ABSTRACT

Principal Component and Neural Network Calibration of a Microwave Frequency Composition Measurement Sensor

Charles Stephen Maule, M.S.E.C.E.

Mentor: Robert J. Marks II, Ph.D.

Microwave sensors are becoming more prevalent throughout a variety of industries. While providing an effective form of measurement, microwave sensors are difficult to calibrate and provide results which can be difficult to interpret. An improved method for calibrating microwave sensors has been developed which transforms the waveform of a microwave spectrometer using principal component analysis and the results are used to train an artificial neural network to analyze a subject material. Broadband microwave spectrum calibration (BBMSC) is demonstrated using waveforms captured by a microwave spectrometer in a circular waveguide containing pulp stock slurry.

This thesis provides a review of the general applications of microwave sensors, details state-of-the-art calibration methods, as well as providing an introduction to principal component analysis and neural networks. The thesis continues by presenting the BBMSC method in detail, as well as how this method is applied to a set of waveforms of pulp-stock data and concludes with a discussion of the potency of BBMSC and recommendations for the future.

Principal Component and Neural Network Calibration of a Microwave Frequency Composition Measurement Sensor

by

Charles Stephen Maule, B.S.E.C.E.

A Thesis

Approved by the Department of Electrical and Computer Engineering

_____
Kwang Y. Lee, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of
Master of Science in Electrical and Computer Engineering

Approved by the Thesis Committee

_____
Robert J. Marks, Ph.D., Chairperson

_____
B. Randall Jean, Ph.D.

_____
John M. Davis, Ph.D.

Accepted by the Graduate School
December 2007

_____
J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN – Artificial neural network, a mathematical or computational model based upon biological neural networks

BBMSC – Broadband microwave spectrum calibration, the proposed method for calibrating a microwave sensor which involves using principal component analysis and artificial neural networks

CAD – Computer aided design, computer based tools used by engineers, architects and design professionals in the design process

PCA – Principal component analysis, a statistical tool for reducing a data set

TE – Transverse electric, a mode of electromagnetic propagation in which the electric component is zero in the direction of propagation

TM – Transverse magnetic, a mode of electromagnetic propagation in which the magnetic component is zero in the direction of propagation

RF – Radio frequency, encompasses electromagnetic waves which oscillate between 3Hz and 300GHz, which includes microwaves

VNA – Vector network analyzer, an instrument used to analyze the properties of electrical networks

CHAPTER ONE

Introduction

Increased automation in many industries has led to the need for sensors which provide effective and accurate measurement [1]. Microwave sensors provide a non-destructive and non-invasive method for measuring materials, without any of the health hazards that other methods may involve. These sensors are used for determining the dimensions, movement, moisture content and composition of a variety of materials. As microwave sensor components have continued to shrink, microwave sensors have become an even more attractive solution. Microwave sensors are have historically been used in determining moisture content but also are highly applicable to variety of industries including the forest industry and processing of wood and paper, the chemical industry and processing of chemicals and plastics, and the food industry, including processing of tobacco, butter and other food products [2]. These microwave sensors are used to determine the composition, dielectric, physical and other properties of the various materials.

Unfortunately, despite their growing success, microwave sensors "have not received the kind of acceptance for composition-measurement applications that, for instance, near-infrared instruments have seen" [3]. One reason for this lack of popularity of microwave sensors is that the calibration methods required for industrial use of these sensors are rigorous. Calibration requires that great care be taken when measuring the product with the sensor and the results are often difficult to interpret. The very design of the sensor often has a greater impact on the resulting waveform than does the makeup of

the material under measurement. These difficulties in actual use of microwave sensors has led to a number of novel solutions for microwave sensors, as well as a variety of calibration methods for those sensors.

Microwave sensors in the Pulp and Paper Industry

In the paper and pulp industry, microwave sensors, for many years, have been an attractive, although expensive, method for solving an increasingly difficult problem, the determination of pulp-stock consistency [4]. The opportunity exists for the introduction of an inline, low-cost microwave sensor and also for a lab-based microwave system that can be used to keep sensors of other types calibrated. Many of these sensors require routine recalibration to keep them operating properly. The recalibration operation currently used is both time consuming and expensive.

The determination of the consistency of the solution has been a difficult measurement for decades. Mechanical shear force sensors have dominated the market since the 1910's. In recent years, research has begun on electrical methods for determining consistency, including optical and γ-ray methods. Unfortunately both of these methods have some serious drawbacks. For optical sensors, applications are limited to conditions that have low consistency values, no mixture of pulp grades, or when pigments are not used [5]. Problems with γ-ray sensors include slow response, inaccuracy, and that they are limited to applications without pigment.

The increased need for accurate consistency measurement arises from the increased use of recycled fibers. This has increased the need for accurate measurements because mechanical shear force sensors are very sensitive to a number of variables including pulp grade, wood species, fiber length distribution, and freeness, which are

unknown factors when working with recycled fibers [5]. Another recent problem has been an increased use of mineral pigments, which disturb optical and radioactive sensors and are ignored by mechanical sensors. A third reason for the need for accurate consistency measurement is the increase in the variety of paper grades being used in the paper and pulp industry. With this increase comes the need to be able to accurately measure each grade as mills may require frequent changes from one grade to the next.

The need for a simple and straightforward calibration method was the motivation for this research. The goal was to achieve a methodology that provided accurate results for the microwave sensors and revealed how well the instrument had been calibrated. The method presented in the rest of the thesis is a simple and accurate method, which requires no expert knowledge of the waveform captured by the sensor.

CHAPTER TWO

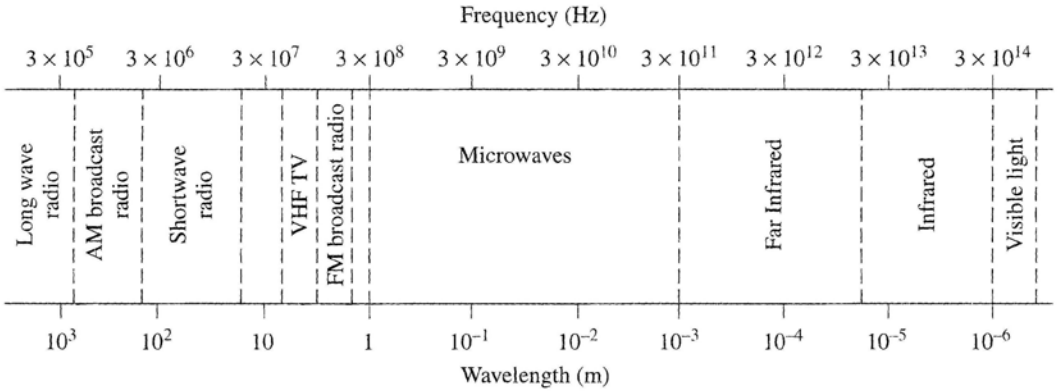
Introduction to Microwave Engineering

The electromagnetic spectrum is the range of electromagnetic radiation, waves with electrical and magnetic components, which are subdivided based upon frequency, ranging from DC (zero frequency) to cosmic rays. Microwaves are a subsection of the electromagnetic spectrum, consisting of alternating current signals with frequencies between 300 MHz (3×10^8 Hz) and 300 GHz (3×10^{11}), with wavelengths ranging between $\lambda = 1$ m and $\lambda = 1$ mm, respectively, as shown in Figure 1 [2]. The IEEE standard 100 and IEC standard 60050 modify that definition slightly by starting microwave frequencies at 1 GHz. Microwave frequencies are unique in the spectrum for a number of reasons including line-of-sight communications and higher possible bandwidth than other RF signals which is ideal for information transfer. Microwaves are also capable of interacting with materials on a molecular, atomic and nuclear level, which is ideal for a variety of applications, including remote sensing and medical diagnostics and treatment [2]. The latter fundamental property of microwaves is what makes this particular portion of the spectrum particularly useful in sensor devices.

Waveguides

Waveguides have been used for the transmission of microwave power since 1936. Lord Walter Raleigh proved that electromagnetic transmission was possible within a waveguide in 1897. Unfortunately, he never made an experimental verification of his theory, which also included the modes of TE and TM types and the cutoff frequency.

Waveguides come in a variety of sizes and shapes: rectangular, circular, coaxial lines, as well as several planar forms such as stripline, microstrip, slotline, and coplanar waveguides. Typically waveguides are used for transferring power or transmitting communication signals. However, while this is their primary application, waveguides can also be exploited by examining the attenuation and distortion characteristics for frequencies below the cutoff frequency and through the passband region of the waveguide [3].



Typical Frequencies	
AM broadcast band	535–1605 kHz
Short wave radio band	3–30 MHz
FM broadcast band	88–108 MHz
VHF TV (2–4)	54–72 MHz
VHF TV (5–6)	76–88 MHz
UHF TV (7–13)	174–216 MHz
UHF TV (14–83)	470–890 MHz
US cellular telephone	824–849 MHz
	869–894 MHz
European GSM cellular	880–915 MHz
	925–960 MHz
GPS	1575.42 MHz
	1227.60 MHz
Microwave ovens	2.45 GHz
US DBS	11.7–12.5 GHz
US ISM bands	902–928 MHz
	2.400–2.484 GHz
	5.725–5.850 GHz
US UWB radio	3.1–10.6 GHz

Approximate Band Designations	
Medium frequency	300 kHz to 3 MHz
High frequency (HF)	3 MHz to 30 MHz
Very high frequency (VHF)	30 MHz to 300 MHz
Ultra high frequency (UHF)	300 MHz to 3 GHz
L band	1–2 GHz
S band	2–4 GHz
C band	4–8 GHz
X band	8–12 GHz
Ku band	12–18 GHz
K band	18–26 GHz
Ka band	26–40 GHz
U band	40–60 GHz
V band	50-75 GHz
E band	60-90 GHz
W band	75-110 GHz
F band	90-140 GHz

Fig. 1: The electromagnetic spectrum [6]. Microwaves are defined as electromagnetic waves with frequencies ranging form 300MHz to 300GHz.

The scenario considered here can be modeled as a waveguide filled with a lossy dielectric material. The propagation constant of a circular waveguide for the TE_{nm} mode is given by

$$k_z = \sqrt{k^2 - \left(\frac{p'_{nm}}{a}\right)^2} = \sqrt{\omega^2 \mu \varepsilon' - \left(\frac{p'_{nm}}{a}\right)^2 - j\omega^2 \mu \varepsilon''}$$

where ω is the radian frequency, ε' and ε'' are the real and imaginary parts of the complex permittivity, μ is the magnetic permeability, and a is the radius of the waveguide. Several values for p_{nm} for the first few TE and TM modes are shown in tables 1 and 2. For a lossy dielectric, the real part of β_{nm}, which is the phase constant of the microwave, will never go to zero, thus the cutoff frequency for the waveguide is not realized. The imaginary part of β_{nm} is the attenuation constant for the propagating microwave. A dramatic decrease in the amplitude of the signal below a certain frequency still reveals the cutoff frequency (see Fig. 3).

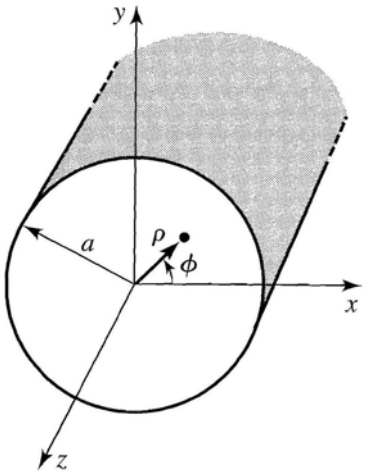


Fig. 2: An example of the geometry of a circular waveguide [6].

