ELECTROSTATIC SIMULATION OF A CONTACTLESS TEST SYSTEM FOR PRINTED CIRCUIT BOARDS

BY

SLAVKO SEREMET

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

ELECTROSTATIC SIMULATION OF A CONTACTLESS TEST SYSTEM FOR PRINTED CIRCUIT BOARDS

BY

SLAVKO SEREMET

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ABSTRACT

The printed circuit board test industry continually requires faster, more accurate and more economical techniques to find manufacturing defects such as open and short circuits. Cirlog Corporation developed a prototype test system, capable of detecting manufacturing defects without contacting the board under test, based on electric field principles called the Contactless Test System (CTS) (U.S. Patent 5517110). This thesis gives an overview of PCB test equipment, the CTS technology and describes the utilization of electrostatic field simulations and experimental measurements in the continued research and development of the CTS technology. An equivalent circuit model is developed for the CTS and various simulations are conducted to determine and demonstrate the characteristics and capabilities of the technology. Simulation techniques utilized include a parallel plate model, empirical formulas and solving the CTS geometry as an electrostatic problem. The electrostatic solution utilizes the Method of Moments (MoM), to solve the integral equation that arises from Poisson's equation subject to boundary conditions. Dielectric regions are incorporated into the simulation models and their influence on the CTS technology characteristics are described. A Complex Images technique is described for multiple layer dielectric geometries. Simulating the CTS technology proved very useful in understanding general characteristics associated with changing various parameters such as the dimensions and spacing of stimulators, sensors and the traces under test. Simulations also assist in understanding and locating the position of defects. Furthermore, the simulations determined that the prototype signal processing electronics use of rectification reduced sensitivity to some faults. Comparing

the simulation results to experimental data demonstrated that in order to accurately calculate the exact signals detected by the sensors, a more complicated model would be required that incorporates multiple dielectric layers. Furthermore, a more accurate positioning technique would be required to locate the traces that are scanned, such that simulation parameters are more accurately established. Not only are simulations an excellent tool for the research and development of the CTS technology, simulations may be utilized to generate the "Gold Board" data required for manufacturing defect analysis in a future test system.

ACKNOWLEDGEMENT

I would like to thank Professor Greg Bridges for his advice and guidance throughout the simulation research and development work on the CTS. I would also like to thank Jacob Soiferman for the opportunity to work for and use Cirlog Corporation's CTS technology as a basis for this thesis. Furthermore, I would like to thank Jim Petersen for devising the CTS, George Zong for the prototype mechanical design, Dave Fraser and Wayne Silvester for their work on the scanning and imaging software. Special thanks to Patricia Danais for her support and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The purpose of this thesis was to utilize simulations in the research and development of a Printed Circuit Board (PCB) Manufacturing Defect Analyzer (MDA). Cirlog Corporation in Winnipeg, Manitoba developed a prototype Contactless Test System (CTS). The CTS technology (U.S. Patent 5517110) utilizes electric field principles to detect faulty PCBs without contacting the surface of the Board Under Test (BUT). This report details electrostatic simulations and experimental measurements that were conducted to better understand and investigate the capabilities of the technology.

During the manufacture and subsequent handling of printed circuit boards, defects such as unwanted open circuits or short circuits may develop in or between circuit pathways and electronic components. Manufacturers continually look for faster, more accurate and more economical ways to find defects. It is necessary and cost effective to perform automated testing of both populated and unpopulated PCBs for manufacturing quality control.

There is a need for a test system and method that will detect manufacturing faults on printed circuit boards without contacting the BUT. Furthermore, there is a need for a technique that does not require functional test vectors, and that does not require isolating adjacent components on the printed circuit board for test purposes. There is a further requirement that the PCB being tested is not functioning or energized in an unique state

during the test procedure. There is still a further need in the art for a system with simple fixturing in which direct electrical contact between sensors or sources and the BUT is not required and the printed circuit board is tested independent of its structure and functionality. The test technique should be capable of detecting a wide range of faults, such as: open circuits, short circuits, missing components, misaligned components and solder cracks.

The CTS invention developed by Cirlog Corporation is an accurate method for detecting manufacturing defects on bare PCBs. The method does not require functional test vectors, isolation of adjacent traces, unique energizing states or direct electrical connection to the BUT. The technique is capable of detecting a wide range of faults such as: open circuits, short circuits, missing traces, misaligned traces and solder cracks. A prototype called the CTS_128, built by Cirlog Corporation, is described in this thesis and simulations and experimental measurements are presented that were utilized to determine characteristics and capabilities of the technology. Cirlog Corporation designed and built the prototype and developed the scanning and imaging software. My work involved developing and performing simulations and collecting experimental data using the prototype to compare and analyze results.

Simulating the CTS technology was important for better understanding the technology in order to predict results. Information from simulations aided in optimization of performance and in defining the characteristics for various geometries, which gave insight into the effects of faults. The ultimate simulator would be able to generate the Gold Board data, which would replace the scanning of a known good board

for defect testing. The requirement of a known good board to compare against for fault detection is an undesirable but common feature in PCB test systems today.

Electrostatic simulations were utilized to demonstrate the characteristics of simple geometries and the effects of changing various parameters, such as spacing and dimensions of tracks and sensors. The characteristics were studied by constructing an equivalent circuit model and then determining the values of the individual components in the model with simulations.

Various simulation techniques were implemented to calculate the signals detected by the CTS sensors. The simulation techniques include: the parallel plate model, empirical formulas and solving the CTS geometry as an electrostatic problem. The electrostatic solution involved utilizing the Method of Moments (MoM), to solve integral equation formulated from Poisson's equation subject to boundary conditions. The CTS electrostatic problem was solved in a homogeneous region (free space) and also with dielectric layers in the geometry.

Chapter 2 provides an overview of PCB test equipment. An overview is provided for both bare and loaded PCB testers and a more detailed description is given for the state of the art testers available today that utilize contactless electromagnetic features.

Chapter 3 describes the CTS technology in detail. A theoretical explanation is given and the CTS_128 prototype built by Cirlog Corporation is described. Experimental data I collected from the prototype is presented and analyzed.

Chapter 4 describes the different simulation techniques utilized in studying the CTS technology. Empirical formulas are described along with electrostatic solutions and the Method of Moments (MoM) technique for solving integral equations. Techniques for incorporating the dielectric effects into the simulation model are included and the accuracy of the techniques is investigated. Simulations are utilized to determine the general characteristics of the CTS technology. CTS_128 prototype experimental data is compared against simulation data for a straight track segment and the incorporation of dielectric layers into the model is presented.

Finally, Chapter 5 concludes on the simulation findings and presents recommendations for future work involving simulations to continue the research and development of Cirlog Corporation's CTS technology.

2. OVERVIEW OF PCB TEST METHODS

This section of the report discusses existing test techniques and equipment in industry to test unpopulated and populated printed circuit boards (PCBs) and gives detailed descriptions of electromagnetic features utilized. The CTS technology is currently implemented for unpopulated PCBs and simulations have only been performed for unpopulated PCBs. Populated PCB testing would introduce numerous tolerance issues for the CTS technology in its current form. The signals detected by the sensors are very sensitive to height variations and proximity to the board is an important factor.

2.1 Overview Of Testing Unpopulated PCBs

Visual inspection with cameras is the most common test technique used for unpopulated PCBs. Electrical testing with probes directly contacting the surface of the PCB is also very common. The disadvantage of using a technique that touches the surface of the PCB is the possibility of damage caused by the probes. The following is a brief description of different types of test technique for unpopulated or bare PCBs.

2.1.1 Automated Optical Inspection (AOI)

Automated Optical Inspection (AOI), or visual inspection, involves the use of cameras to photograph a known good board, or "Gold Board", which becomes the standard other boards are tested against. The black and white images of bare PCB traces are compared with sophisticated software techniques to find and isolate faults. The

length and width of the smallest bit of acquired visual information must be smaller than the smallest fault needed to detect. AOI test systems are the most expensive unpopulated test systems.

2.1.2 Optical Inspection

Low volume production testing of PCBs is sometimes performed by human inspection[1]. The operator is supplied with a split image, one of a Gold Board and one of the board under test.

2.1.3 Flying Probe Tester

Flying Probe testers [2] are used for both low and high volume testing. This technique is an electrical inspection technique using two probes. Both probes are positioned and connected to the board under test and continuity or isolation measurements are made. The test systems can be found with multiple probe pairs to speed up testing for higher volume production with larger sized PCBs that have many test points. By touching the surface of the PCB, the probes could potentially cause damage. These systems can be used without testing of a known good board, depending on the test requirements. A Flying Probe tester is shown in Figure 2-1.

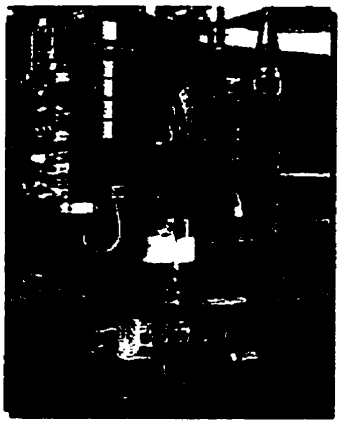


Figure 2-1: Bare Board Flying Probe Tester [source: Bath Scientific]

2.1.4 Universal Grid Tester (Bed-Of-Nails)

A Universal Grid Tester [3], or Bed-Of-Nails test system, utilizes a large matrix of spring contact probes to contact the BUT. Probe grid sizes vary depending on the testing requirements. The test system exercises the test by utilizing switching circuitry and making measurements at specified test locations. By using information about the layout of the test board, test programs are written to switch the test measurement points as required. A measurement and electrical connection must be made between the BUT and the test system for each test point. One test channel is required for each point on the probe grid. A Bed-Of-Nails tester is illustrated in Figure 2-2 and a picture is shown in Figure 2-3.

