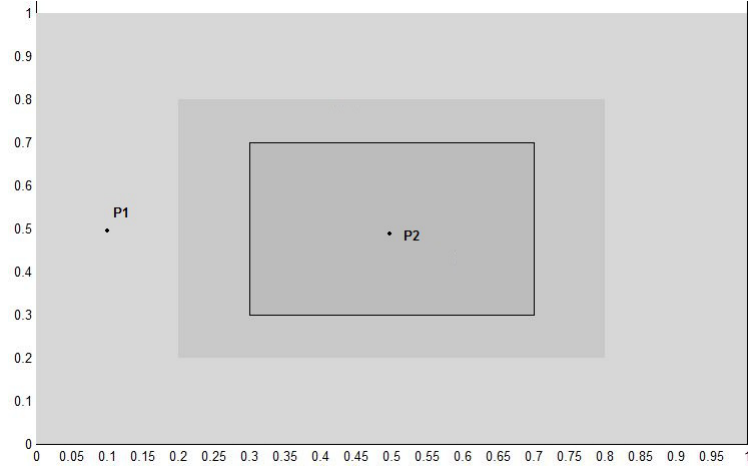


Figure 6.6: Preprocessing techniques for normalizing neural network input data.

Solving for a second iteration using $A_o = 55,000nT$, a change of 10%, we again observe $P1 = 6.2 \times 10^{-5}$ and $P2 = 4.36 \times 10^{-5}$. This reveals that a 10.7% change outside of the shell correlates to a 9.87% change within the shell. With relatively similar field changes and flux absorption taking place, the environment is deemed sensitive enough for successful data extraction.

The second environmental factor is magnetic noise. Introducing noise to the simulation is important because it is inevitable in practical applications, and because it prevents our environments from being too static. The sources of magnetic noise are various and complex: internal machinery, circuitry, free currents, and plate tectonics name only a few [2, 4]. Thus it is important in modeling to have an intimate knowledge of operational systems and surroundings of potential usage environments. Empirical results from other work show large influxes of noise to sensors, while noise mitigation and location optimization techniques reduce these occurrences to a signal to noise

ratio of approximately $20dB$. In our environments, this equates $\pm 5,000nT$ of the incident field, which is introduced by adding noise blocks to the system with small areas of varying μ .

Validating the simulation results can be done through comparison to other models, comparison to real world data, or performance in laboratory tests. The simulation was carried out due to real world data being unavailable, nor were laboratory materials to compliment the simulation parameters. The magnetic signature from chapter 3 was attempted to be estimated using the current network architecture, but without a shell object, the results were off as they tended towards mimicking the shells presence. However, modeling was previously available for spheres and prolate spheroids from [7, 8], in which a signature was modeled using a $20m$ radius with a $0.02m$ shell thickness with a relative permeability of 80 in Earths field. A PDETool representation of the model was configured, and resulted in an accurate signature model. The setup and results are depicted in figures 6.7 and 6.8.

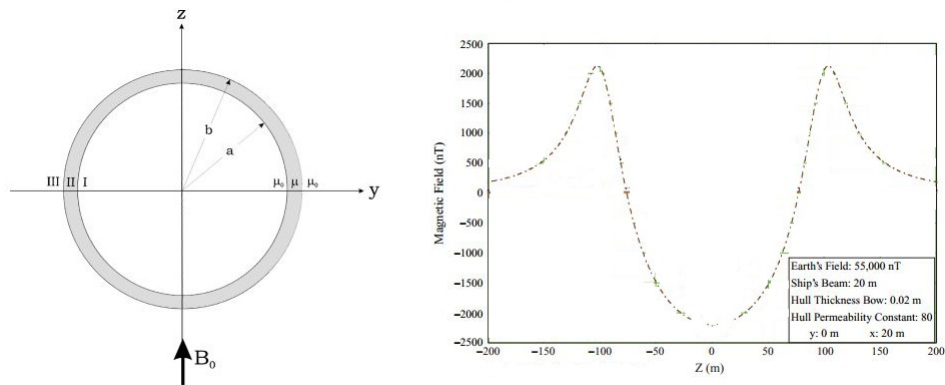


Figure 6.7: Geometric representation and first principle signature of a sphere.

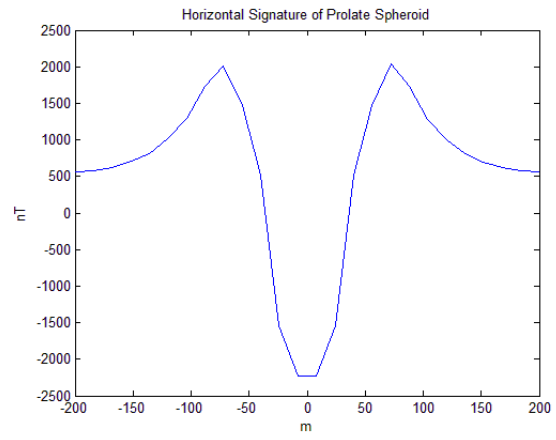


Figure 6.8: PDETool Magnetic Signature of the sphere from figure 6.7.