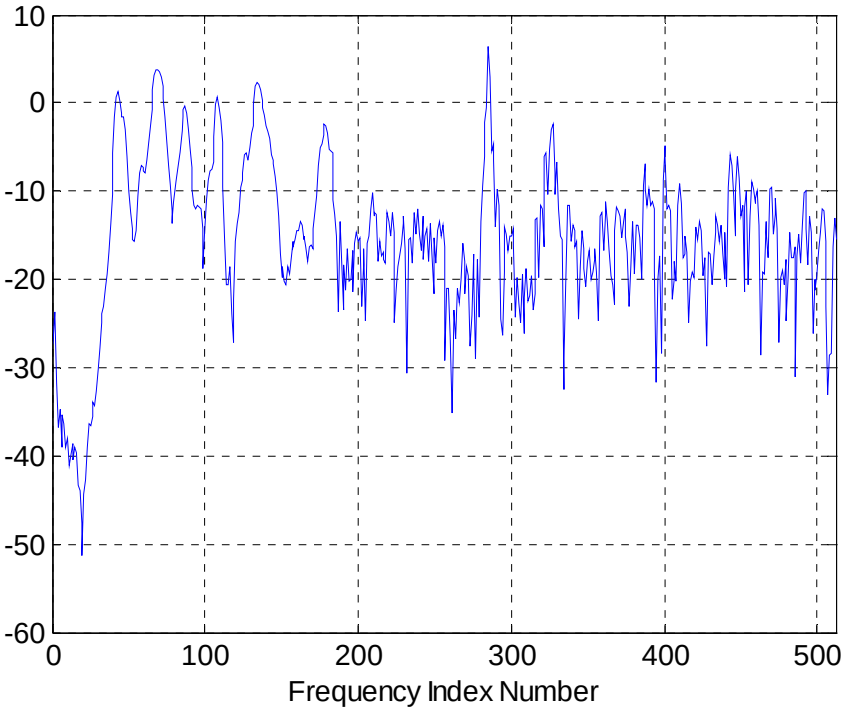


Fig. 3: A log-magnitude plot of a waveform captured from a circular waveguide. Notice the attenuation below the 40th frequency index (the cutoff frequency for this chamber), where the microwave does not propagate.

The prepared samples were contained within a circular pipe measuring approximately 7.62 cm in diameter by 30.38 cm in length. This piping acts as a circular waveguide, for transmitting the microwaves, which are distorted based upon the electrical properties of the slurry solution. Based upon the dimensions of the waveguide, the cutoff frequencies of the various modes can be determined using the permittivity of the material filling the chamber (here, mostly water) and the roots of the Bessel functions. For the TE_{11} with water with dielectric constant 76, the cutoff frequency would be around 265 MHz. An example of log magnitude of the waveform collected by the microwave spectrometer, is revealing the cutoff frequencies for several modes, is shown in Fig. 3. Circular waveguides, which are made up of a single conductor are only able to support TE and TM modes. The first few modes for both TE and TM modes of the device follow in

Tables 1 and 2 and Table 3 contains the first 10 cutoff frequencies for the waveguide and their mode. Table 3 contains the cutoff frequencies and modes of the first 10 cutoff frequencies of the circular waveguide. For TE modes, the cutoff frequency can be determined using:

$$f_c = c\frac{k}{2\pi\sqrt{\mu\varepsilon}} = \frac{p'_{nm}}{2\pi a\sqrt{\mu\varepsilon}} = \frac{p'_{nm}c}{2\pi a\sqrt{\varepsilon_r}}.$$

For TM modes, the cutoff frequency can be determined using:

$$f_c = c\frac{k}{2\pi\sqrt{\mu\varepsilon}} = \frac{p_{nm}}{2\pi a\sqrt{\mu\varepsilon}} = \frac{p_{nm}c}{2\pi a\sqrt{\varepsilon_r}}.$$

Table 1: Values of p'_{nm} for TE Modes of a Circular Waveguide [6].

n	p'_{n1}	p'_{n2}	p'_{n3}
0	3.832	7.016	10.174
1	1.841	5.331	8.536
2	3.054	6.706	9.970

Table 2: Values of p_{nm} for TM Modes of a Circular Waveguide [6].

n	p'_{n1}	p'_{n2}	p'_{n3}
0	2.405	5.520	8.654
1	3.832	7.016	10.174
2	5.135	8.417	11.620

Table 3: Cutoff Frequencies for a circular waveguide with radius 3.81cm (a = .0381) and filled with water with dielectric constant 76.

Mode	Cutoff Frequency (MHz)
TE_{11}	264.5
TM_{01}	345.7
TE_{21}	439.0
TE_{01}	550.9
TM_{11}	550.9
TE_{31}	603.9
TM_{21}	738.2
TE_{41}	764.2
TE_{12}	766.3
TM_{02}	793.5

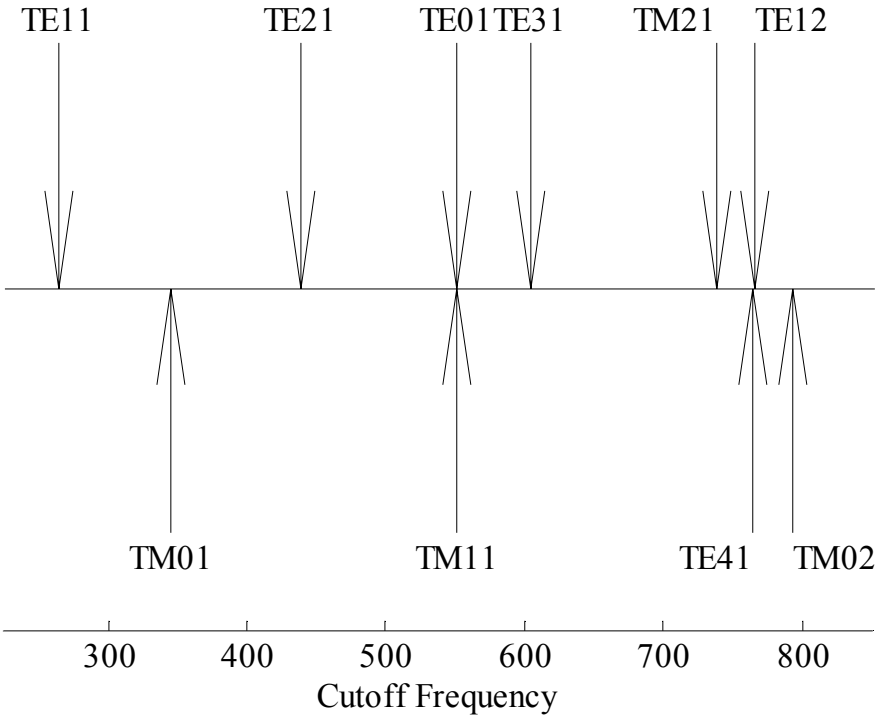


Fig. 4: Cutoff frequencies for the circular waveguide with radius of 3.81cm (adapted from [6]).

CHAPTER THREE

Microwave Sensors and Calibration

Composition determination is a typical application for microwave sensors. In these applications, the sensor is used to determine the relative concentration of the known contents of a solution based upon changes in the permittivity [3]. In theory, a microwave sensor is used to determine the permittivity of the material but, in practice, the permittivity is neither computed or a desired result of the measurement. The calibration of microwave sensors is often done using something more that is more easily determined, such as the amplitude or phase of the waveform at one or multiple frequencies [3].

A variety of methods for calibrating microwave sensors exist. The direct approach works well for simple situations, where the electrical properties of the material are known. Another method involves determining the lumped-element equivalent of the sensor. Once this is known, the theoretical output of the sensor is compared with the measured output and corrections are made to the sensor. This method produces accurate results; unfortunately it is also a complex, time-consuming and expensive process.

An earlier method for increasing efficiency by decreasing data dimensionality when determining consistency can be found in [3]. This method exploits the changing properties of the waveform, such as attenuation and distortion, above and below the cutoff frequency of the waveguide chamber. Microwave spectral data from selected regions of the waveform are reduced to line segments. The captured waveform is visually inspected for regions where significant shift occurs. These shifts ideally correspond with known changes in the properties of the substance. The selected regions

are then reduced to line segments (slopes and offsets) for characterization. The slope and offset parameters are determined by fitting a best-fit (least squares) line segment to specific regions of the waveform, including the region below the cutoff frequency and several regions within the passband of the circular waveguide [3]. These parameters are simple to determine and provide useful information about the shape of the waveform.

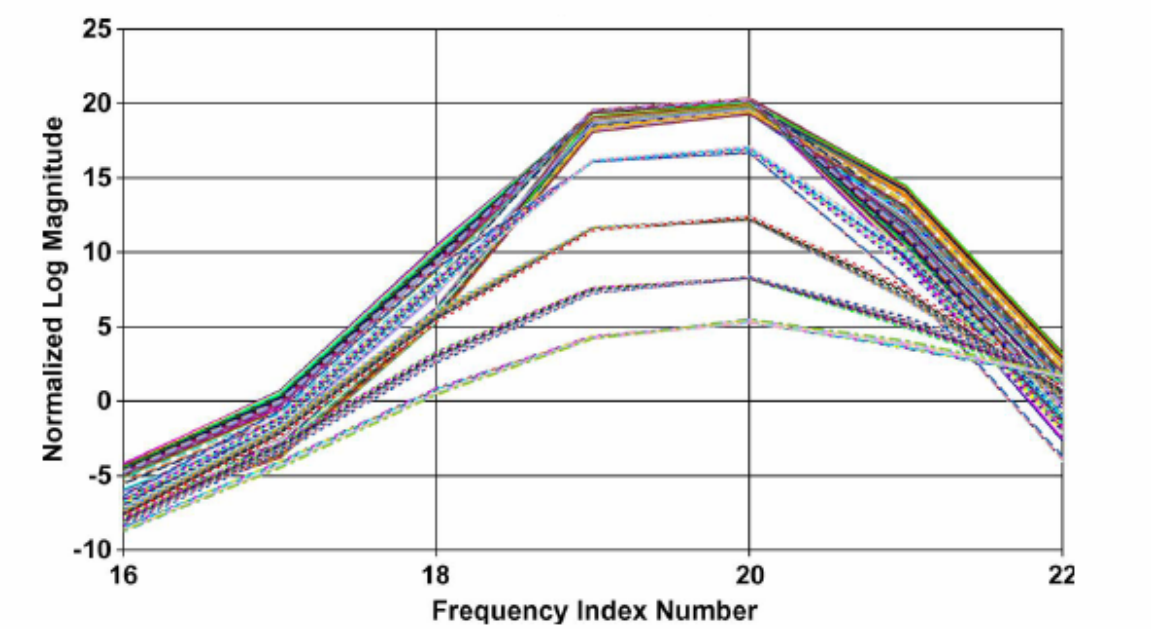


Fig. 5: A visually selected region of a microwave spectrometry waveform. Such regions are selected for the variation in the signals, which ideally corresponds with a known change in the properties of the solution [7].

In theory, the frequency response of a material's permittivity is a relatively smooth curve, and various regions of the response characteristic of the waveguide will be indicative of such smooth changes. However, like the waveform shown in Fig. 3, a variety of practical issues, such as impedance mismatch of cables, coupling loops and connectors, unintended resonances and errors in the electronic measurement devices, interfere with the signal to produce the more chaotic curve. Thus it would seem that the desired information is stored in the general curve of the signal. The line segments

provide a general shape for the curve and thus also contain the information inherent in the waveform. An algorithm was also developed to attempt to fit the waveform data to an ideal curve, as an alternative to simple line-segments, but this proved to be a problem for the limited processing power of the equipment used. The strength of this line segment method over the curve fitting algorithm is in the speed at which the line segments are determined, as well as providing more efficient data capture and processing.