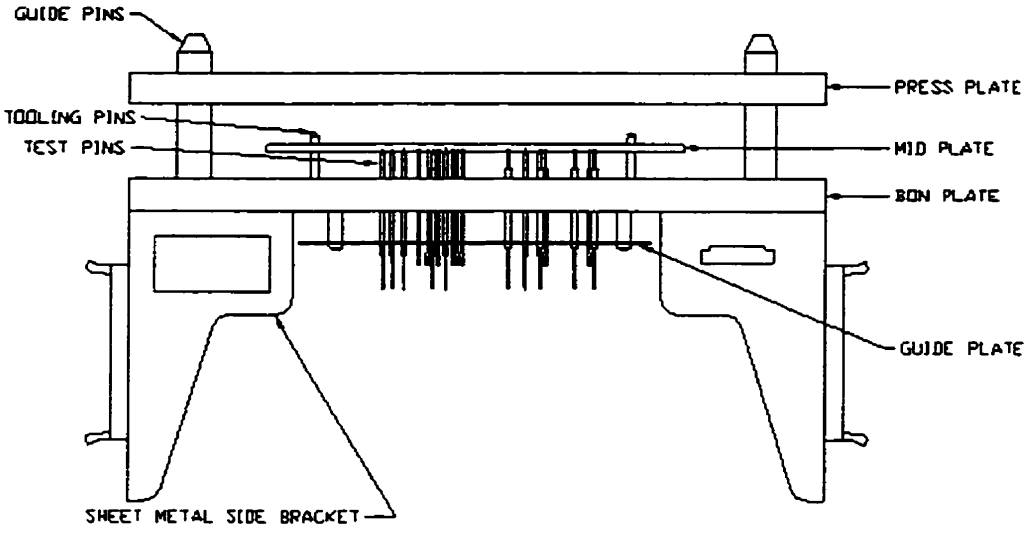


Figure 2-2: Bed-Of-Nails Test [source: Test Electronics]

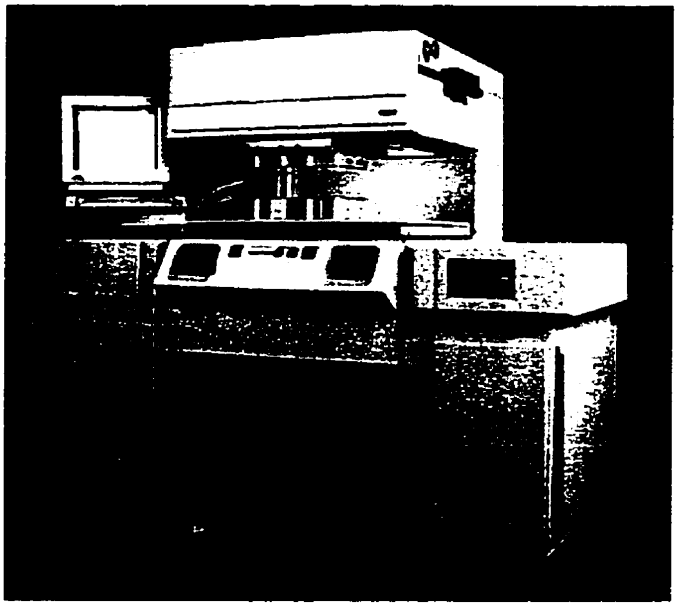


Figure 2-3: Bed-Of-Nails Tester [source: Everett Charles Technologies]

2.1.5 Dedicated Test Fixtures

Dedicated test fixtures are customized Bed-Of-Nails system containing only those probes required to test a particular BUT. These system are typical for large scale

production of specific PCBs. If the geometry of the board was to change the fixture would have to be redesigned accordingly.

2.2 Overview Of Testing Populated PCBs

Loaded PCBs are tested using a variety of different approaches [1][4]. Different types of test systems include: in-circuit testers (ICTs), manufacturing defects analyzers (MDAs), functional testers, automated optical inspection (AOI), analog signature analysis testers (ASAs), X-ray techniques and dedicated test systems. Many of these test systems have cross over capabilities and some are designed for high volume testing while others are designed for low volume testing and troubleshooting. All the test systems have undesirable features ranging from mechanical requirements to lack of fault coverage. A brief description of these test systems follows.

2.2.1 In-Circuit Testers (ICT)

ICTs are designed to test components that are in a circuit and on PCBs that are unpowered. Most ICTs use a Bed-Of-Nails fixture, which utilizes spring contact probes or "nails", to connect to the BUT. The nails connect to the BUT by pressure electrical contacts made to nodes and connect to the test system with wires that carry the test signals back and forth. The ICT tests each component, passive and active, one at a time. Passive components, such as resistors and capacitors, are tested by applying a voltage across the component and measuring the resulting current. Active components, such as transistors and ICs, are tested by applying specific inputs and checking the outputs with the use of truth tables.

ICTs also include vectorless testing techniques used to detect manufacturing faults not detected by simple probing. Vectorless technique options are usually add-on features and include capacitive and inductive techniques designed for finding open circuit faults. Vectorless testing techniques are also found on MDAs and as stand alone systems. Many ICTs have Boundary Scan[5][6][7] testing capabilities, which is a test technique involving specific designing of integrated circuits (ICs) for testability.

2.2.1.1 Boundary Scan

In many cases it is not possible to make contact on the BUT with a Bed-Of-Nails or a Flying Probe type tester. Surface mount technology (SMT) and high density board designs prevent direct nodal access and require a different way to perform continuity tests between devices and the PCB. These types of designs require Boundary Scan test capabilities to find manufacturing defects such as open circuits. Boundary Scan requires powering up the BUT.

Boundary Scan (ANSI/IEEE 1149.1 standard) is a special type of scan path with a register added at every I/O pin on a device. The BUT must contain components that support Boundary Scan and by sending test patterns through scan chains (interconnected Boundary Scan components) the output pattern is evaluated. This technique allows fault isolation at the component level. Many integrated circuits do not include Boundary Scan circuitry, however more and more SMT device are available with the Boundary Scan capability. The Boundary Scan test technique is an add on feature for ICTs and can also be found as a stand alone test system.

2.2.2 Manufacturing Defects Analyzers (MDAs)

Manufacturing Defects Analyzers (MDAs) [8] assume that the ICs on the BUT are good and testing is done on the BUT unpowered. An MDA uses a test fixture to probe circuit nodes much like an ICT, but only tests for manufacturing faults such as opens, shorts and missing components. MDAs use either Bed-Of-Nails fixtures or Flying Probes to connect to PCB nodes. Flying Probes use contact probes attached to xyz positioning scanners instead of the traditional Bed-Of-Nails to send test signals back and forth from the BUT. A Flying Probe tester is illustrated in Figure 2-4. Most MDAs measure the impedance between pairs of nodes on the unpowered BUT. A voltage is applied to one node and the current is measured exiting at another node, which is connected to the first node by a component.

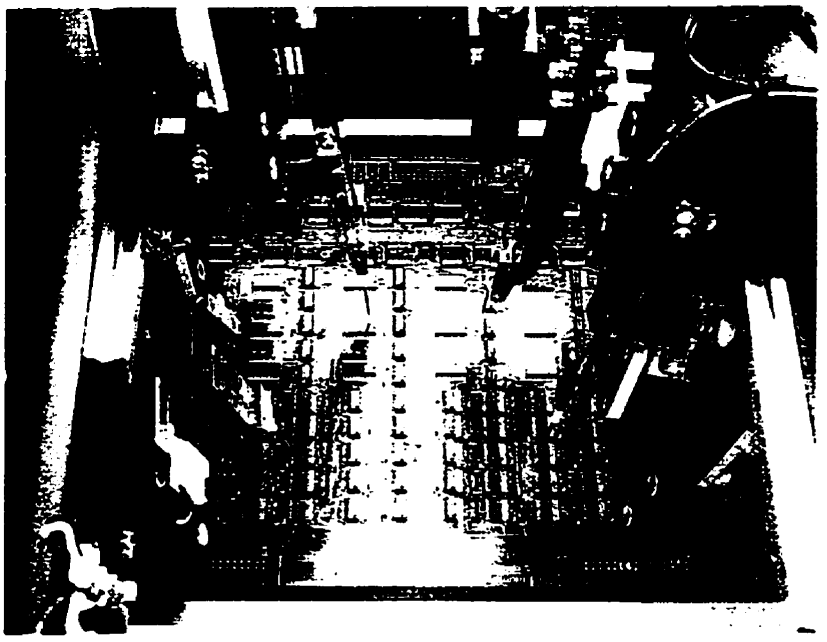


Figure 2-4: Flying Probe Tester [source: Takaya Corp.]

2.2.2.1 Vectorless Test

Another test technique that detects manufacturing faults and in some cases defective ICs is called "vectorless testing" [9][10]. The three types of vectorless testing are: P-N junction testing, capacitive testing and inductive testing. All three techniques require a Bed-Of-Nails test fixture to access nodes and the capacitive and inductive techniques require over clamp fixtures as shown in Figure 2-5. Vectorless testing was designed to eliminate the need for lengthy generation of test patterns required by other types of testers. Test programs for vectorless testing are relatively easy to generate and they supply excellent repair information. Like other MDAs, vectorless techniques allow PCB testing without powering the BUT.

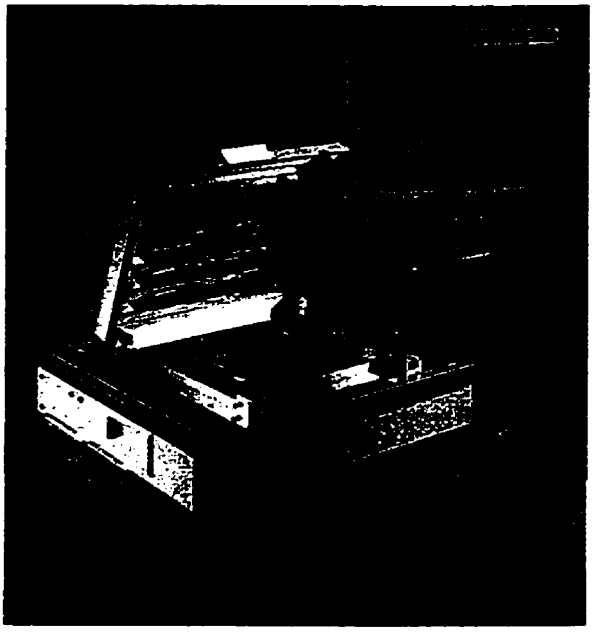


Figure 2-5: Vectorless Test System (source: Teradyne)

2.2.3 Functional Testers

Functional test systems have the capability to power up the BUT to find performance oriented problems. The BUT is powered up to its operating environment and checked against its functional specifications. Functional testing is required since some BUTs will pass tests on ICTs and MDAs, which only test individual components, but fail when run under normal operating conditions. The reasons for failing under normal operating conditions include problems with interactions between components or ICs that fail under normal operating clock rates.

A functional test system determines whether or not the BUT works properly. However it provides less detailed information on faults when the BUT fails, as compared to ICTs and MDAs.

2.2.4 Analog Signature Analysis (ASA)

Analog Signature Analysis (ASA) is similar to an MDA except that ASA testers are based on the idea that semiconductor devices' nonlinearities contain useful information. MDAs assume impedance is linear between nodes, whereas ASA determines the shape of the I vs V curve that represents the nonlinear impedance between nodes. ASA is sometimes referred to as VI trace. An ASA tester applies an electrical stimulus to a component, then it creates a unique current-voltage analog signature (impedance signature) of the in-circuit behavior of that component. ASA testers do not use a Bed-Of-Nails, but rather come with hand held probes or as a Flying Probe style tester. ASA testers are useful when a Bed-Of-Nails style test is not practical due to pin density or physical limits or when high cost fixturing is not practical due to low volumes.

ASA is a power off test like ICTs and MDAs, but is designed to usually do low volume testing and troubleshooting. Some companies offer bench top ASAs with options to build up to an automated testing station with a computer interface, oscilloscope, positioning camera and xyz scanner. The automated ASA setup is sometimes used as a low volume MDA. Figure 2-6 is a picture of bench top ASA systems.