CHAPTER 7

CONCLUSION

7.1 Summary

A procedure has been created for modeling magnetic environments for use in neural network training. The purpose of the network is to utilize magnetic measurement data from local sensors to estimate magnetic field magnitudes in areas not readily accessible. Implementation of such a system provides real time determination of ferrite object magnetic signatures, which can be used against structures for detection and classification. Accurate estimation of signatures allows restricted usage plans to prevent structures from being detected, and can be instrumental to future work for developing active signature reduction systems. Results from both simulated training and experimental testing provide solid evidence about the systems effectiveness, which generated negligible simulation errors, with simulated training data subjected to a $20dB$ magnetic noise level. Thus the system can be expanded upon to implement and test more complex structural models, as this proof of concept was developed on a 2-dimensional rectangular shell. However, correlation between empirical and simulated systems has been provided through a mathematical framework, and deemed correct through the generated data.

Through the literature review it was apparent a neural network based estimation

system had not been considered for magnetic signature estimation, although results from other research suggested it was possible. A magnetic simulation environment was created and solved based around Poisson's Equation using finite element analysis in MATLAB's Partial Differential Equation toolbox. This allowed the setup for desired incident field magnitudes, object shapes, sizes, and properties, with the introduction of noise. The simulation results could be further analyzed to obtain the magnetic flux density of the environment. The simulation was used to collect data for neural network training. The Neural Network toolbox, also provided through MATLAB, was configured to create a network using seven inputs generated from magnetic flux data from within the rectangular shell. The input data would be trained to approximate the objects magnetic signature at locations chosen outside the shell, known as targets, selected at twenty-one external points. Temple University's High Performance Cluster aided in the intensive simulation and training procedures. After the network trained over an input range emulating Earth's average magnetic field magnitude, it was tested with a previously unknown seen data set. The simulation results concluded that the use of neural networks for magnetic signature estimation is a valid and accurate technique.

7.2 Suggestions for Future Research

The results obtained from this work deem further investigation of modeling and neural network estimation of magnetic signatures a worthwhile endeavor. From a modeling perspective, more complex and accurate representations of structures can be developed, which more closely behave like their physical implementations. Networks can be trained from the simulation data, and physical data should be obtained to test and verify the network. Improvements can be made to the network architecture, including sensor node optimization; however this is more prudent to be done on a

case by case basis, as different physical systems will be subjected to different types of interferences and incident fields. Further improvements can be investigated in using networks by modifying their activation functions to represent an objects hysteresis function. While the estimation ability of the model and network improve, it will also become possible to investigate magnetic signature control and reduction. A fully functional estimator will also lend itself to the completion of first order modeling of the unknown characteristics and behaviors of magnetization in objects. While the system still requires further verification empirically and in more complex geometries, initial results here reveal a promising technique for modeling and estimation of magnetic signatures.

BIBLIOGRAPHY

- [1] *Partial Differential Equation Toolbox User's Guide*. Natick, MA, 2011.
- [2] George I. Allen, Robert Matthews, and Michael Wynn. *Mitigation of Platform Generated Magnetic Noise Impressed on a Magnetic Sensor Mounted in an Autonomous Underwater Vehicle*. Accessed May 2010, IEEE Xplore.
- [3] Mark Hudson Beale, Martin T. Hagan, and Howard B. Demuth. *Neural Network Toolbox User's Guide*. Natick, MA, 2011.
- [4] J.T. Bono, D.J. Overway, and W.M. Wynn. Magnetic sensor operation onboard a uuv: Magnetic noise investigation using a total-field gradiometer.
- [5] Michael J. Caruso and Lucky S. Withanawasam. Vehicle detection and compass applications using amr magnetic sensors.
- [6] Simon Haykin. *Neural Networks and Learning Machines*. Pearson Education, Inc., Upper Saddle River, New Jersey, 2009.
- [7] John J. Holmes. *Exploitation of A Ship's Magnetic Field Signatures*. Morgan and Claypool, Denver, CO, 2006.
- [8] John J. Holmes. *Modeling of a Ship's Ferromagnetic Signatures*. Morgan and Claypool, Denver, CO, 2007.
- [9] Chang Seop Koh and Osama A. Mohammed. Detection of magnetic body using artificial neural network with modified simulated annealing. *IEEE Transactions on Magnetics*, 30(5):3644–3647, September 1994.
- [10] T.S. Low and Bi Chao. The use of finite elements and neural networks for the solution of inverse electromagnetic problems. *IEEE Transactions on Magnetics*, 28(5):2811–2813, September 1992.
- [11] James Clerk Maxwell. *A Treatise on Electricity and Magnetism, Vols. 1 and 2*. Cambridge University Press, New York, 2010.
- [12] International Association of Geomagnetism and Aeronomy. International geomagnetic reference field, 1990.
- [13] US Department of Interior. National geomagnetism program, December 2011. <http://geomag.usgs.gov/>.

- [14] Herivelton A. Oliveira, Fabio R. Barbosa, Otacilio M. Almeida, and Arthur P.S. Braga. A vehicle classification based on inductive loop detectors using artificial neural networks. 9th IEEE/IAS International Conference on Industry Applications, 2010.
- [15] Anastasis C. Polycarpou. *Introduction to the Finite Element Method in Electromagnetics*. Morgan and Claypool, Denver, CO, 2006.
- [16] Matthew N.O. Sadiku. *Elements of electromagnetics*. Oxford University Press, New York, 2001.
- [17] Jerry P. Selvaggi, Sheppard Salon, O.-Mun Kwon, M.V.K. Chari, and Mark DeBortoli. Computation of the external magnetic field, near-field or far-field, from a circular cylindrical magnetic source using toroidal functions. volume 43, pages 1153–1156, April 2007.
- [18] S Tumanski. *Handbook of Magnetic Measurements*. CRC Press, Boca Raton, Fl., 2011.