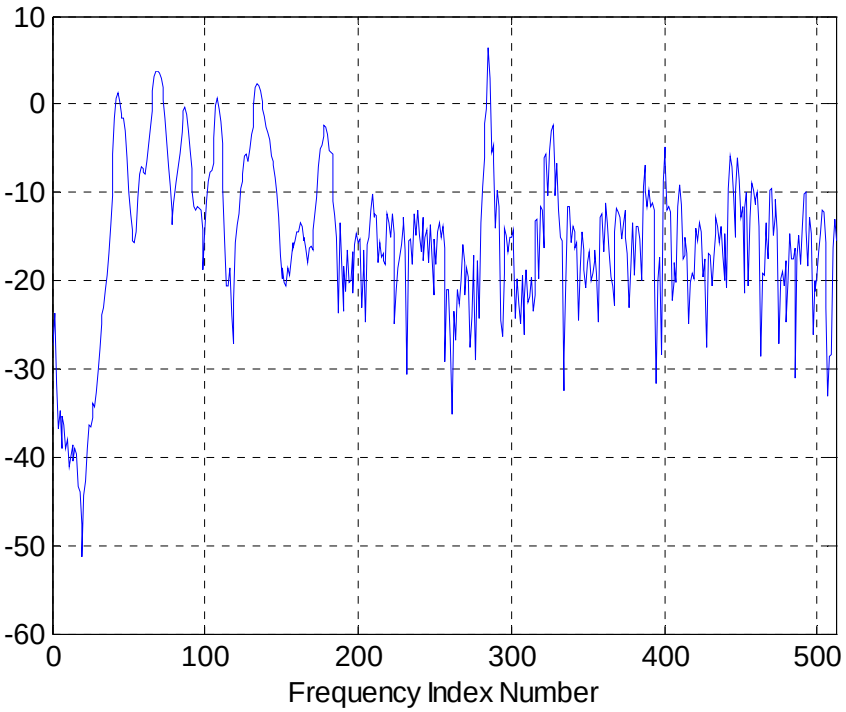


Fig. 6: A sample waveform from the microwave spectrometer, with 512 frequency indices. Data was collected from a circular waveguide, as shown by signal increases in magnitude at the cutoff frequency, around the 25th frequency index. Also note the frequency response is not a smooth curve.

The line-segment method is demonstrated in a variety of scenarios including determining the temperature of tap water, predicting the amount of fat, protein and water in ground meat, and predicting the percentage of diphenyl oxide in a solution. One thing that these examples reveal is that in samples where water is a significant percentage, it is

often necessary to determine the temperature of the solution, due to the changing dielectric properties of water with temperature. A final example of the line-segment method is the determination of consistency and conductivity of pulp-stock mixture. This example shows that the sensor was able to accurately determine the consistency of the solution, when the conductivity was known. This phenomenon requires that the determination of consistency be a two-step calibration process, but produces good results.

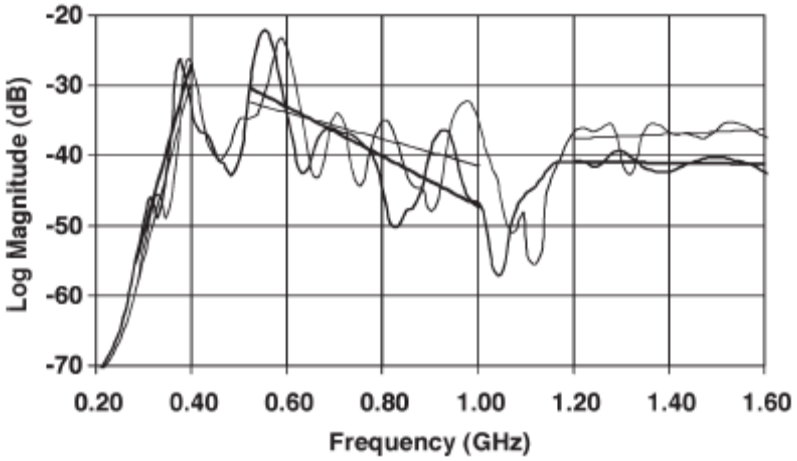


Fig. 7: Frequency response of tap water [3]. This figure shows the type of waveform that is captured by a microwave spectrometer and the line segments used by Jean[3] to characterize the waveform as it changes with water temperature.

While the line-segment calibration method provided efficient data capture and data processing, it also has several drawbacks. This method requires that the waveform be inspected for specific regions for variation due to the changing properties of the solution. This requires a personal judgment call about what sections appeared most varied and also discards regions of the waveform. By discarding sections of the waveform, useful information may also be discarded. Another problem with this method is that it is possible for no specific region of the waveform to exhibit clear variations that correlated with the change in the material's properties.

Another solution to the on the pulp-stock problem makes use of neural network for determining the pertinent information from the waveform [7]. Rather than reducing the complexity of the waveform through line-fitting techniques, Green, et al. introduce a neural network, a mathematical structure which is efficient at determining complex patterns in data sets. The first 100 frequency points are used as inputs for the neural network. This contrasts with the line-segment method in that there are no selected regions used in determination, but that the whole waveform is used as information to inform the neural network. Two separate networks determine the solution's characteristics: one neural network to determine the conductivity and the other to determine the consistency. The two networks use the same waveform as the input and also have the same architecture, using three hidden layers, with 101, 10 and 2 nodes in the three hidden layers, with logistic functions for the hidden layers and a linear function for the final output layer. They achieve excellent and accurate results with both of their neural networks, improving significantly on Jean's line segment technique.

The method presented by [7] improves on the line-segment method in several ways. By using the entire waveform as an input to the neural network, there is no need for a judgment call about which portions of the waveform appear to exhibit variations based upon changing properties. The neural network also is capable of determining complex patterns from large data sets and thus should extract a pattern between the changing waveform and the properties of the pulp-stock solution. Another advantage of the neural network approach is that this method is able to determine the consistency of the solution without prior knowledge of the conductivity, which was a limitation for the line-segment approach.

Broadband microwave spectrometry attempts to retain the strengths of the line-segment and the neural network approach. The line-segment method provides efficient data capture and data analysis by reducing the waveform to a few simple line segments and is able to produce accurate results for both consistency and conductivity, albeit in a two-step process. A reduced portion of the original waveform is used without line-fitting by [7], but includes the use of a neural network to extract the useful information from the signal. The goal of BBMSC is to provide the efficiency of the line-segment approach, while combining the pattern recognition abilities of a neural network to extract the important parts of the waveform. Another goal is to achieve those results in a straight-forward and simple manner in order to provide an efficient calibration method for microwave sensors. The next chapter details the methods used to achieve that efficiency and how BBMSC incorporates the neural network.

CHAPTER FOUR

Principal Component Analysis and Artificial Neural Networks

Principal Component Analysis

Broadband microwave spectrometry calibration is the improved method for microwave sensor calibration. This section details the most important parts of the technique, principal component analysis and artificial neural networks, and explains how they function and the strengths of these analysis methods.

Principal component analysis (PCA) is a statistical analysis technique which reduces the complexity of a multi-dimensional data set to fewer dimensions, while retaining the most important information [8]. PCA involves the computation of the eigenvalue decomposition, or the singular value decomposition, after mean centering the data for each attribute. A key feature of PCA algorithms is that the user is able to specify the amount of information that is retained: to be more efficient by retaining less information or less efficient to retain as much information as possible. PCA retains the aspects of the data set that contribute most to the variance, which is ideally the most important part of the data set. In doing so the amount of redundant information in the signal is reduced as well as reducing the dimensions of the data set significantly.

More specifically, PCA is an orthogonal linear transformation which transfers the data to a new coordinate system. In doing so, the greatest variance in the data is placed on the first coordinate (also called the first principal component), the second greatest variance is placed on the second coordinate, etc. In least square terms, PCA is the

optimum transform for a data set. PCA is also considered an excellent tool in pattern recognition.

PCA is used extensively in feature reduction problems and in analyzing multi-dimensional data sets. Lam et al. have shown PCA to be the most effective of four feature reduction methods available for text characterization, and make use of both PCA and neural networks in their comparisons [8]. PCA and neural networks are also used in conjunction in a variety of situations including online fuel tracking and classification of fuzzy data [10][11].

Additionally, PCA has been used in a variety of microwave sensing applications. Applications include the determination of food properties such as water content, the amount of water added to food products and the percentages of constituents [12]. PCA has also been used as an optimization tool when collecting data over large frequency regions and with cheaper and less accurate equipment [13]. Development of a low-cost composition measurement microwave instrument using a set of discrete frequencies has also been reported, which uses PCA in reducing the spectra captured by the device [14].

Artificial Neural Networks

Artificial neural networks are the second significant component of the BBMSC technique. Artificial neural networks were inspired by the biological central nervous system and the manner in which neurons function. An artificial neural network is

> an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation. . . In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. [15].

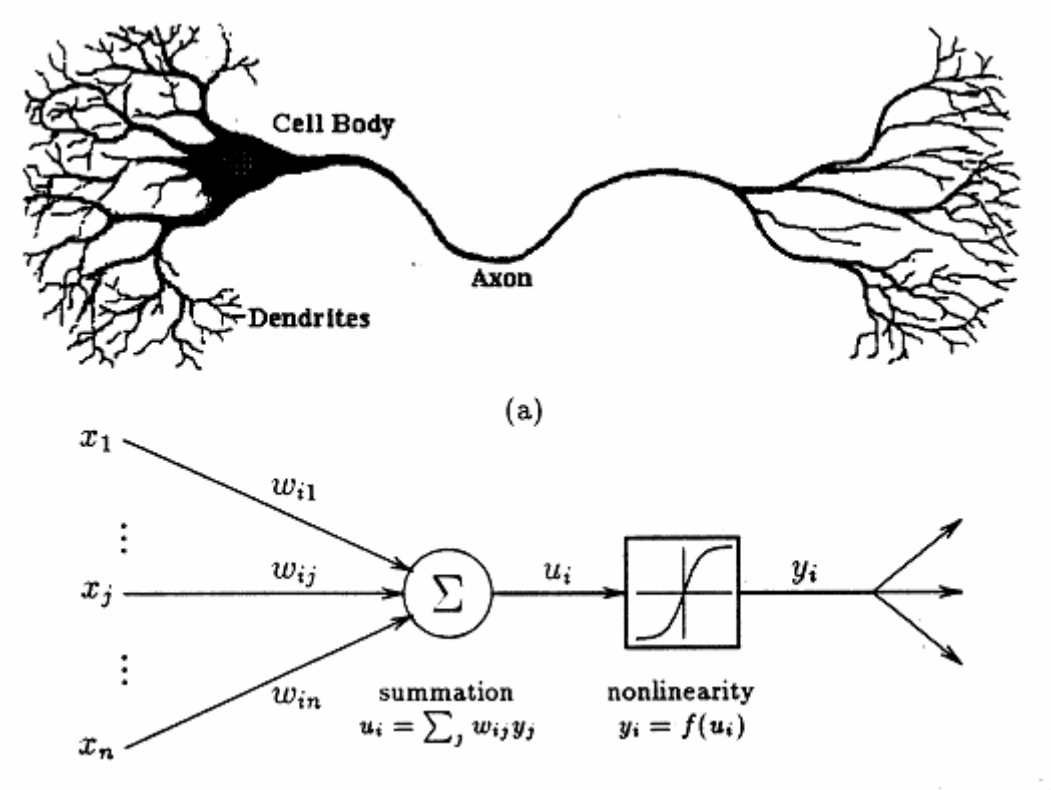


Fig. 8: An example of a biological neuron (a) which collects information from nearby neurons, the cell body which processes the information, and the axon which transmits the signal to other neurons; An artificial neuron model (b), which parallels the design of a biological neuron [16].