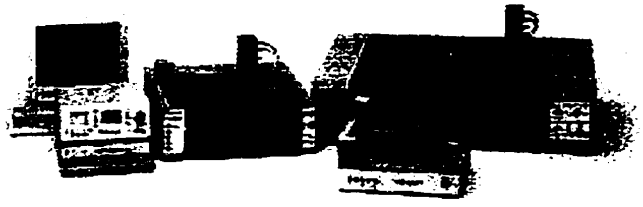


Figure 2-6: Analog Signature Analysis Tester [source: Huntron]

2.2.5 Automated Optical Inspection (AOI)

Automated Optical Inspection (AOI) test systems [1][11] test for manufacturing faults such as shorts, opens, missing components, misoriented components and track width or length variations. The AOI testers are usually placed at the end of an assembly line to catch manufacturing faults in place of an MDA. An advantage of optical inspection is that no connections are made to the loaded BUT.

2.2.6 Dedicated Testers

In many cases a dedicated test fixture is designed for specific PCBs in order to facilitate testing. Dedicated testers are essentially a Bed-Of-Nails tester, containing only those probes required to test a specific BUT. Furthermore they are designed to test for specified types of faults, which means a dedicated test system could be an ICT, MDA,

functional tester or a combinational tester. PC motherboard test stations are a common example of a dedicated tester.

2.2.7 Other Techniques

Some other techniques that are more sophisticated and expensive include X-ray imaging, Scanned-Beam Laminography and Thermal imaging systems [1][12][13][14]. X-ray imaging is used primarily for multilayer PCBs and detects minute defects such as hairline cracks around a via, that escape other methods. Scanned-Beam Laminography is an X-ray technique that separates the top and bottom sides, or any other layers, into separate images. Thermal imaging systems indicate hot spots on operating PCBs and point out defects such as shorts and over stressed components.

2.2.8 Summary of Loaded PCB Testers

All the test systems, except for optical and x-ray inspection systems, require nodal access to connect probes electrically to the BUT. Bed-Of-Nails, Flying Probes or manual probes are used to contact the BUT and the direct contact with the BUT can damage and introduce faults on the BUT. Some testers require test vectors to be written before the BUT can be tested. The added time to begin testing is undesirable.

Mechanical fixturing requirements add cost to test systems. All the Bed-Of-Nails type test systems require one of three ways to electrically connect the BUT to the nails: vacuum fixtures, pneumatic fixtures (air pressure) or simply mechanical force fixtures. Vectorless techniques also require a Bed-Of-Nails fixture and the capacitive and inductive techniques require extra over-clamp fixturing.

Some techniques are limited since the components on the BUT can not be tested. Boundary Scan testing requires the components on the BUT to have Boundary Scan capabilities built in. Added circuitry makes Boundary Scan components more expensive and as a result many components on BUTs are not Boundary Scan compatible. However, SMT and density of BUTs is resulting in more and more ICs with this capability.

2.3 Electromagnetic Based Test Techniques

This section of the report discusses state of the art test systems that utilize electromagnetic principles and represent the competition for the CTS technology in the marketplace. The competition includes conventional MDAs and vectorless test systems that find manufacturing faults. Most MDAs use either a Bed-Of-Nails fixture or a Flying Probe fixture to probe circuit nodes and test for manufacturing faults such as opens, shorts, missing components and misoriented components. The vectorless test techniques also require direct circuit node probing and some techniques also employ contactless probes in conjunction with the direct probing.

The state of the art test systems all require direct electrical contact with the BUT and perform electrical measurements by applying voltages or currents to circuit nodes and measuring other circuit nodes. A few of the techniques, including Hewlett Packard's TestJet, GenRad's Opens Xpress and Teradyne's WaveScan, utilize non-contact probes to carry out these electrical measurements and are considered as Vectorless techniques. A detailed description of the technology behind some state of the art testers follows.

2.3.1 TestJet

TestJet [15], from Hewlett-Packard Company, utilizes a capacitive technique in conjunction with a Bed-Of-Nails and determines if open solder connections exist on populated printed circuit boards. TestJet was the first technique to utilize non-contact electromagnetic means to test a board.

As illustrated in Figure 2-7, open solder joints are found by measuring the capacitance between the lead frame of an IC and a probe. The probe is positioned over top of the device under test and consists of a copper plate and a high gain amplifier to boost the signal, which is read by an analog measuring system. The copper plate acts like a transducer and forms a capacitor with the IC lead frames. An input signal is run through a lead of an IC (standard Bed-Of-Nails technique) and the capacitance between lead and plate is measured. The capacitance level indicates if a proper connection between the lead and the printed circuit board exists. The capacitance of an open connection is much less than the capacitance of a good connection. A probe is required for each device under test and covers the device to within 1 mm of the top surface. The board under test is not powered.

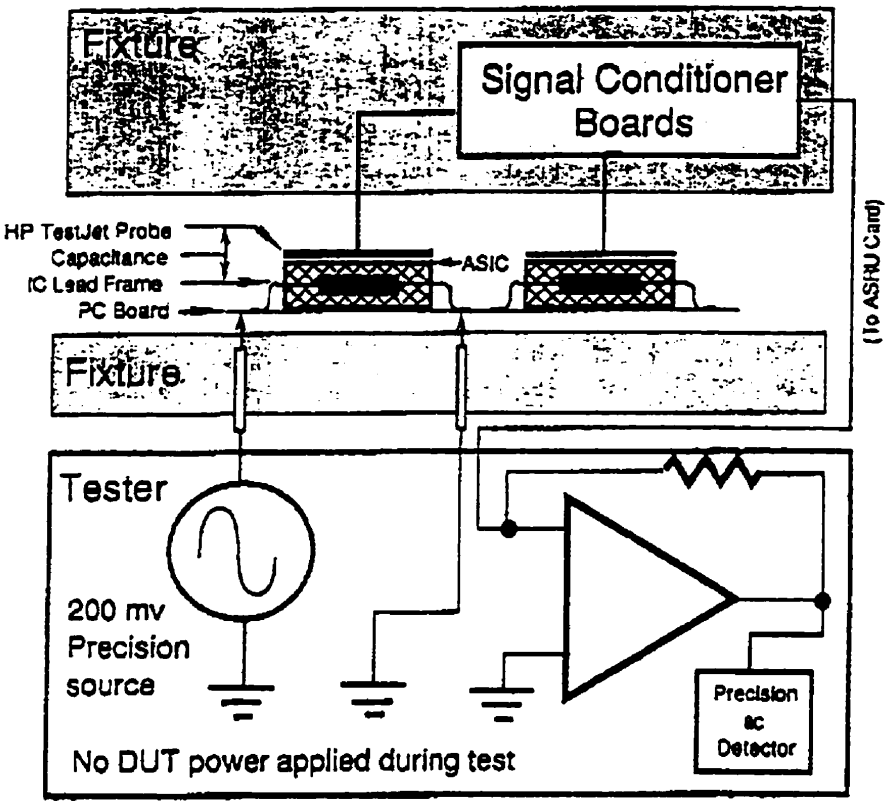


Figure 2-7: TestJet Operation [source: Hewlett Packard]

2.3.2 Opens Xpress

Opens Xpress [16][17] from GenRad Incorporated, like TestJet, determines if open solder connections exist on populated printed circuit boards using a capacitive technique and a Bed-Of-Nails. The technique finds open faults by measuring the capacitance between the lead frame of a component and a plate clamped over the device under test as shown in Figure 2-8 and Figure 2-9. The plate is in direct physical contact with the device under test and no powering of the device is required. One plate is required for each device to be tested.

The technique relies on applying an AC voltage to each lead of a device (Bed-Of-Nails technique) and then measuring the resulting voltage coupled through the device. The plate detects this voltage as it forms a capacitor with the lead of the device under test. Levels are compared to measurements from known good boards on a pin-by-pin basis to determine faults.

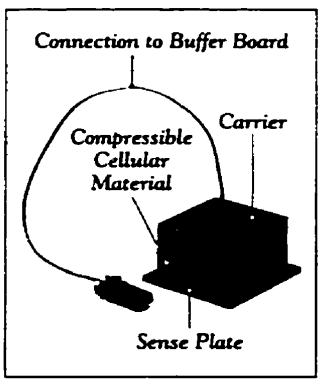


Figure 2-8: Sense Plate [source: GenRad]

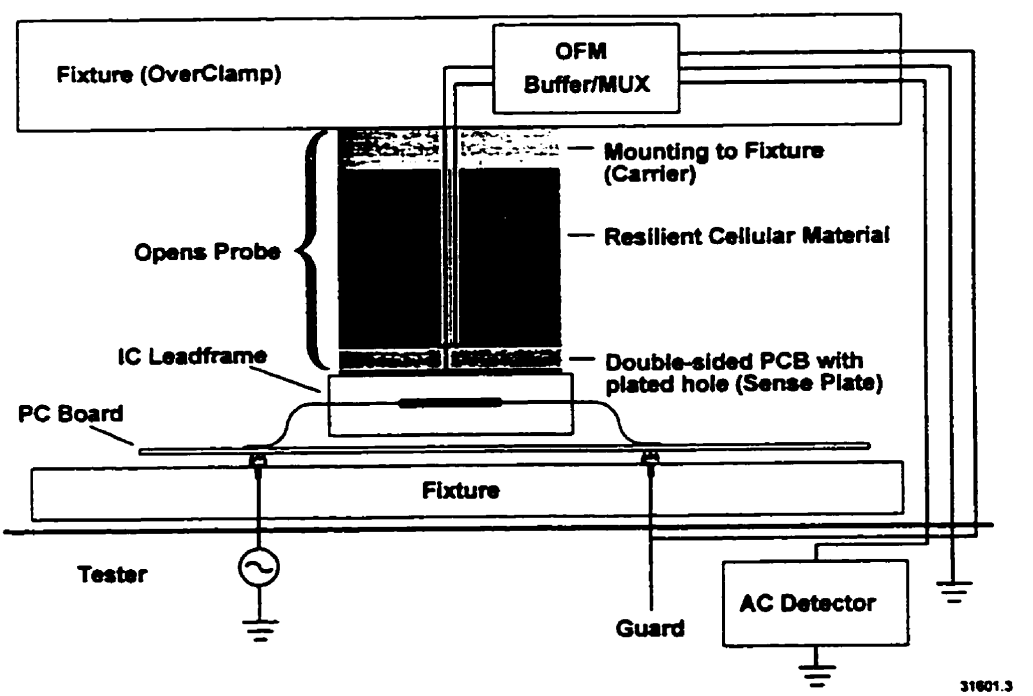


Figure 2-9: Opens Xpress Measurement Connections [source: GenRad]

2.3.3 MultiScan Vectorless Test System

Teradyne offers a vectorless test system for finding manufacturing faults called MultiScan. MultiScan includes a two pin diode technique (DeltaScan), an inductive technique (WaveScan) and a capacitive technique (FrameScan) similar to TestJet and Opens Xpress.

DeltaScan, illustrated in Figure 2-10, is an analog junction test that uses a Bed-Of-Nails fixture. It performs simple DC current measurements on unique pin pairs on the device under test, using the protection (or parasitic) diodes present on I/O pins.