Layered perceptron neural networks are made up of a series of layers, typically three or more [17]. The input layer is the input data for the network and is typically not included when counting the number of layers within the network. The output layer is the output data from the network. Between the input and output layer are the hidden layers. The number of hidden layers varies based upon the application for which the network is being trained. Fewer hidden layers results in reduced training times, but may also limit the ability of the network to replicate the desired function. Each hidden layer is made up of an arbitrary number of nodes. The output of a node is dependant upon the information received from other nodes and the weight associated by the connection to those nodes.

The weighted information is then passed to the next layer of nodes, and so on, until it reaches the output layer. A single node is capable of generating only a scalar output, but many nodes in combination are capable of generating complex and non-linear functions.

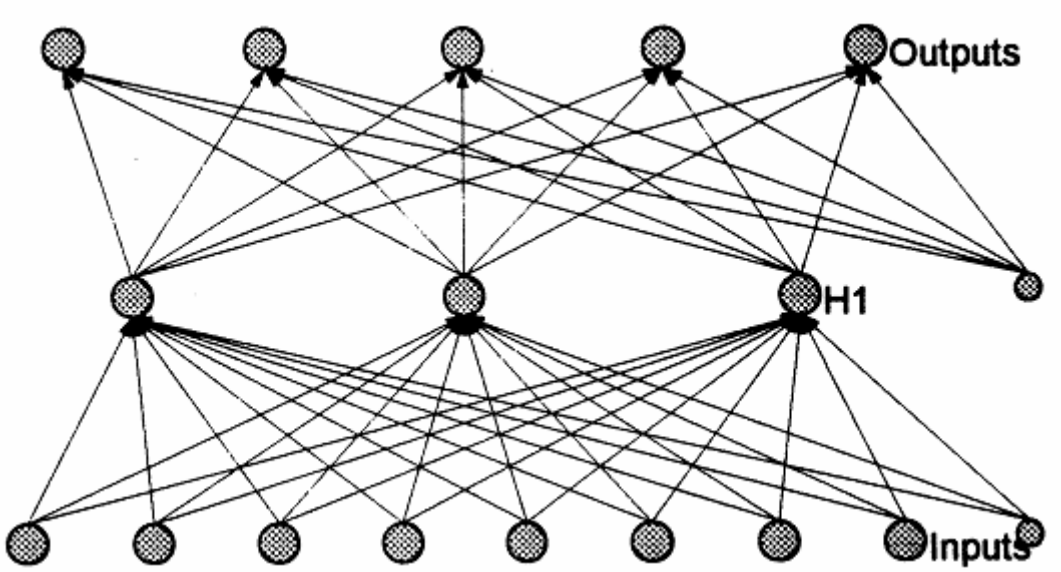


Fig. 9: A layered feed-forward artificial neural network. Includes input layers, one or more hidden layers (H1) and an output layer [16].

There are several learning paradigms associated with neural network training: supervised learning, unsupervised learning and reinforcement learning. For this application, supervised learning was most appropriate. In a supervised learning situation, data is presented to the network in an effort to determine a relationship between the inputs and outputs. The desire with this form of learning is to minimize the amount of error between the network mapping and the data. Typically a mean-squared error is used to minimize the average error between the neural network output and the target data. This form of learning is typically used for pattern recognition and regression (or function approximation) [15].

To achieve the best results, the network generates an arbitrary weight and connection for each node in the hidden layers of the network. The network is trained by varying the weights and the connections between nodes, through a series of iterations, until the network is able to achieve more accurate results. These results are determined by comparing the neural network outputs with the target values. The longer the network trains, the more accurate the results for the training data. The network can also be exposed to test data. The test data is not used to change the weights and connections of the network but allows the user to observe the error margins for a second set of data. The test data, if it is similar to the training data, should also experience diminishing error values between the target values and the neural network outputs. Extended training times can lead the network to reproduce the function of the training data too closely and thus produce less accurate results for the test data. Ideally the network architecture, both weights and connections, should be retained when the training data has achieved acceptable results and before overexposure to the training data.

The number of hidden layers and nodes in each layer requires some experimentation [18]. With too few layers and nodes, the network may have trouble approximating the desired function, while with too many parameters the ANN may perform well on the training set and still generate errors on other sets of data. This reinforces the need to test the network with both testing and validation data. Testing data provides a glimpse of how the network will perform on other data and the verification data, which is used after an architecture has been determined, shows the actual performance of the network on new data. This process of using information that the

network has not seen before is known as cross validation and is a standard test technique for evaluating a network's performance [16].

In this situation, the data available included both the input (the waveform captured by the microwave spectrometer) and the desired outputs (the conductivity and the consistency of the material inside the waveguide). The goal was also to determine an accurate mapping between the input data and the output data, which became a pattern recognition problem, making the problem ideal for a supervised learning situation.

The strength of neural networks lies in their ability to infer a complex and non-linear functions based on a set of observations. More specifically, a neural networks' ability for feature recognition has laid the ground work for their introduction as tools in the detection of dielectric objects or determining dielectric properties of an object [19]. Neural networks have also been proposed as tools to determine sample properties in the papermaking industry by classifying samples more quickly than by standard methods [20]. Instrument response time is an important factor in improving efficiency.

ANNs have also been used in several microwave applications. ANNs are being used in computer-aided design (CAD) programs for modeling passive and active microwave components as well as microwave circuit designs [21]. They are being used to simulate a variety of different microwave components based upon measured values. Neural networks are also being used for simulation and optimization of a variety of different structures, because they are more efficient and accurate than other models, allow for more dimensions than traditional lookup tables and are easy to develop for new technologies. ANNs are being used in vector network analysis (VNA) [22], as well as to model several calibrations and to determine the permittivity of liquids and the moisture

content of wheat. Neural networks provide several advantages over other methods, which include numerical modeling, analytical methods and empirical modeling [23]. Neural networks provide quicker results, are easier to develop and are only limited by the data provided to them.

QwikNet

QwikNet, a commercial software package for artificial neural network development, was used to generate and test the neural network for this thesis [16]. QwikNet implements a feed-forward multi-layer perceptron, meaning data is passed forward through the network from the inputs to the outputs. QwikNet allows for the specification of up to 5 hidden layers, with several options on the layer's activation function: logistic, tanh, linear and Gaussian. Each layer has an arbitrary number of nodes, which can be specified by the user. Associated with each node is a weight, which can be positive or negative, and varies through the training process to determine what set of weights which produces the lowest RMS error between the output data and the target data. When an ideal set of weights is determined (a set where the training and testing average RMS error is minimized), the set of weights can be saved for future use. A number of other options are available to the user as well to further customize the neural network, including the training algorithm, weight minimum and maximum, stopping criteria for training and training properties.

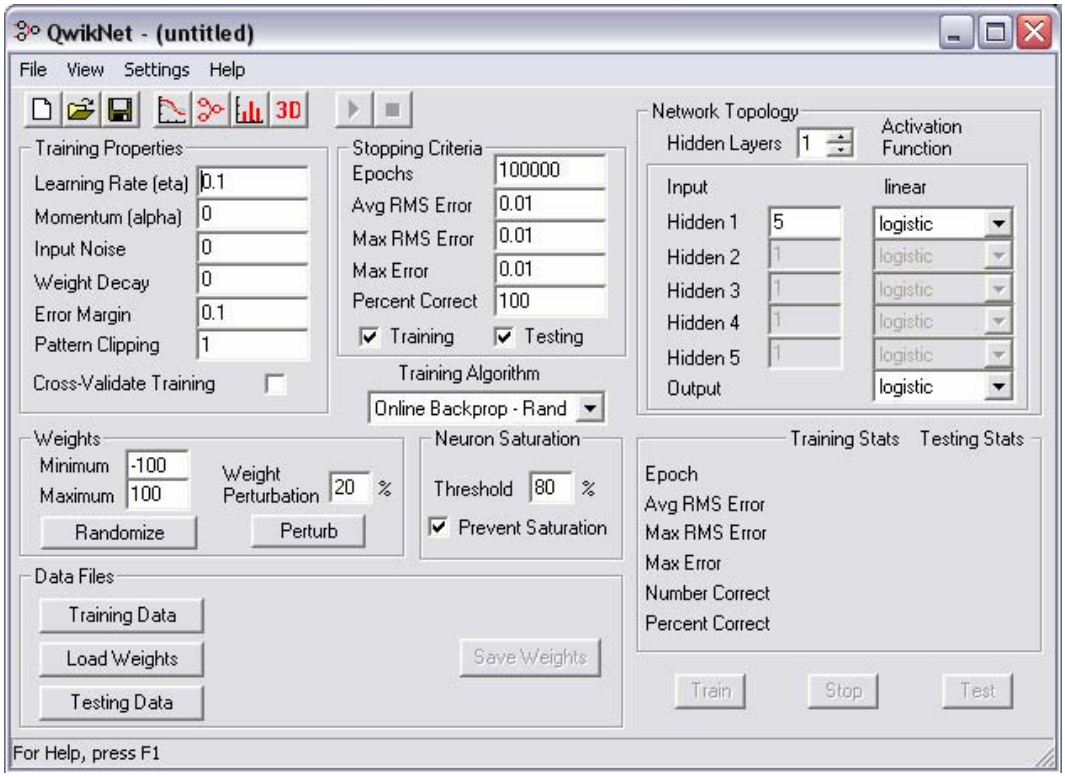


Fig. 10: A screenshot of the QwikNet neural network software [16].

CHAPTER FIVE

Broadband Microwave Spectrometry Calibration

Data used to illustrate neural network microwave sensor calibration are from prepared samples of pulp stock typically used in paper production. Samples were prepared at several discrete values across a range of consistencies and conductivities and spectrometry measurements were obtained using a prototype lab-based microwave sensor. The microwave signal is transmitted to the pulp sample by coupling loops arranged on one side of the chamber, mounted in ceramic dielectric seals, and spaced about 10 cm apart. Data consists of samples of the log magnitude spectrum from the microwave sensor at 512 frequencies indices. The temperature of each data sample was also recorded at the time of microwave sensor reading. While the log magnitude is recorded for 512 frequencies, only the first 150 samples are used for calibration due to a decrease in the signal to noise ratio beyond the 150^{th} frequency index.

The microwave signal is produced by ultra wideband pulses having an effective bandwidth of approximately 850 MHz and a pulse repetition of approximately 4 MHz. The dispersed output pulses are sampled with an extended time-sampling system and Fourier transformed to produce the output spectra.

Data Processing

The microwave sensor waveforms are stored in a proprietary format, which requires that they be exported to Excel and, from there, read into MATLAB, where they are stored in an m-file for further manipulation. MATLAB is a high-level language

development environment that is capable of executing computationally intensive tasks more efficiently than other languages, such as C or C++ [24]. Further manipulation of the data was done using algorithms developed for repeated use in MATLAB.

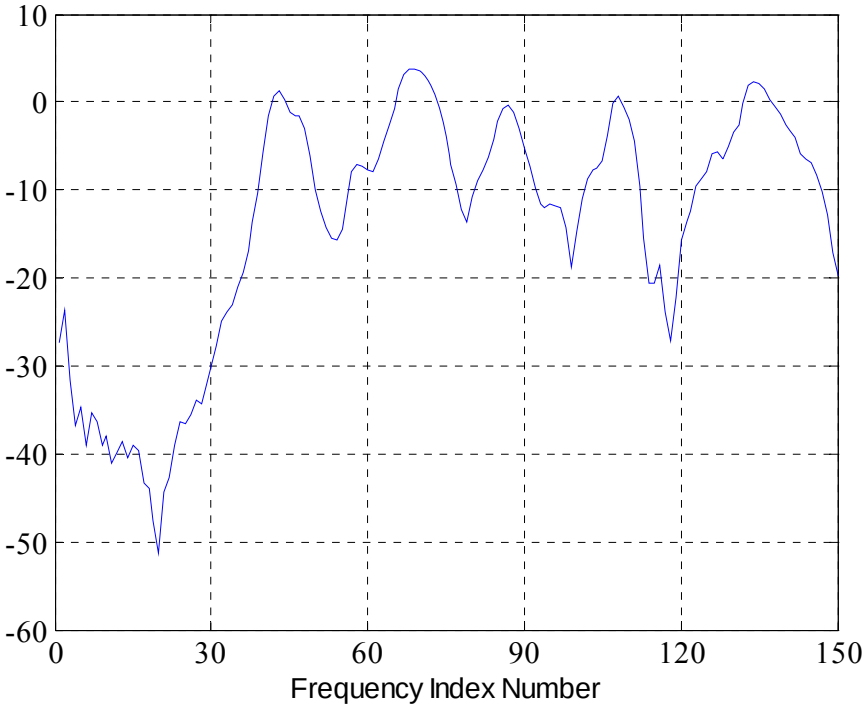


Fig. 11: An example of the information used to train the network. The first 150 frequency indices of a microwave spectrometry data for a solution with conductivity of 700 µS/cm and consistency of 1.25%, at 22.817°C.

The microwave data are stored in an array, where each column is a single captured waveform and each row contains a frequency index for all of the captured waveforms. The original array was 150 x 2170, with 150 frequency index points and 2170 total waveforms. The PCA algorithm is then used to reduce the data set, from the first 150 frequency indices down to the first 11 principal components (see Appendix A for the algorithm used and Appendix B for the MATLAB implementation of that algorithm). The PCA algorithm was specified to retain 99% of the information, while

also reducing the number of index points per observation by 92.67%. The 11 principal components, along with the temperature of the solution when the waveform was captured, are the inputs to the neural network.

In order to format the information for the neural network, another MATLAB algorithm was developed for quick and efficient creation of the input files for QwikNet (see Appendix C). This segment of code parses the data into the neural network inputs (the microwave spectra) and target information (the known consistency and conductivity values of that waveform). It also produces three separate files, which are used for the training, testing and validation of the neural network. The necessity for three separate files was already discussed in Chapter 4. The QwikNet file creation algorithm also ensures that each document contains all possible combinations of consistency and conductivity values, in order to guarantee that the neural network is exposed to the full spectrum of input principal components in each phase of the training process. The training file contains the 1^{st}, 4^{th}, 7^{th}, etc. sensor observations, the testing file contain the 2^{nd}, 5^{th}, 8^{th}, etc. observations and the verification file contains the 3^{rd}, 6^{th}, 9^{th}, etc. observations from each set of consistency and conductivity pairings. The separation of observations in a non-sequential fashion is done in order to provide a wide range of possible temperature values for each document, in the case that the waveforms were captured consecutively. By separating waveforms with similar temperature into each of the three files, the neural network is exposed to a greater range of temperature values in training and results in more accurate results in the verification process.

Neural Network Training

The neural network training process involved experimenting with the architecture of the neural network. Each architecture was replicated in several windows of QwikNet. The QwikNet software provides the average RMS error, the maximum RMS error, the maximum error, the number of correct outputs and the percent of outputs that are correct for each epoch (an epoch is a single pass through the training information). The best set of weights and connections from each ANN structure was selected by comparing the average RMS and max RMS at the $10{,}000^{th}$ epoch across a variety of structures. After extensive experimentation, a structure emerged as most accurate. Once this structure was determined, it was replicated in several instances of QwikNet for further training. Small variations in the training weights led to some neural network weights providing better results. The best results at each step were kept and trained further. This series of iterations led to the final network weights and connections.

The final architecture for the ANN contains three hidden layers with 22, 15, and 8 nodes respectively. The activation function for the hidden layers and the output layer is logistic. The range of weights is from -200 to 200. The network was allowed to train for up to 100,000 epochs for best results. The QuickProp training algorithm was used throughout the training process (for further information on QuickProp, see [17]).

Cross validation of the neural network is completed by examining the correlation of the target data with the neural network outputs. Cross validation uses the third set of data to determine how the network would perform on new data. This requires that the saved neural network makes estimates about the solution properties based only on the new input waveforms. In order to export the neural network estimates, a new QwikNet

neural network is created and loaded with the previously determined weights and with the unexposed third data set. The neural network produces a document with the target values and the neural network estimates for the third data set. A MATLAB algorithm was developed to compare these two values and determine how successful the ANN was in actually determining the conductivity and consistency from the spectral data. The next chapter evaluates the performance of the selected network by comparing the target values and the neural network outputs.

CHAPTER SIX

Results and Discussion

The results of the cross validation of the network appear in Figs. 11 and 12. Linear regression analysis on the output of the neural network for conductivity has an adjusted R-squared of .9957 with a standard deviation of 17.51 (span of the conductivity is 100 to 1000 µS). Linear regression analysis on the output of the neural network for consistency has an adjusted R-squared of .8944 with a standard deviation of .0750% (span of consistency is 0.00% to 1.25%).

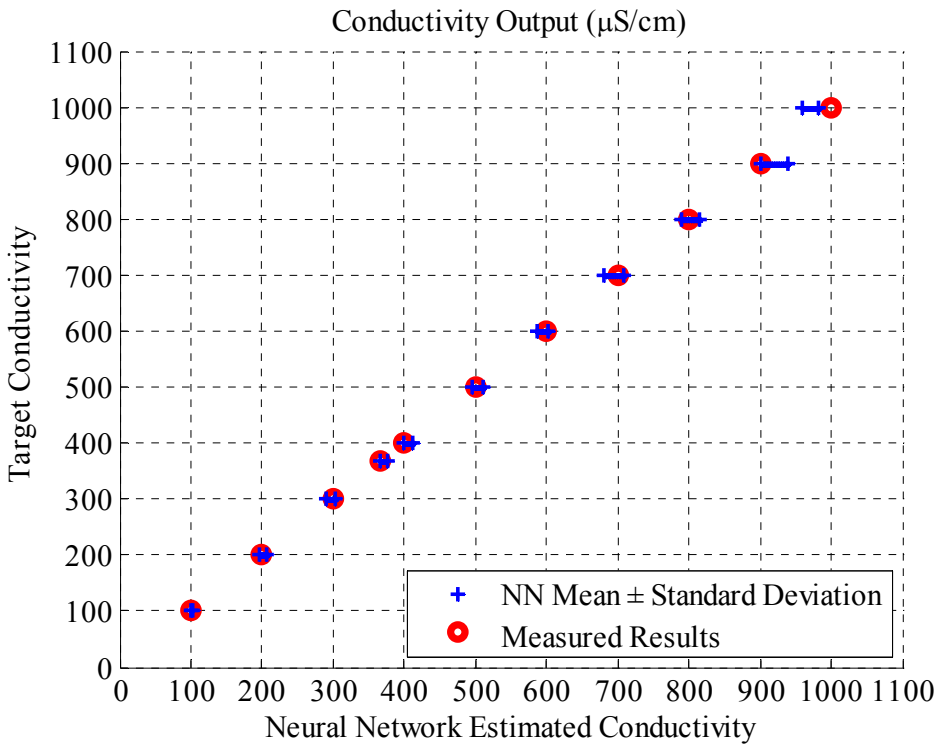


Fig. 12: Plot of the target data vs. neural network conductivity output.

The network gives excellent results on conductivity data. It is able to accurately determine the conductivity of the solution across the entire range of inputs. Minimal variation in the predicted-versus-known plot also indicates the network was accurate and indicates that the microwave sensor is well-calibrated. The results here are similar to those achieved by Green, et al. when using separate networks for determining conductivity and consistency. Results are also an improvement on the line-segment method detailed by [3].

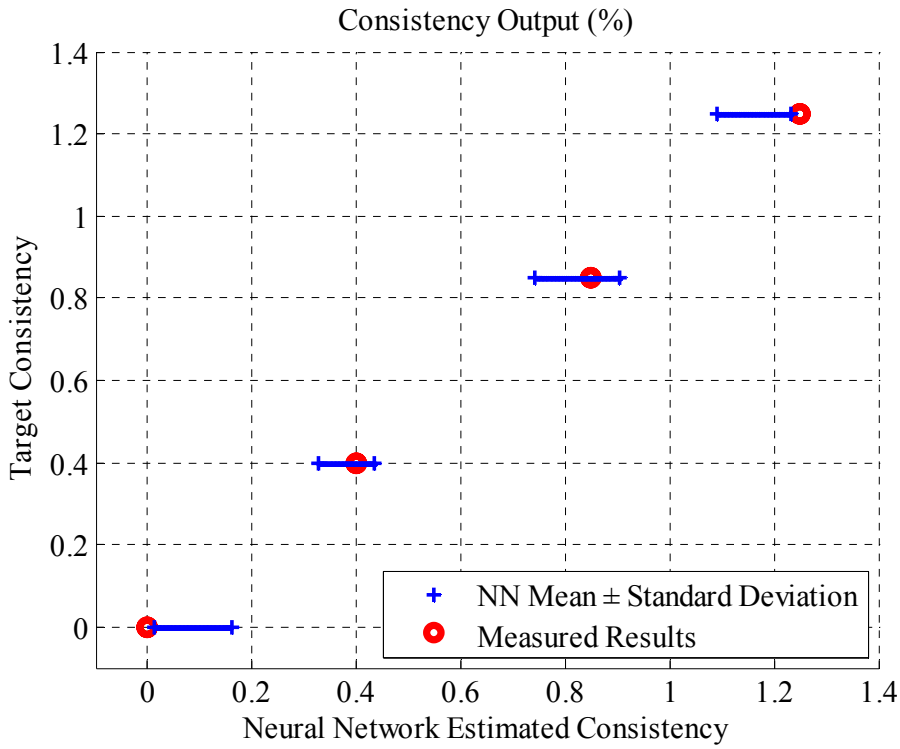


Fig. 13: Plot of the target data vs. neural network consistency output.

The results for consistency do not show an improvement over the method used in [7]. The lack of improvement indicates that combining the determination of consistency and conductivity into a single network results in less accurate outputs than when using a single network, or perhaps that no patterns became apparent to the network in the training

process. However, after training the network it became known that the data used to test the network contained samples which had been incorrectly prepared or presented to the instrument. Low consistency samples appear to have contained more paper pulp and high consistency samples contained less paper pulp. If so, then the network had been trained using data with incorrect target values. While it would seem that this would render the results invalid, it is interesting to note the output of the network with regards to consistency. The neural network has difficulty accurately predicting any of the consistency values. This shift shows that the network indicates that the network had trouble determining any patterns in the data set and thus is returning incorrect values. Due to the problems with the network recognizing patterns and returning accurate predicted values, a new set of data should be captured, that is carefully prepared and presented, in order to train an accurate network.