WaveScan, illustrated in Figure 2-11, utilizes an inductive technique and a Bed-Of-Nails to find open circuit faults for various devices that connect to printed circuit boards. The technique depends on establishing a conductive path through a device (short circuit) by forward biasing the substrate diode within the device. Spiral loop antennas [18] (inducers) are placed above the device under test. The inducers radiate a magnetic field onto the device, thus inducing current flow on the conductive path through the device. Voltage measurements are taken via a Bed-Of-Nails fixture, from which it is determined whether a proper connection exists.

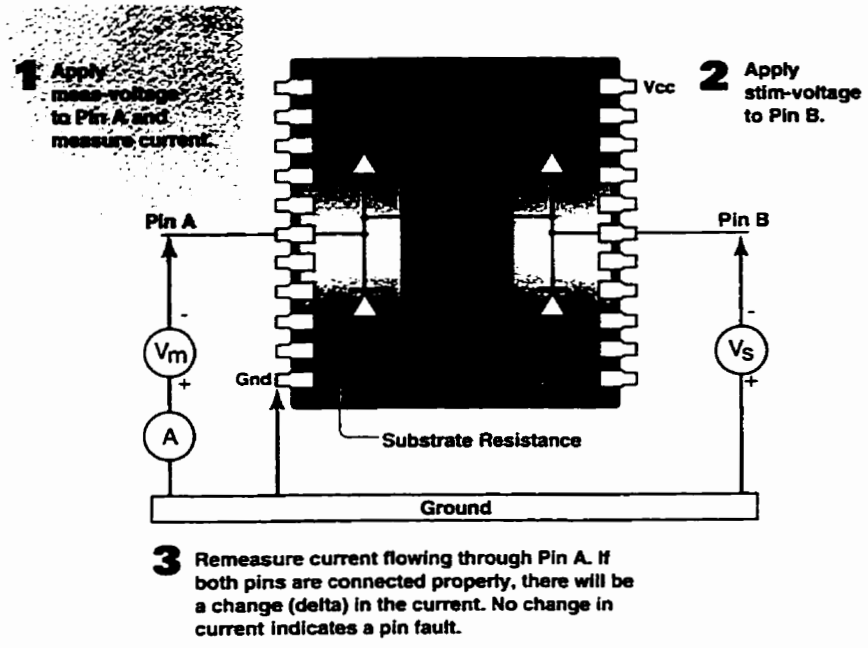


Figure 2-10: DeltaScan [source: Teradyne]

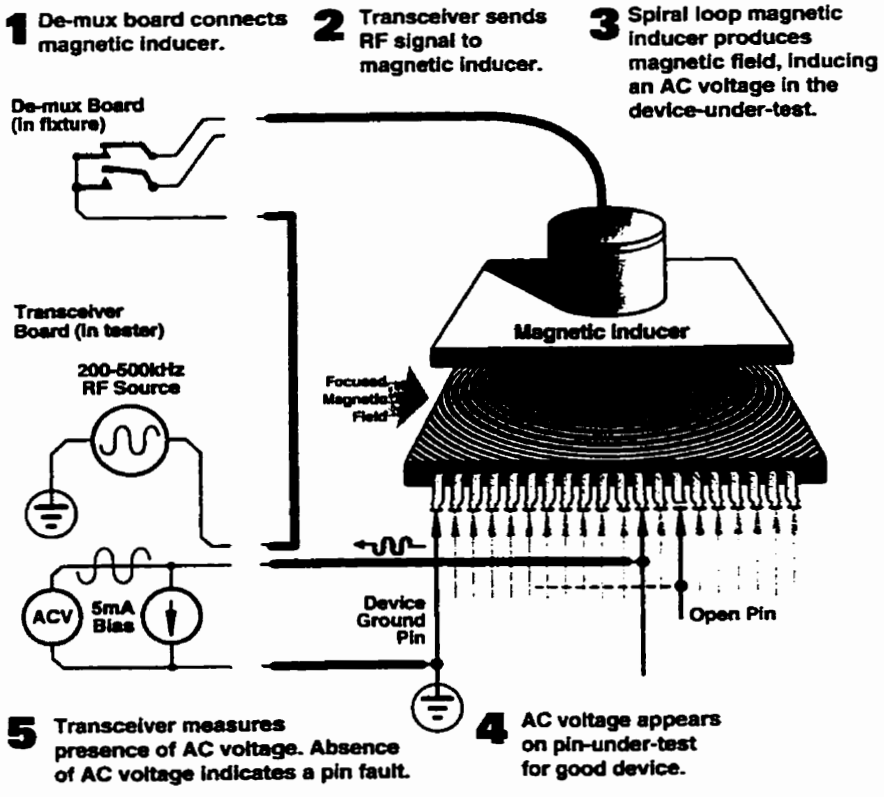


Figure 2-11: WaveScan [source: Teradyne]

The transmitters (inducers), illustrated in Figure 2-12, consist of an array of spiral loop antenna, designed to radiate a magnetic field onto devices under test. The magnetic field cuts through the conductors leading to the substrate diode inside the device. This magnetic excitation will give rise to a current, which will flow through the device if a proper connection exists between its leads and the printed circuit board. During the testing of a device, only the antenna structure directly above the device under test in powered. The frequency of the RF signal used to stimulate the device under test is in a range from 500 KHz to 2 MHz.

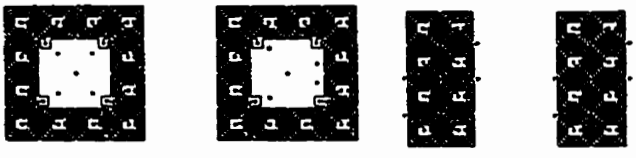


Figure 2-12: WaveScan Inducers [source: Teradyne]

FrameScan is illustrated in Figure 2-13.

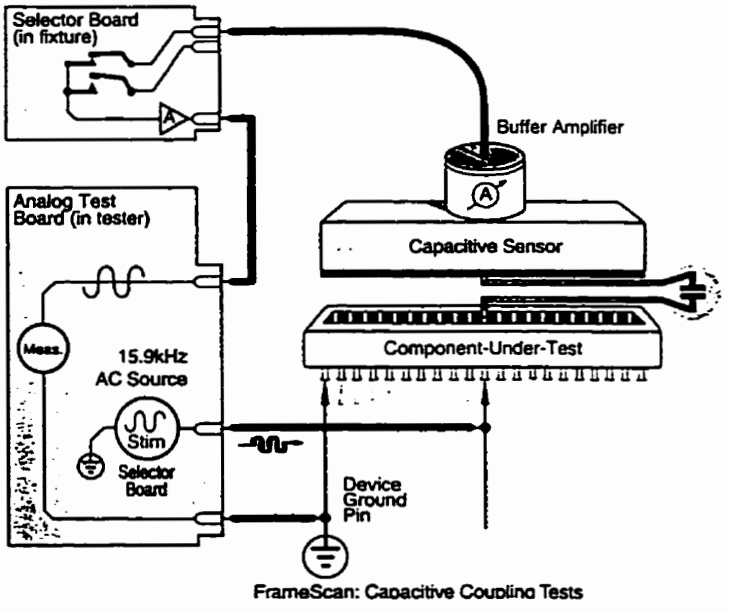


Figure 2-13: FrameScan [source: Teradyne]

2.3.4 ChipScan and C-Scan

ITA Corporation's ICT and MDA testers, which are standard Bed-Of-Nails and Flying Probe testers as discussed earlier, include two techniques called ChipScan and C-Scan [4]. ChipScan is a technique that performs a three pin test on I/O pins, biasing the pins so as to generate an active transistor current, which it measures to determine continuity. The method scans the IC until it has used all pins in at least one three-pin test, then it isolates a detected failure to one of three pins or by deduction to one pin. C-Scan is a technique very similar to TestJet, Opens Xprss and FrameScan except that it uses a dipole type probe instead of a patch to detect opens and misaligned polarized capacitors. Figure 2-14 illustrates C-Scan and reversed polarized capacitor testing.

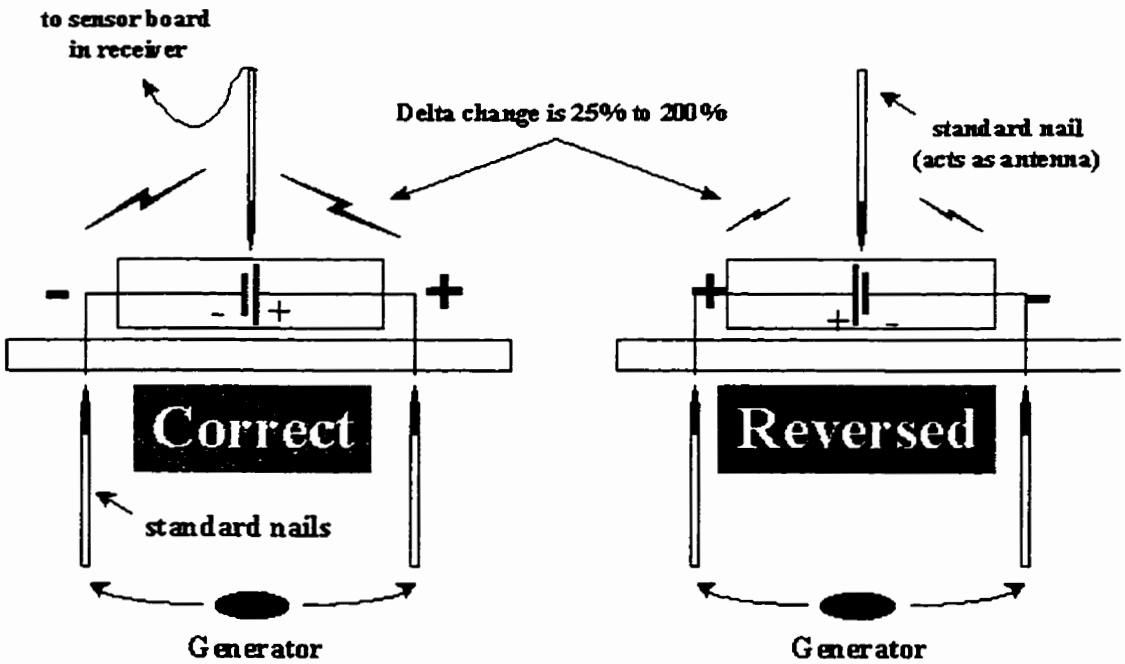


Figure 2-14: C-Scan - Reverse Polarized Capacitor Testing [source: ITA Corp.]

3. Contactless Test System Technology

Cirlog Corporation's patented Contactless Test System (CTS) technology is designed to test for manufacturing defects on PCBs, includes defects such as open circuits and short circuits. The test apparatus includes the Sensor/Stimulator Board (SSB), mechanical means for placing the board under test (BUT), mechanical means for scanning the SSB across the BUT and a computer work station for data analysis. The SSB does not electrically contact the BUT, as a result many advantages are gained over other PCB test system that require contacting the BUT with probes.

The SSB consists of two stimulator plates, a row of sensors and two ground strips plus signal processing electronics. Patch sensors are located between the two stimulators and ground strips are located between the stimulators and the row of sensors. Conducting plates constitute the stimulators, which radiate electric fields onto the BUT. Ground strips reduce direct coupling from the stimulators into sensors. The sensors detect the signals from the BUT. Figure 3-1, which is not to scale, depicts the SSB face geometry and clarifies the arrangement of sensors, stimulators and ground strips.