An interesting property of BBMSC is that this method presents an opportunity to determine whether inputs other than the captured microwave spectra represent an independent variable. In the process of training the neural network, an important question arose about whether the temperature of the sample was a necessary input or whether the information contained in the temperature was already contained within the principal components. In order to examine this, a series of plots were generated. For each plot, 2 of the 11 principal components were plotted against one another, while sorting (coloring) each point based upon temperature. The results showed that many of the plots showed strong patterns dependent on temperature (see Figs. 13 and 14). Where

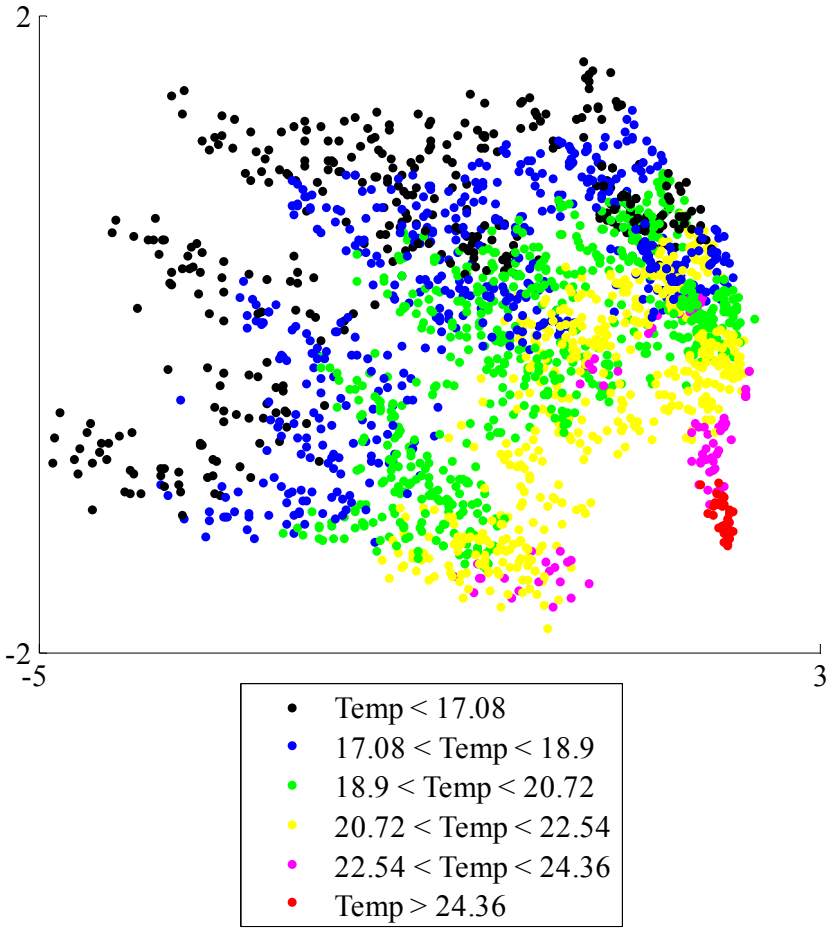


Fig 13: Example of how temperature affects microwave spectrometry data, which in turn affects principal components. Figures are of 2 of the 11 PCA values plotted on x- and y-axes. Each color represents a range of temperatures, from the warmest (red, ~26 °C) to the coolest (black, ~15 °C).

plotting any number of the pairs from the first 150 frequency indices, plotting all of the possible combinations of the 11 principal components resulted in only 55 separate plots. Thus by examining these plots and the general pattern of patterns dependant on temperature, it is possible to determine that temperature was acting as an independent variable and thus was a necessary inclusion in the neural network training process. Similarly, due to the reduction properties of PCA, it is possible that this could also be

used to determine the independence of other measured values when attempting to calibrate microwave sensors.

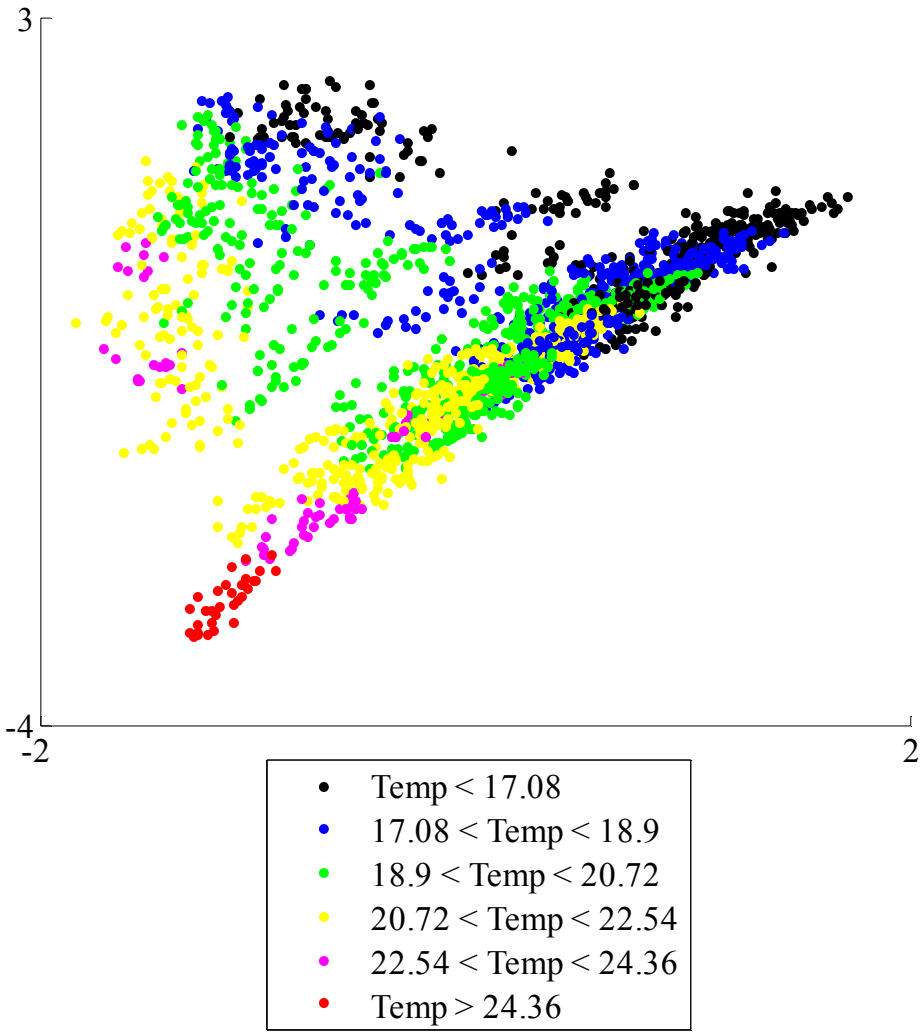


Fig 14: Example of how temperature affects microwave spectrometry data, which in turn affects principal components. Figures are of 2 of the 11 PCA values plotted on x- and y-axes. Each color represents a range of temperatures, from the warmest (red, ~26 °C) to the coolest (black, ~15 °C).

CHAPTER SEVEN

Conclusion

Microwave spectrometry has been shown to be a powerful tool for determining the properties of a subject material. Microwave sensors often produce vast sets of data, which require processing and interpretation of less than obvious trends to produce meaningful information. Principal component analysis is a powerful tool for minimizing the complexity of a data set and retaining information, while artificial neural networks are able to determine non-obvious and non-linear relationships within a data set. By using principal component analysis as a preprocessing step, the dimensionality of the data can be significantly reduced. This reduction allows for simpler processing including shorter training times for the neural network development process. The use of a neural network eliminates the need for expert knowledge of the waveform or the medium, by being able to examine the errors between target values and neural network outputs. As shown in the previous examples, both well- and poorly-calibrated data can be used to inform the network. The output of the network reveals where further calibration is required, simply by plotting the mean and standard error of the solution properties. The introduction of PCA also simplifies the process of determining the independence of other properties of the solution to the microwave spectrum data and whether their addition as neural network inputs is valid.

BBMSC achieves results which are comparable to earlier methods for determination of conductivity. It has yet to be tested on well-calibrated consistency data. However, it appears that BBMSC is capable of replicating any incorrect measurements or

poorly calibrated samples, thus revealing where further calibration is required. It also achieves a straight-forward approach for calibrating a microwave sensor. Instead of reducing select regions of the waveform to a characterization values, PCA is capable of reducing a large portion of the waveform to a few principal components, while retaining nearly all of the important information. Furthermore, BBMSC is also capable of retaining more information from the microwave spectral data as input to the ANN (150 frequency indices compared with 100 indices used by [7]), due to the reduction of PCA, while also reducing the training time required for the network to achieve accurate results. BBMSC also further simplifies the development process by using a single neural network.

Further development of BBMSC would include repeat testing with a new set of carefully prepared consistency samples. By implementing the neural network in MATLAB, the BBMSC process could be streamlined, by eliminating the need to generate the separate training, testing and validation files, as well as streamlining the validation process. Another test of BBMSC might include testing the process on other solutions or a wider range of target values. For more accurate results, it would be possible to develop separate networks for determining each property of future solutions as well, such as separate networks for the prediction of consistency and conductivity.

APPENDICES

APPENDIX A

Principal Component Algorithm

Before beginning the data should be in column form, where a single observation of the microwave spectra is in each column, and the log magnitude of the signals at a specific frequency index is stored in each row. For example, with 100 observations and 150 frequency indices, the data should be formatted with 150 rows and 100 columns.

Determine the empirical mean at each point in the observation.

$$\boldsymbol{u}[\boldsymbol{m}] = \frac{1}{\boldsymbol{N}} \sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \boldsymbol{X}[\boldsymbol{m},\boldsymbol{n}]$$

The data is normalized by the empirical mean:

$$\boldsymbol{B} = \boldsymbol{X} - \boldsymbol{u} \cdot \boldsymbol{h}$$

where h is a 1x150 row rector of ones:

$$\boldsymbol{h}[\boldsymbol{n}] = 1 \qquad \text{for } n = 1, 2, \ldots \text{ N}$$

The covariance matrix is produced by taking the outer product of B with itself.

$$\boldsymbol{C} = \frac{1}{\boldsymbol{N}} \boldsymbol{B} \cdot \boldsymbol{B}^{*}$$

The matrix of eigenvectors and eigenvalues are then computed:

$$\boldsymbol{V}^{-1} \boldsymbol{C} \boldsymbol{V} = \boldsymbol{D}$$

D is an MxM diagonal matrix where

$$\boldsymbol{D}[\boldsymbol{p},\boldsymbol{q}] = \lambda_{\boldsymbol{m}} \qquad \text{for } p = q = m$$

is the mth eigenvalue of the covariance matrix C, and

$$\boldsymbol{D}[\boldsymbol{p},\boldsymbol{q}] = 0 \qquad \text{for } p \neq q.$$

The matrix *V* contains eigenvalues that are paired with the eigenvectors in matrix *D*. The values in *V* are sorted in descending order, from greatest to least, while maintaining the pairing between V and C. In order to determine the most important dimensions of the data, the energy of the eigenvalues are summed to determine the total energy of the data. Some percentage of the energy is then kept, based upon the desired amount of information retention. Typical cutoff percentages are 90% or higher (a value of 99% was used). After determining the amount of energy desired to be retained, a subset of the eigenvectors, *W*, is selected as the new basis vectors, where

$$\boldsymbol{W}[\boldsymbol{p},\boldsymbol{q}] = \boldsymbol{V}[\boldsymbol{p},\boldsymbol{q}] \qquad \text{for } p = 1\ldots M,\ q = 1\ldots L$$

Create an *M* x 1 standard deviation vector *S* from the square root of each element along the diagonal of the covariance matrix C:

$$\boldsymbol{s} = \sqrt{\boldsymbol{C}[\boldsymbol{p},\boldsymbol{q}]} \qquad \text{for } p = q = m = 1\ldots M$$

Calculate the *M* x *N* z-score matrix.

$$\boldsymbol{Z} = \frac{\boldsymbol{B}}{\boldsymbol{s} \cdot \boldsymbol{h}} \qquad \text{(divide element by element)}$$

Finally project the z-scores of the data onto the new basis set to produce the new lower-dimension data.

$$\boldsymbol{Y} = \boldsymbol{W}^{*} \cdot \boldsymbol{Z}$$

where Y is the new data.