Typical dimensions for the patch sensors on the SSB are 0.05 inches square (1.27 mm^2) with a center to center spacing of 0.06 inches (1.52 mm). The ground strip width between the sensor and stimulators is typically 0.03 inches (0.76 mm).

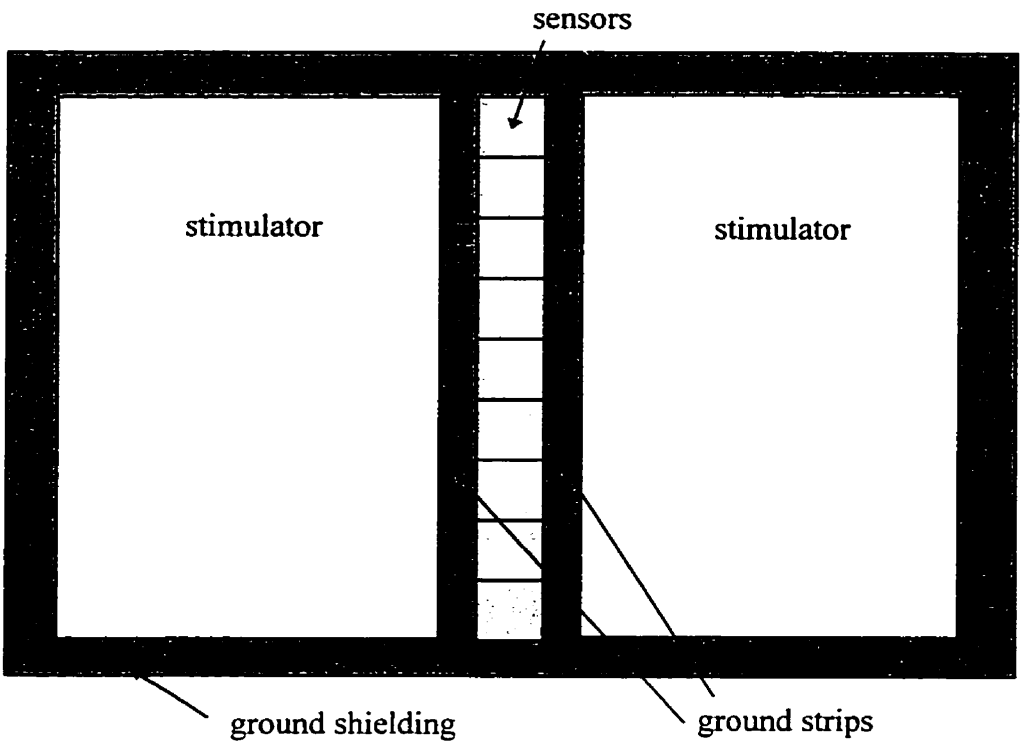


Figure 3-1: Sensor / Stimulator Board Face (not to scale)

3.1 Contactless Test System Operation

In operation an AC source provides a signal to the stimulators. Best results are obtained when the stimulators are supplied with equal amplitudes and 180 degrees "Out-of-Phase" sinusoidal signals. An "In-Phase" signal could also be supplied. The stimulators in turn induce an electric field onto the BUT. A voltage gradient is created on the BUT traces during the scanning motion and the sensors pick up the displacement currents[19][20] from the BUT traces. Sensor electronics utilize a transimpedance amplifier (a high bandwidth amplifier) to generate a low impedance at the sensors. The low impedance at the sensors provides good isolation between adjacent sensors, means

for generating an electric field gradient, high signal-to-noise ratio and means to detect displacement current flowing between the sensors and traces on the BUT. A voltage is obtained as output from the SSB, which is representative of the displacement current. The voltage signal is digitized by an A/D board and read by the computer for analysis and fault detection.

By scanning across the entire surface of the BUT, a displacement current signature of the BUT is obtained. In other words, by scanning across the entire surface of the BUT, the voltage on the traces changes for each new position and the sensors detect the change in voltage by the displacement current that is detected. Either the SSB is stationary and the board under test moves or vice versa. Figure 3-2 depicts the scanning motion. Manufacturing faults can be detected with the aid of computer analysis. A fault recognition system compares the pattern of the BUT with a known or desired pattern of an identical, non-faulty board. Such comparisons result in a measure of the difference between the non-faulty and faulty boards. If the difference is larger than a pre-determined threshold the BUT is diagnosed to be faulty, or out of tolerance. If the difference is smaller than the threshold, the BUT is diagnosed to be non-faulty.

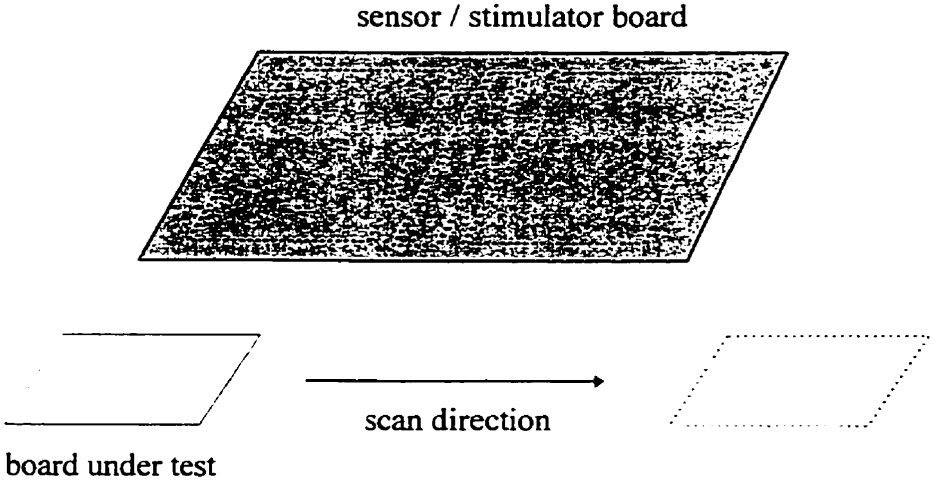


Figure 3-2: Contactless Test System Scanning Motion

3.2 Theoretical Explanation

Electromagnetic theory [19][20] states wherever there are electric charges, there are electric fields and wherever there are electric currents, there are magnetic fields. Electric field lines originate from positive charges and terminate at negative charges, and magnetic field lines loop around currents. Temporal variations of electric fields produce magnetic fields and temporal variations of magnetic fields produce electric fields. Furthermore, Ohm's law states that electric currents will be induced by electric fields in a conductor and Gauss's law states that the electric flux passing through any closed surface is equal to the total charge enclosed by that surface.

The CTS_128 can be described by considering electric fields, with the two stimulators acting like an electric dipole. That is, the stimulators are equal in amplitude

and 180 degrees out of phase. The equipotential (solid) and electric field lines (dashed) of an electric dipole are depicted in Figure 3-3. An equipotential surface is a surface composed of all those points having the same value of potential. No work is involved in moving a unit charge around on an equipotential surface, since there is no potential difference between any two points.

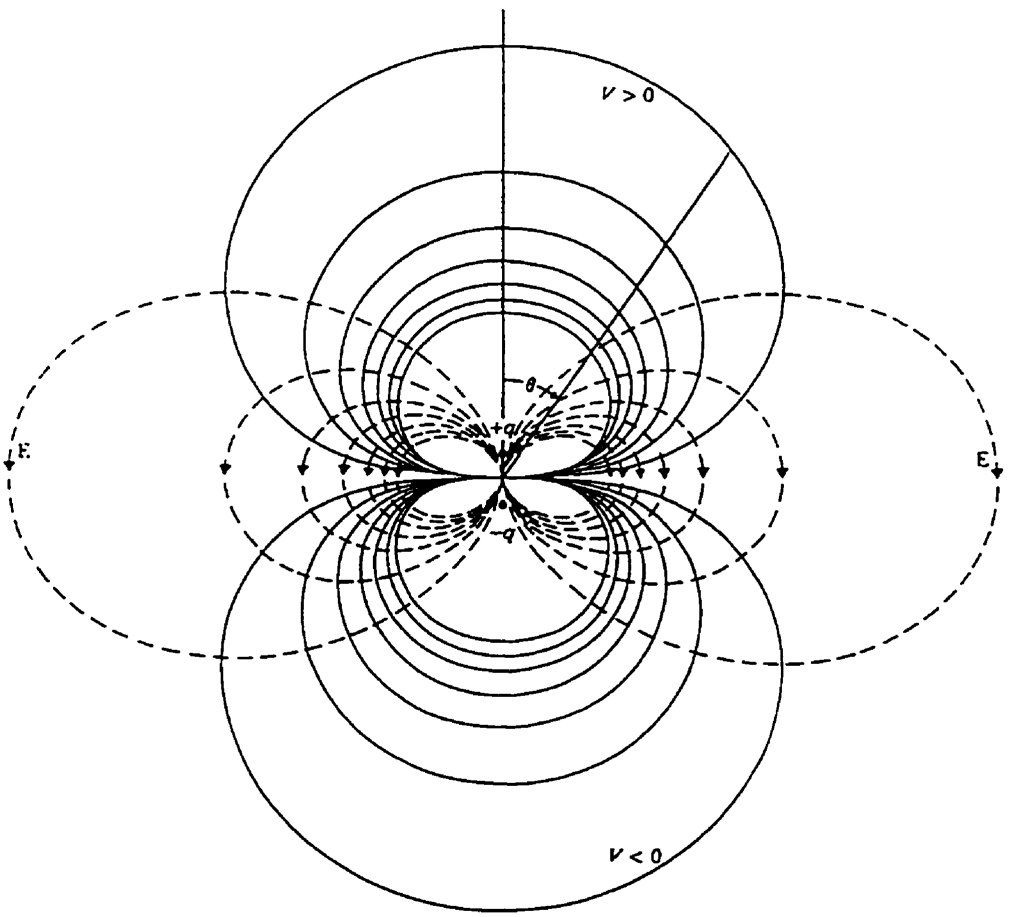


Figure 3-3: Electric Dipole [19]

The signal picked up by the sensors is proportional to the displacement current flowing into the sensors. This current is proportional to the frequency and voltage of the conductive elements below the sensors and to the capacitance of these conductive elements to the sensors. As the field is changed, charges have to realign in accordance with Gauss's Law.

$$q = \varepsilon \oint \bar{E} \cdot d\bar{S} \tag{3-1}$$

Basically, the charge in the conductors must realign itself in accordance with the external Electric field so that there is no Electric field inside the conductors. The external Electric field is caused by the conductive elements below the sensors being at a different potential than the sensors. The sensors are considered to be at ground potential. Figure 3-4 illustrates the charge realignment and displacement current flow during operation at a single track postion (scan stopped). The process is similar to charging a metal object by induction [21].

The flow of charge on and off the sensor patch is continuously changing as the board under test moves across the SSB and also changes as the signal applied to the stimulators oscillates. These illustrations show a realignment of charges due to a change in signal applied to the stimulators.

Another way to express the charge realignment is:

$$q = \sum C_j V_j \tag{3-2}$$

where C_j is the capacitance between the sensor and a particular track, and V_j is the potential of the track. The individual capacitances to the tracks can be determined by setting the voltages of surrounding tracks to 0V, applying a voltage to the track in question and measuring the charge transferred to the sensor. This has to be done for all tracks which have a significant affect on the charge transferred to the sensor. This is the superposition principal for electric fields [19]. Changing the geometry or the applied voltage will change the capacitance. It should be noted there is no superposition of capacitances, since if a track has a certain capacitance to the sensor, bringing another track close to it will change that capacitance. You can not consider that the same capacitance remains and then the capacitance of the other track can be added to it.

The capacitance is independent of the potential and total charge, for their ratio is constant. If the charge density is increased by a factor N, Gauss's Law indicates that the electric flux density also increases by N, as does the potential difference. The capacitance is a function only of the physical dimensions of the system of conductors and the permittivity of the dielectrics [19].

As the distance between the SSB and the surface of the BUT increases, the more the neighboring tracks affect the sensor response. If one track is closer to a particular sensor, it tends to reduce the exposure of that sensor to the fields from other tracks.

The displacement current is calculated as:

$$i = \frac{dq}{dt} = \sum C_j \frac{dV_j}{dt} \tag{3-3}$$

Since V_j is a sinusoidal wave, the amplitude of the displacement current can be expressed as:

$$i_{peak} = 2\pi f \sum C_j V_{j\,peak} \tag{3-4}$$

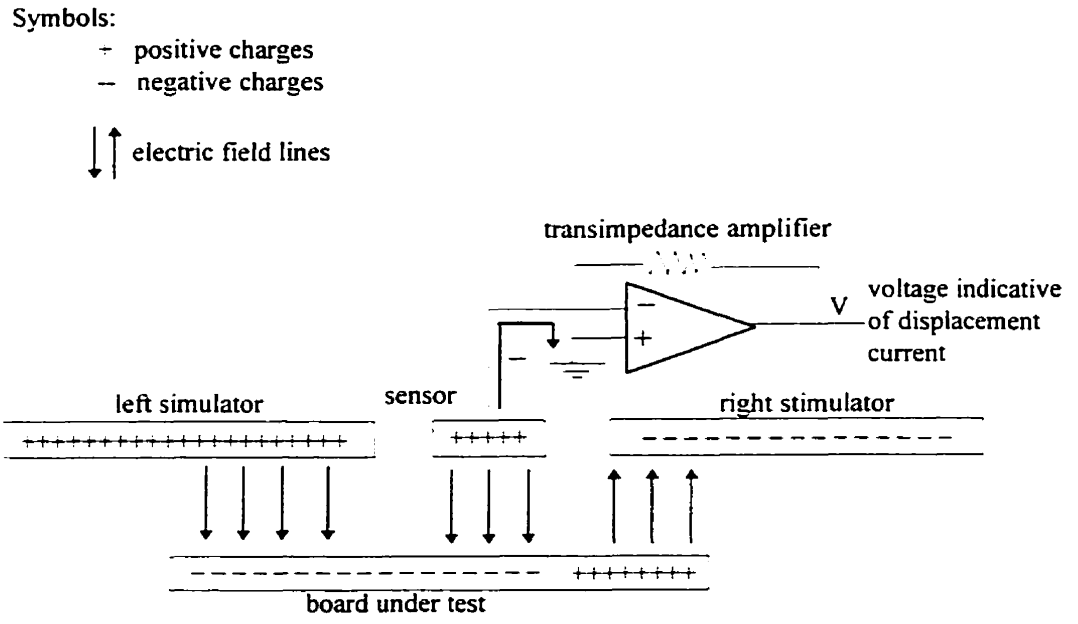
The amplitude of the displacement current increases as the frequency of the track voltage increases and the potential of the track depends on a number of parameters. The potential of a track depends on:

- capacitance between the stimulus and the track
- capacitance between the track and ground
- capacitance between the track and the ground strips
- the potential of the stimulus

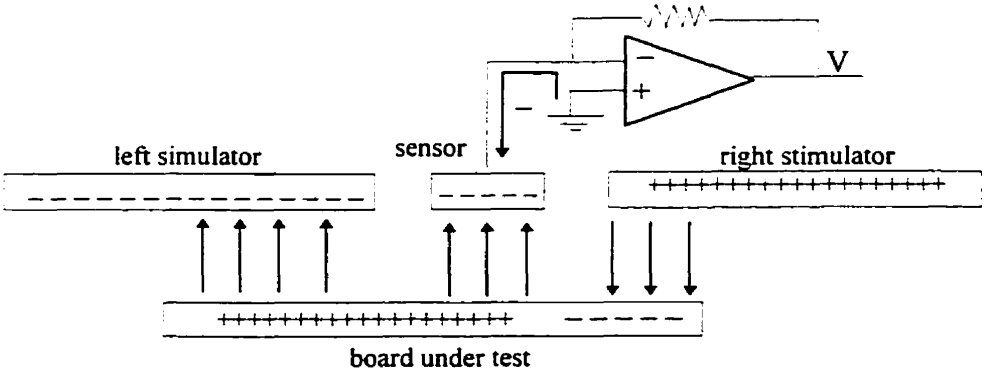
These relationships are expressed in the equivalent circuit model for the CTS as depicted in Figure 3-5 and the following equations that result from solving the equivalent circuit using nodal or mesh analysis[22]. The CTS can be modeled in terms of capacitance and the track voltage can be expressed by:

$$V_{track} = \frac{C_{track,stim1} V_{stim1} + C_{track,stim2} V_{stim2}}{C_{total}}$$

$$C_{total} = C_{track,stim1} + C_{track,stim2} + C_{track,ground} + C_{track,ground\ strip1} + C_{track,ground\ strip2} \tag{3-5}$$



(A) With Left Stimulator More Positive



(B) With Left Stimulator More Negative

Figure 3-4: Displacement Current Flow

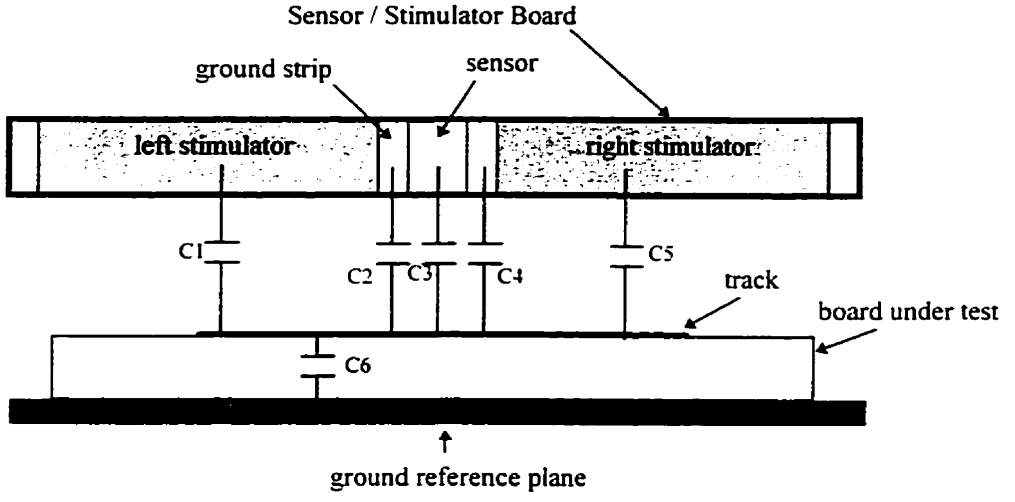
The current picked up by the sensor is expressed by:

$$i_{sensor} = \frac{V_{track}}{Z_{C_{track,sensor}}} = V_{track}\,(j\omega C_{track,sensor}) = j\,\omega\,V_{track}\,C_{track,sensor}$$

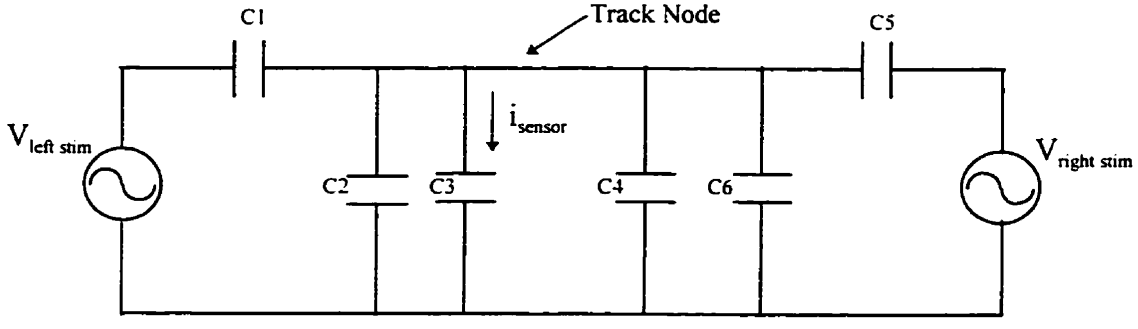
where, (3-6)

$\omega = 2\pi f$

A voltage is obtained from the transimpedance amplifier output (on the sensor stimulator board) that is representative of this current. This analog voltage signal passes through many stages on the SSB (multiplexers, filters, rectifiers, integrators). Refer to Appendix B for the sensor signal processing circuit details. Eventually the voltage signal is digitized by an A/D board and read by the computer for analysis and fault detection.



(a) Cross Sectional View



(b) Equivalent Circuit Diagram

Notation:
C1 - capacitance of left stimulator to the track
C2 - capacitance of the track to the ground strip between the sensor and the left stimulator
C3 - capacitance of the track to the sensor
C4 - capacitance of the track to the ground strip between the sensor and the right stimulator
C5 - capacitance of the right stimulator to the track
C6 - capacitance of the track to the ground reference plane

$V_{left\ stim}$ - voltage signal applied to the left stimulator
$V_{right\ stim}$ - voltage signal applied to the right stimulator

i_{sensor} - current picked up by the sensor

Figure 3-5: Cross Section of CTS and Equivalent Circuit

3.3 Prototype Description

The CTS_128 Prototype consists of a SSB, custom designed mechanical frame, positioning table, platform for the board under test, positioning pins, a vacuum fixture and a computer workstation. Figure 3-6 illustrates the prototype setup. Refer to Appendix C for the frame dimensions.