APPENDIX B

MATLAB Implementation of the PCA Algorithm

```
% u is a column vector with the mean of X
u = mean(X,2);
size(u)

h = ones(1,N);
B = X - u*h;
size(B)

C = 1/N * (B*B');

[V D] = eig(C);
% largest eigenvalues are last in the V matrix
for j = 1:size(D,1)
    for i = 1:size(D,2)
        Ddot(i,j) = D(end+1-i,end+1-j);
        Vdot(i,j) = V(end+1-i,end+1-j);
    end
end
V = Vdot;
D = Ddot;

% save the first L columns of V in new matrix W
% choose L based upon energy G (say 99% of G)
for m = 1:M
    G(m) = D(m,m);
end
energy = sum(G) % = 12801

L = 1;
while (sum(G(1:L)/energy)) < .99
    L = L+1;
end

W = V(:,1:L);

% these steps are not necessary
% but are intended to normalize data with respect to its variance
for i = 1:M
    s(i) = sqrt(C(i,i));
end
s = s';

Z = B./(s*h);

Y = W' * Z;
```

APPENDIX C

MATLAB Code for Writing the QwikNet Inputs

```
% write to text for qwiknet to process


load('pulp_stock_freq.mat')

% writes the training file
fid = fopen('pulpstock_freq.trn','w');
fprintf(fid,'[inputs] %3.0f\r',size(Y,1)+1);
fprintf(fid,'[outputs] 2\r\r');

start = 1;
stop = size(Y,1); % 5
count = 0;


%%%%%%%%%%%%%%%% 100uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq100us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 100',temp(j+count));
end
count = count+size(freq100us000Cs,2);

for j=1:3:size(freq100us040Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  040 100',freq100us040Cs(1,j));
end
count = count+size(freq100us040Cs,2);

for j=1:3:size(freq100us085Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  085 100',freq100us085Cs(1,j));
end
count = count+size(freq100us085Cs,2);

for j=1:3:size(freq100us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
```

```
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 100',freq100us125Cs(1,j));
end
count = count+size(freq100us125Cs,2);

%%%%%%%%%%%%%%%% 200uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq200us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 200',freq200us000Cs(1,j));

end
count = count+size(freq200us000Cs,2);

for j=1:3:size(freq200us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 200',freq200us125Cs(1,j));
end
count = count+size(freq200us125Cs,2);

%%%%%%%%%%%%%%%% 300uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq300us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 300',freq300us000Cs(1,j));
end
count = count+size(freq300us000Cs,2);

for j=1:3:size(freq300us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 300',freq300us125Cs(1,j));
end
count = count+size(freq300us125Cs,2);

%%%%%%%%%%%%%%%% 367uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq367us085Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  085 367',freq367us085Cs(1,j));
end
count = count+size(freq367us085Cs,2);
```

```
%%%%%%%%%%%%%%%% 400uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq400us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 400',freq400us000Cs(1,j));
end
count = count+size(freq400us000Cs,2);

for j=1:3:size(freq400us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 400',freq400us125Cs(1,j));
end
count = count+size(freq400us125Cs,2);

%%%%%%%%%%%%%%%% 500uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq500us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 500',freq500us000Cs(1,j));
end
count = count+size(freq500us000Cs,2);

for j=1:3:size(freq500us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 500',freq500us125Cs(1,j));
end
count = count+size(freq500us125Cs,2);

%%%%%%%%%%%%%%%% 600uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq600us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 600',freq600us000Cs(1,j));
end
count = count+size(freq600us000Cs,2);

for j=1:3:size(freq600us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 600',freq600us125Cs(1,j));
end
count = count+size(freq600us125Cs,2);
```

```
%%%%%%%%%%%%%%%% 700uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq700us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 700',freq700us000Cs(1,j));
end
count = count+size(freq700us000Cs,2);

for j=1:3:size(freq700us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 700',freq700us125Cs(1,j));
end
count = count+size(freq700us125Cs,2);


%%%%%%%%%%%%%%%% 800uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq800us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 800',freq800us000Cs(1,j));
end
count = count+size(freq800us000Cs,2);

for j=1:3:size(freq800us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 800',freq800us125Cs(1,j));
end
count = count+size(freq800us125Cs,2);

%%%%%%%%%%%%%%%% 900uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq900us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 900',freq900us000Cs(1,j));
end
count = count+size(freq900us000Cs,2);

for j=1:3:size(freq900us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 900',freq900us125Cs(1,j));
end
```

```
count = count+size(freq900us125Cs,2);

%%%%%%%%%%%%%%%% 1000uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=1:3:size(freq1000us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 1000',freq1000us000Cs(1,j));
end
count = count+size(freq1000us000Cs,2);

for j=1:3:size(freq1000us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 1000',freq1000us125Cs(1,j));
end
count = count+size(freq1000us125Cs,2);

status = fclose(fid);

%%

% write to text for qwiknet to process

load('pulp_stock_freq.mat')

% writes the test file
fid = fopen('pulpstock_freq.tst','w');
fprintf(fid,'[inputs] %3.0f\r',size(Y,1)+1);
fprintf(fid,'[outputs] 2\r\r');

start = 1;
stop = size(Y,1); % 5= 5;
count = 0;


%%%%%%%%%%%%%%%% 100uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq100us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 100',freq100us000Cs(1,j));
end
count = count+size(freq100us000Cs,2);

for j=2:3:size(freq100us040Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  040 100',freq100us040Cs(1,j));
```

```
end
count = count+size(freq100us040Cs,2);

for j=2:3:size(freq100us085Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  085 100',freq100us085Cs(1,j));
end
count = count+size(freq100us085Cs,2);

for j=2:3:size(freq100us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 100',freq100us125Cs(1,j));
end
count = count+size(freq100us125Cs,2);

%%%%%%%%%%%%%%%% 200uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq200us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 200',freq200us000Cs(1,j));

end
count = count+size(freq200us000Cs,2);

for j=2:3:size(freq200us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 200',freq200us125Cs(1,j));
end
count = count+size(freq200us125Cs,2);

%%%%%%%%%%%%%%%% 300uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq300us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 300',freq300us000Cs(1,j));
end
count = count+size(freq300us000Cs,2);

for j=2:3:size(freq300us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
```

```
    end
    fprintf(fid,' %2.3f  125 300',freq300us125Cs(1,j));
end
count = count+size(freq300us125Cs,2);

%%%%%%%%%%%%%%%% 367uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq367us085Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  085 367',freq367us085Cs(1,j));
end
count = count+size(freq367us085Cs,2);

%%%%%%%%%%%%%%%% 400uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq400us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 400',freq400us000Cs(1,j));
end
count = count+size(freq400us000Cs,2);

for j=2:3:size(freq400us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 400',freq400us125Cs(1,j));
end
count = count+size(freq400us125Cs,2);

%%%%%%%%%%%%%%%% 500uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq500us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 500',freq500us000Cs(1,j));
end
count = count+size(freq500us000Cs,2);

for j=2:3:size(freq500us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 500',freq500us125Cs(1,j));
end
count = count+size(freq500us125Cs,2);

%%%%%%%%%%%%%%%% 600uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq600us000Cs,2)
    fprintf(fid,'\r');
```

```
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 600',freq600us000Cs(1,j));
end
count = count+size(freq600us000Cs,2);

for j=2:3:size(freq600us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 600',freq600us125Cs(1,j));
end
count = count+size(freq600us125Cs,2);

%%%%%%%%%%%%%%%% 700uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq700us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 700',freq700us000Cs(1,j));
end
count = count+size(freq700us000Cs,2);

for j=2:3:size(freq700us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 700',freq700us125Cs(1,j));
end
count = count+size(freq700us125Cs,2);


%%%%%%%%%%%%%%%% 800uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq800us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 800',freq800us000Cs(1,j));
end
count = count+size(freq800us000Cs,2);

for j=2:3:size(freq800us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 800',freq800us125Cs(1,j));
end
count = count+size(freq800us125Cs,2);
```

```
%%%%%%%%%%%%%%%% 900uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq900us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 900',freq900us000Cs(1,j));
end
count = count+size(freq900us000Cs,2);

for j=2:3:size(freq900us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 900',freq900us125Cs(1,j));
end
count = count+size(freq900us125Cs,2);

%%%%%%%%%%%%%%%% 1000uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=2:3:size(freq1000us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 1000',freq1000us000Cs(1,j));
end
count = count+size(freq1000us000Cs,2);

for j=2:3:size(freq1000us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 1000',freq1000us125Cs(1,j));
end
count = count+size(freq1000us125Cs,2);

status = fclose(fid);

%%

% write to text for qwiknet to process

load('pulp_stock_freq.mat')

% writes the 2nd test file, used for verification
fid = fopen('pulpstock_freq_verify.tst','w');
fprintf(fid,'[inputs] %3.0f\r',size(Y,1)+1);
fprintf(fid,'[outputs] 2\r\r');

start = 1;
stop = size(Y,1); % 5= 5;
count = 0;
```

```
%%%%%%%%%%%%%%%% 100uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq100us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 100',freq100us000Cs(1,j));
end
count = count+size(freq100us000Cs,2);

for j=3:3:size(freq100us040Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  040 100',freq100us040Cs(1,j));
end
count = count+size(freq100us040Cs,2);

for j=3:3:size(freq100us085Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  085 100',freq100us085Cs(1,j));
end
count = count+size(freq100us085Cs,2);

for j=3:3:size(freq100us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 100',freq100us125Cs(1,j));
end
count = count+size(freq100us125Cs,2);

%%%%%%%%%%%%%%%% 200uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq200us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 200',freq200us000Cs(1,j));

end
count = count+size(freq200us000Cs,2);

for j=3:3:size(freq200us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 200',freq200us125Cs(1,j));
end
count = count+size(freq200us125Cs,2);
```

```
%%%%%%%%%%%%%%%% 300uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq300us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 300',freq300us000Cs(1,j));
end
count = count+size(freq300us000Cs,2);

for j=3:3:size(freq300us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 300',freq300us125Cs(1,j));
end
count = count+size(freq300us125Cs,2);

%%%%%%%%%%%%%%%% 367uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq367us085Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  085 367',freq367us085Cs(1,j));
end
count = count+size(freq367us085Cs,2);

%%%%%%%%%%%%%%%% 400uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq400us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 400',freq400us000Cs(1,j));
end
count = count+size(freq400us000Cs,2);

for j=3:3:size(freq400us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 400',freq400us125Cs(1,j));
end
count = count+size(freq400us125Cs,2);

%%%%%%%%%%%%%%%% 500uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq500us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 500',freq500us000Cs(1,j));
end
```

```
count = count+size(freq500us000Cs,2);

for j=3:3:size(freq500us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 500',freq500us125Cs(1,j));
end
count = count+size(freq500us125Cs,2);

%%%%%%%%%%%%%%%% 600uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq600us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 600',freq600us000Cs(1,j));
end
count = count+size(freq600us000Cs,2);

for j=3:3:size(freq600us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 600',freq600us125Cs(1,j));
end
count = count+size(freq600us125Cs,2);

%%%%%%%%%%%%%%%% 700uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq700us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 700',freq700us000Cs(1,j));
end
count = count+size(freq700us000Cs,2);

for j=3:3:size(freq700us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 700',freq700us125Cs(1,j));
end
count = count+size(freq700us125Cs,2);


%%%%%%%%%%%%%%%% 800uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq800us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
```

```
    end
    fprintf(fid,' %2.3f  000 800',freq800us000Cs(1,j));
end
count = count+size(freq800us000Cs,2);

for j=3:3:size(freq800us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 800',freq800us125Cs(1,j));
end
count = count+size(freq800us125Cs,2);

%%%%%%%%%%%%%%%% 900uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq900us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 900',freq900us000Cs(1,j));
end
count = count+size(freq900us000Cs,2);

for j=3:3:size(freq900us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 900',freq900us125Cs(1,j));
end
count = count+size(freq900us125Cs,2);

%%%%%%%%%%%%%%%% 1000uS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for j=3:3:size(freq1000us000Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  000 1000',freq1000us000Cs(1,j));
end
count = count+size(freq1000us000Cs,2);

for j=3:3:size(freq1000us125Cs,2)
    fprintf(fid,'\r');
    for k = start:stop
        fprintf(fid,'%4.0f',Y(k,j+count));
    end
    fprintf(fid,' %2.3f  125 1000',freq1000us125Cs(1,j));
end
count = count+size(freq1000us125Cs,2);

status = fclose(fid);
```