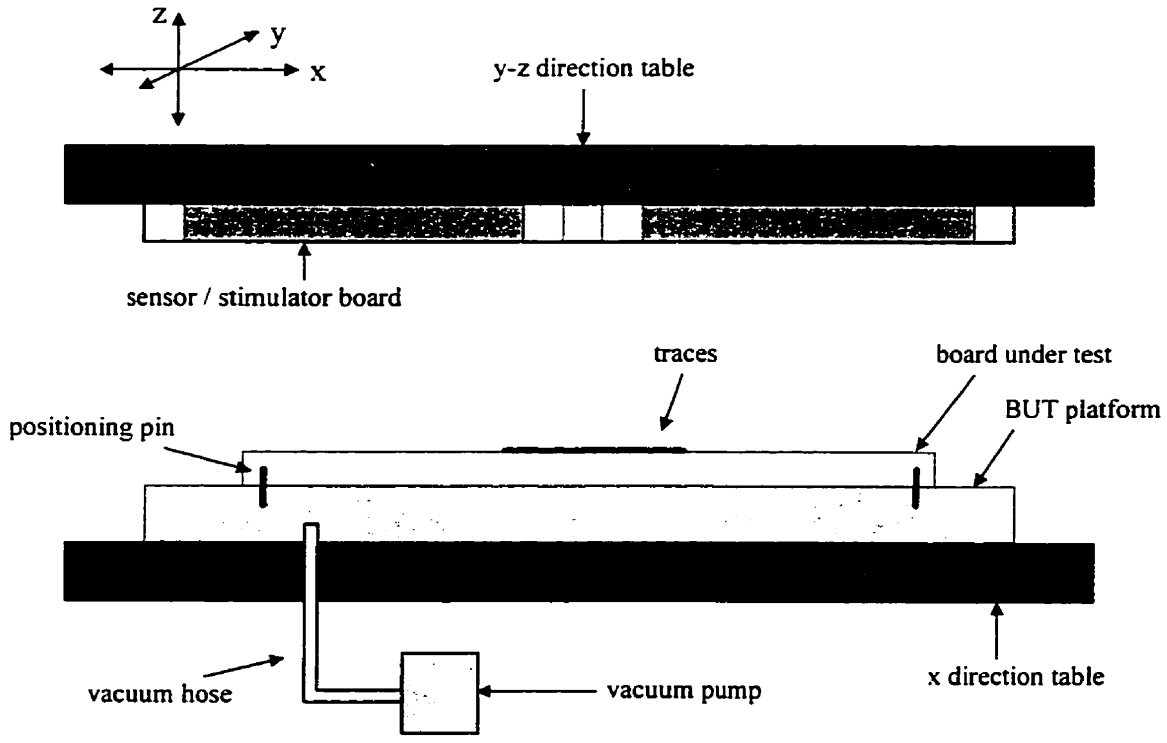


Figure 3-6: Prototype Setup

As shown in the above figure, the SSB is fixed above the BUT and is mounted to a y-z direction table for manual adjustments. The BUT rests on the test board platform, which is connected to the x direction table for motorized adjustments and scanning. A vacuum fixture and positioning pins are used to position and hold the test board in place. The vacuum pump is separate from the frame and the vacuum hose, which is connected to

the platform, is flexible and moves with the scan. The BUT placement and SSB positioning is done by the operator and the scanning is automated via the computer workstation. Pictures of the CTS_128 are illustrated in Figure 3-7.

The CTS_128 system block diagram is shown in Figure 3-8. The computer controls the system initialization, scanning, data collection and analysis by passing instructions to the analog to digital converter board (A/D board) and the motion controller. The A/D board digitizes the DC analog voltage signal from the sensor / stimulator assembly, which is representative of the magnitude of the displacement current from the BUT. Furthermore, the A/D board generates the system clock for timing, sets up the sensor addressing, sets up the stimulator configuration and generates the AC input signal for the stimulators. The sensor / stimulator assembly consists of the SSB, stimulator balancing board and the scan sequencing board. A motor is utilized for automated scanning. A motion controller is used to interface the motor to the computer workstation. An encoder generates information on the motor position. The signal from the encoder is split by the encoder signal splitter and passed to the motion controller indicating the position of the motor and passed to the SSB to initiate scans.

The frequency of operation for the CTS_128 prototype is 250 kilohertz and the stimulator voltage magnitude is approximately 10 volts maximum.

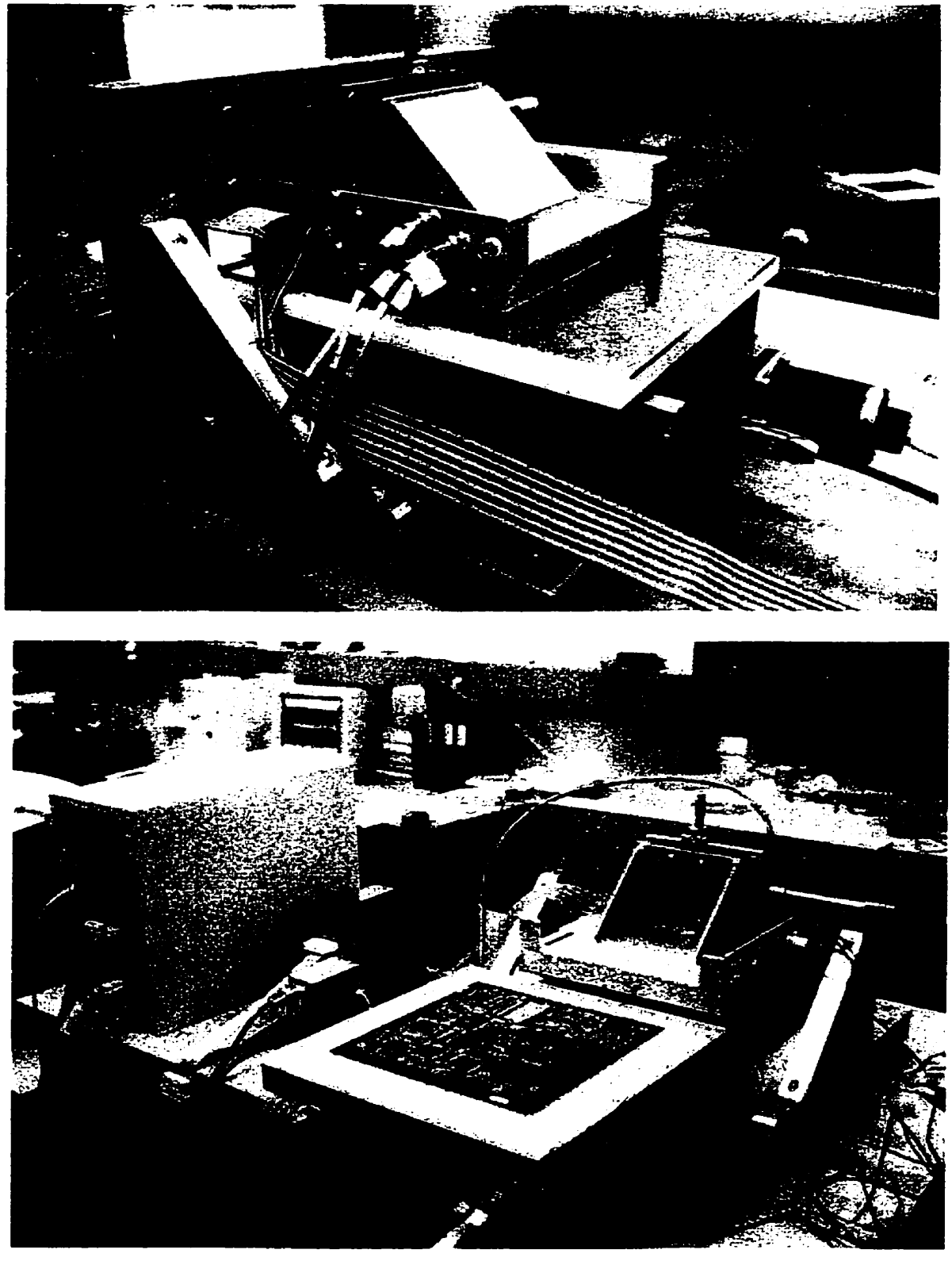


Figure 3-7: CTS_128 Pictures (source: Cirlog)

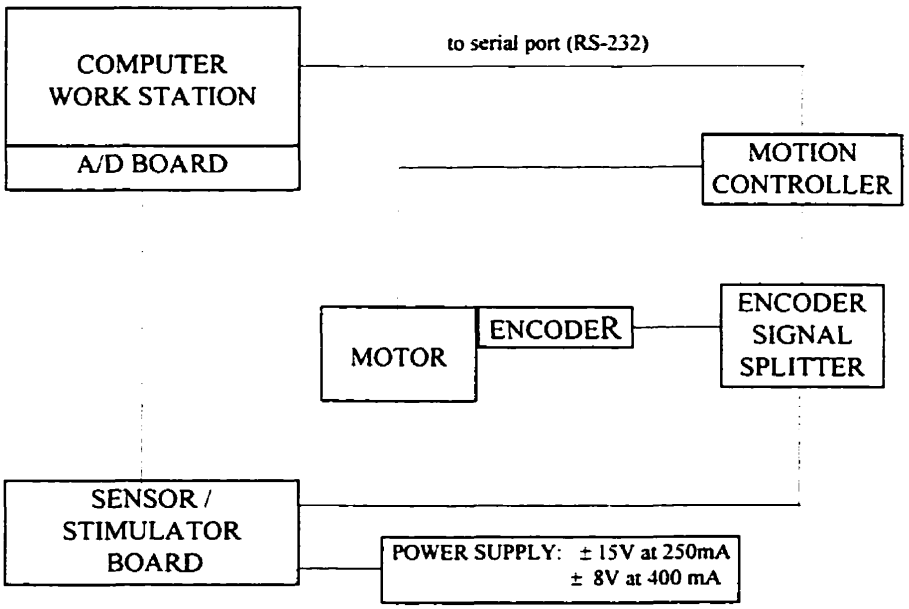


Figure 3-8: CTS_128 System Block Diagram (source: Cirlog)

3.3.1 Sensor / Simulator Board (SSB)

The SSB for the CTS_128 prototype consists of 128 patch sensors and two stimulators. The arrangement is the same as shown in Figure 3-1. Appendix A depicts the actual SSB. The ground strip separating the sensors from the stimulators varies in width linearly across from sensor #0 to sensor #127. A varying strip width design was implemented to study the effects of different spacings between the sensors and stimulators. Each sensor is 0.05 inches square (0.127 mm^2) and the center to center spacing is 0.06 inches. The ground strip width varies from 0.05 inches at sensor #0 to 0.01 inches at sensor #127.

3.3.2 Sensor Signal Processing

The CTS_128 prototype SSB consists of 128 patch sensors. Each patch sensor requires a transimpedance amplifier to detect the displacement current flow. A transimpedance amplifier is a high bandwidth amplifier, which is required to obtain a low input impedance. A low enough input impedance permits the displacement current to flow in the feedback resistor. Thus, the amplifier output voltage is representative of the displacement current flowing through the feedback resistor. Processing of this voltage signal includes multiplexing, filtering, amplification, rectification and integration before it eventually reaches the A/D board, which digitizes the signal for computer analysis.

The 128 sensors are multiplexed down to two channels that are used by the A/D board. Each channel handles 64 sensors and requires the rectification, integration and sample and hold circuitry. Figure 3-9 is a simplified illustration of the sensor to A/D board electronics. Refer to Appendix B for more details on the sensor signal processing. Figure 3-10 is a picture of the CTS_128 Sensor Board electronics.