APPENDIX D

MATLAB Algorithm for Comparing Neural Network Outputs with Target Values

```
load('pulp_stock_freq.mat')

numpoints = 4;
start = 1;

fid = fopen('original150-12.out','r');
actual = fscanf(fid,'%f',[numpoints inf]);
fclose(fid);

consist = 0;
permit = 0;
conRight = 0;
permRight = 0;
for j=1:size(actual,2)
    newCon = sum(abs(actual(1,j)-actual(2,j)));
    newPerm = sum(abs(actual(3,j)-actual(4,j)));
    consist = newCon + consist;
    permit = newPerm + permit;

    if abs(newCon) < .01*max(actual(4,:))
        conRight = conRight + 1;
    end
    if abs(newPerm) < .01*max(actual(2,:))
        permRight = permRight + 1;
    end
end
avgConsistancyError = consist/size(actual,2)
fullScaleConsistancyError = avgConsistancyError/125;
avgConductivityError = permit/size(actual,2)
fullScaleConductivityError = avgConductivityError/1000;

fullScaleConsistancyPercent = fullScaleConsistancyError*100
fullScaleConductivityPercent = fullScaleConductivityError*100


[consistancyR2 p] = corrcoef(actual(1,:),actual(2,:));
consistancyR2 = consistancyR2(2,1)
[conductivityR2 p] = corrcoef(actual(3,:),actual(4,:));
conductivityR2 = conductivityR2(2,1)


%%%%%%%% plots the data for consistancyto show how the neural
%%%%%%%% network estimations are distributed
actual(1,:) = round(actual(1,:));
val1 = [0,40,85,125];
```

```
z1=zeros(size(val1,2),140);
err1 = zeros(size(val1));
avg1 = zeros(size(val1));
stdev1 = zeros(size(val1));
for k=1:4
    ind = find(actual(2,:) == val1(k));
    for j=ind
        z1(k,actual(1,j)+1) = z1(k,actual(1,j)+1)+1;
        err1(k) = err1(k) + abs(actual(2,j)-actual(1,j));
        avg1(k) = avg1(k) + actual(1,j);
    end
    err1(k) = err1(k)/size(ind,2);
    avg1(k) = avg1(k)/size(ind,2);

    for j=ind
        stdev1(k) = stdev1(k) + abs(actual(1,j) - avg1(k));
    end
    stdev1(k) = (1/size(ind,2))*stdev1(k);
end

figure(1)
clf
hold on

%plot based upon the standard deviations
for k=1:size(z1,1)
    b=plot([val1(k)/100 val1(k)/100],val1(k)/100,'ro','LineWidth',3);
    a=plot([avg1(k)/100-stdev1(k)/200
avg1(k)/100+stdev1(k)/200],val1(k)/100,'b+','LineWidth',2);
    c=plot([avg1(k)/100-
stdev1(k)/200:.001:avg1(k)/100+stdev1(k)/200],val1(k)/100,'b','LineWidt
h',.5);
end
title('Consistency Output ')
ylabel('Recorded Consistency')
xlabel('Neural Network Estimated Consistency')
legend([a(1);b],'NN Mean ± Standard Deviation','Measured
Results','Location','SouthOutside')
grid on
axis([-0.1, 1.4, -0.1, 1.401])


%%%%%%%% plots the data for conductivity to show how the neural
%%%%%%%% network estimations are distributed
actual(3,:) = round(actual(3,:));
val2 = [100:100:300 367 400:100:1000];

z2=zeros(size(val2,2),1000);
err2 = zeros(size(val2));
avg2 = zeros(size(val2));
stdev2 = zeros(size(val2));
for k=1:size(val2,2)
    ind = find(actual(4,:) == val2(k));
    for j=ind
        z2(k,actual(3,j)+1) = z2(k,actual(3,j)+1)+1;
        err2(k) = err2(k) + abs(actual(4,j)-actual(3,j));
```

```
        avg2(k) = avg2(k) + actual(3,j);
    end
    err2(k) = err2(k)/size(ind,2);
    avg2(k) = avg2(k)/size(ind,2);

    for j=ind
        stdev2(k) = stdev2(k) + abs(actual(3,j) - avg2(k));
    end
    stdev2(k) = (1/size(ind,2))*stdev2(k);
end

figure(2)
clf
hold on

for k=1:size(val2,2)
    b=plot(val2(k),val2(k),'ro','LineWidth',3);
    a=plot([avg2(k)-stdev2(k)/2
avg2(k)+stdev2(k)/2],val2(k),'b+','LineWidth',2);
    plot([avg2(k)-
stdev2(k)/2:.1:avg2(k)+stdev2(k)/2],val2(k),'LineWidth',.5);
end
title('Conductivity Output ')
ylabel('Recorded Conductivity')
xlabel('Neural Network Estimated Conductivity')
legend([a(1);b],'NN Mean ± Standard Deviation','Measured
Results','Location','SouthOutside','Orientation','Horizontal')
grid on
axis([0, 1100, 0, 1100])
```

REFERENCES

[1] E. Nyfors and P. Vainikainen, "Industrial Microwave Sensors," *IEEE MTT-S International Microwave Symposium Digest*, Vol. 3, pp. 1009-1012, June 1991..

[2] M. Tiuri. "Microwave Sensor Applications in Industry," *17th European Microwave Conference*, pp. 25-32, Oct. 1987.

[3] B. R. Jean, "Process Composition Monitoring at Microwave Frequencies: A Waveguide Cutoff Method and Calibration Procedure," *IEEE Trans. Instrumentation and Measurement*, Vol. 55, No. 1, pp. 180-186, Feb. 2006.

[4] Metso Corporation. [Online]. Available: http://www.metsopaper.com/

[5] P. Jakkula and E. Tahkola, "Microwave sensor system for the consistency measurement in the pulp and paper industry," in *Sensors Update*, vol. 7, K. Kupfer, A. Kraszewski, R. Knøchel, Eds. Hoboken, NJ: Wiley-VCH, 2000, ch. 8, pp. 211-232.

[6] D. M. Pozar, *Microwave Engineering*. Hoboken, NJ: John Wiley & Sons, Inc., 2005.

[7] E. C. Green, B. R. Jean, and R. J. Marks, II, "Artificial Neural Network Analysis of Microwave Spectrometry on Pulp Stock: Determination of Consistency and Conductivity," *IEEE Trans. Instrumentation and Measurement*, Vol. 55, No. 6, pp. 2132-2135, Dec. 2006.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego, CA: Academic Press Professional, 1990.

[9] S. L. Y. Lam, and D. L. Lee, "Feature Reduction for Neural Network Based Text Categorization," *6th International Conference on Database Systems for Advanced Applications*, pp. 195-202, April 19-21, 1999.

[10] L. Xu, Y. Yan, S. Cornwell, and G. Riley, "Online Fuel Tracking by Combining Principal Component Analysis and Neural Network Techniques," *IEEE Trans. Instrumentation and Measurement*, Vol. 54, No. 4, pp. 1640-1645, Aug. 2005.

[11] T. Denœux, and M.-H. Masson, "Principal Component Analysis of Fuzzy Data Using Autoassociative Neural Networks," *IEEE Trans. Fuzzy Systems*, Vol. 12, No. 3, pp. 336-349, June 2004.

[12] F. Daschner, R. Knoechel and M. Kent, "Rapid Monitoring of Selected Food Properties Using Microwave Dielectric Spectra," *2nd International Conference on Microwave and Millimeter Wave Technology*, pp. 670-673, Sept. 2000.

[13] F. Daschner, R. Knoechel and M. Kent, "Optimization of the Microwave Determination of Water in Foods Using Principal Component Analysis," *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference*, Vol. 1, pp. 12-16, May 2000.

[14] F. Daschner, R. Knoechel and M. Kent, "Multiparameter Microwave Sensors for Determining Composition or Condition of Substances," *2000 IEEE MTT-s International Microwave Symposium Digest*, Vol. 3, pp. 1567-1570, June 2000.

[15] Wikipedia. [Online]. Available: http://en.wikipedia.org/wiki/

[16] *QwikNet Professional Neural Network Software*, Craig Jensen. Version 2.23. [Online]. Available: http://qwiknet.home.comcast.net/

[17] R. D. Reed and R. J. Marks, *Neural Smithing*. Cambridge, MA: MIT Press, 1999.

[18] J. E. Rayas-Sánchez, "EM-Based Optimization of Microwave Circuits Using Artificial Neural Networks: The State-of-the-Art," *IEEE Trans. Microwave Theory and Techniques*, Vol. 52, No. 1, pp 420-435, Jan. 2004.

[19] E. Bermani, S. Caorsi, and M. Raffetto, "Microwave Detection and Dielectric Characterization of Cylindrical Objects from Amplitude-Only Data by Means of Neural Networks," *IEEE Trans. Antennas and Propagation*, Vol. 50, No. 9, September 2002.

[20] P. J. Edwards, A. F. Murray, G. Papadopoulos, R. Wallace, J. Barnard, and G. Smith, "The Application of Neural Networks to the Papermaking Industry," *IEEE Trans. Neural Networks*, Vol. 10, No. 6, pp. 1456-1464, Nov. 1999.

[21] F. Wang, V. K. Devabhaktuni, C. Xi and Q.-J. Zhang, "Neural Network Structures and Training Algorithms for RF and Microwave Applications," *International Journal of RF and Microwave Computer-Aided Engineering*, Vol. 9, Issue 3, pp. 216-240, May 1999.

[22] J. A. Jargon, K. C. Gupta and D. C. DeGroot, "Applications of Artificial Neural Networks to RF and Microwave Measurements," *International Journal of RF and Microwave Computer-Aided Engineering*, Vol. 12, Issue 1, pp. 3-24, January 2002.

[23] Q.-J. Zhang, K. C Gupta, and V. K. Devabhaktuni, "Artificial Neural Networks for RF and Microwave Design – From Theory to Practice," *IEEE Trans. Microwave Theory and Techniques*, Vol. 51, No. 4, pp. 1339-1350, April 2003.

[24] *MATLAB*, Version 7.1. [Online]. Available: http://www.mathworks.com/products/MATLAB/