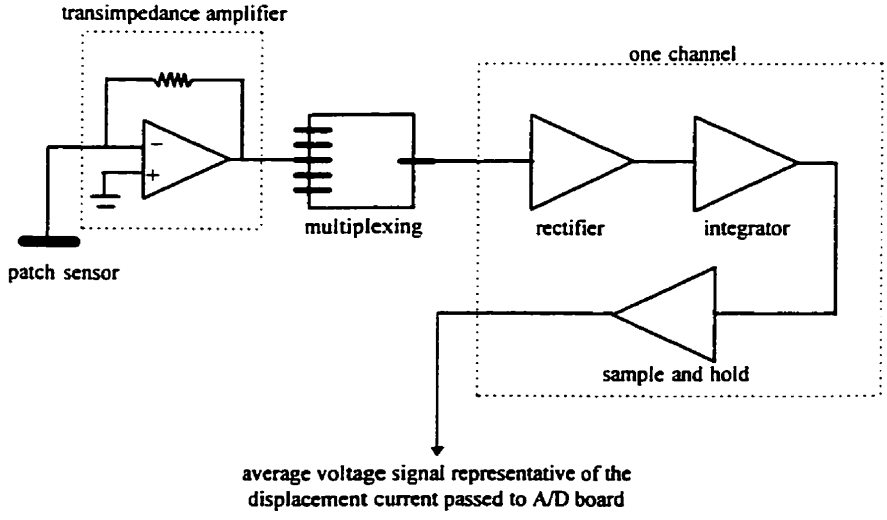


Figure 3-9: Sensor Interface Electronics (source: Cirlog)

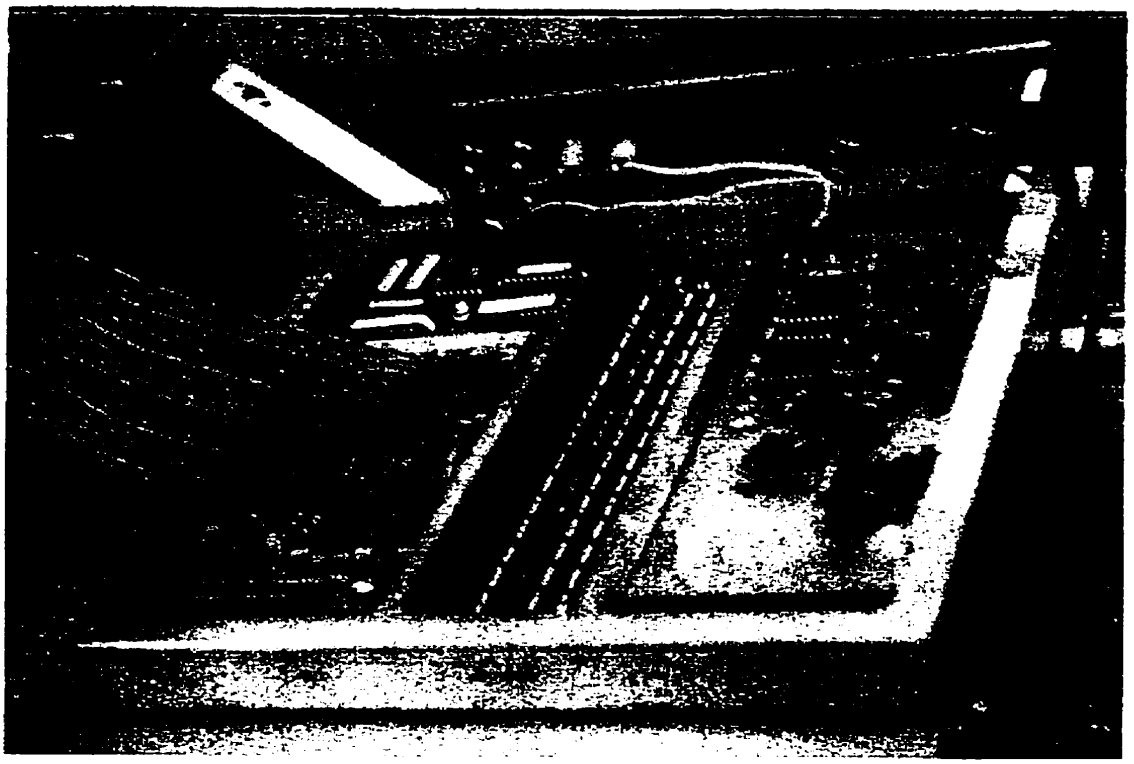


Figure 3-10: CTS_128 SSB Electronics (source: Cirlog)

3.3.3 CTS_128 Scan

The CTS_128 is capable of scanning a BUT with the stimulator plates either "In-Phase" or "Out-Of-Phase". In-Phase means both are supplied by the same AC voltage source and Out-Of-Phase means that the stimulator plates are supplied with 180° phase shift between each AC input to the stimulators. Test results at Cirlog Corporation indicated that the Out-Of-Phase fault detection results are superior and therefore the Out-Of-Phase case is discussed primarily. The figures below illustrate the display images for each case. Figure 3-11 is a picture of a BUT and Figure 3-12 and Figure 3-13 illustrate the Out-Of-Phase and In-Phase images respectively. Notice that the Out-Of-Phase data results in a clearer image of the BUT, with the characteristic nulls located at the center of

the straight track segments. The Out-Of-Phase stimulator arrangement is the key behind the CTS technology, creating a "voltage gradient" along the traces as they are scanned.

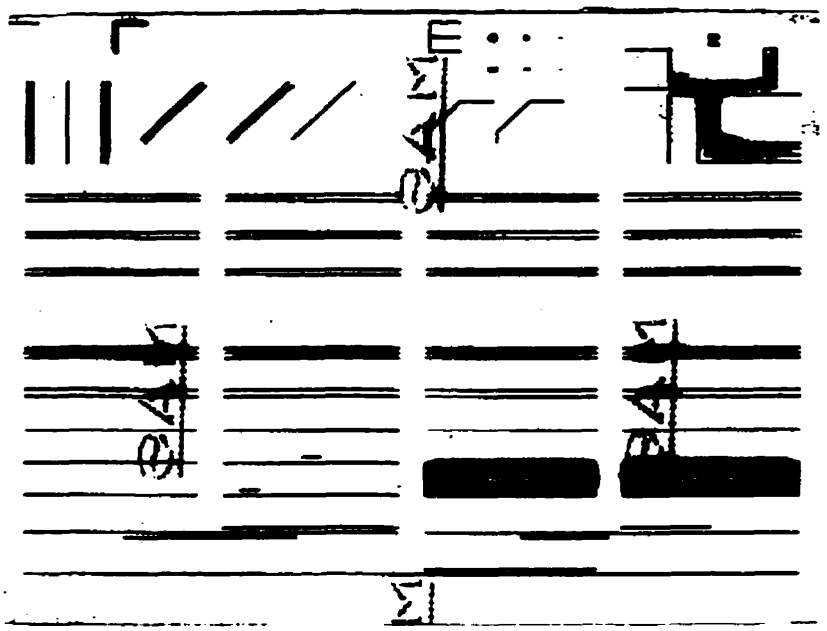


Figure 3-11: Board Under Test

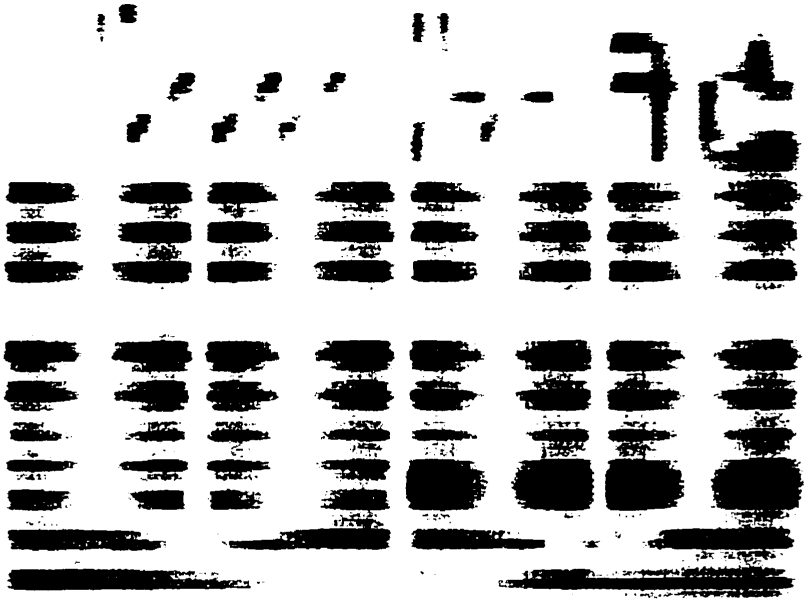


Figure 3-12: Out-Of-Phase Scan - Image Display (source: Cirlog)

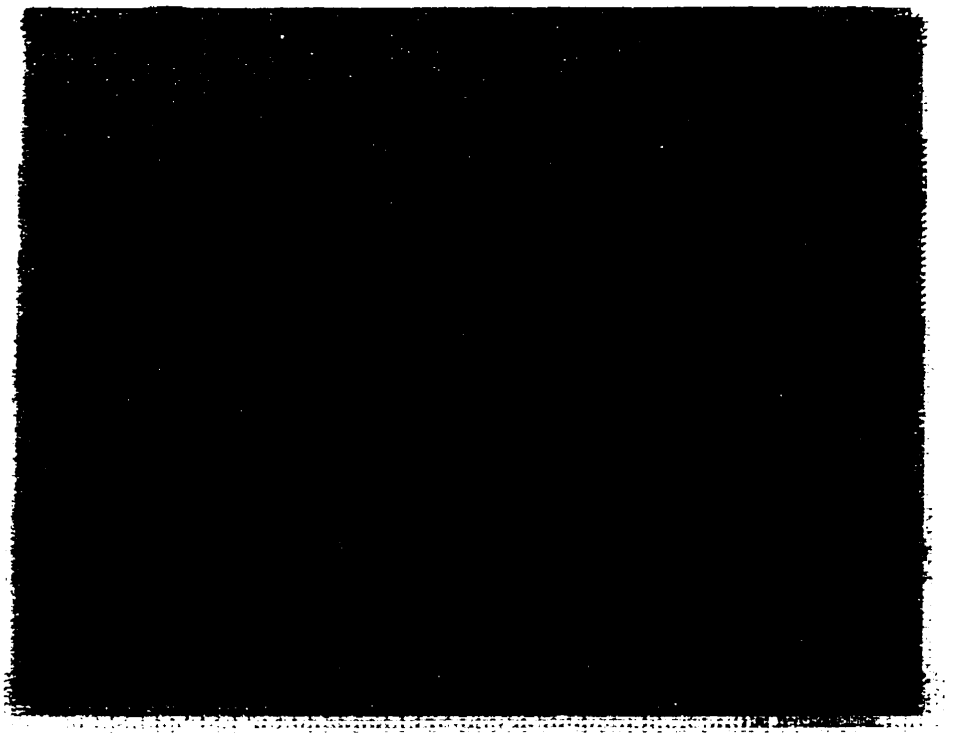


Figure 3-13: In-Phase Scan - Image Display (source: Cirlog)

3.3.3.1 CTS_128 Open Circuit Fault Image

The following illustrations show the changes in an image of a straight track segment on a BUT when an open circuit is made in the center of the track. Figure 3-14 shows the track prior to the fault, Figure 3-15 shows the image with the fault and Figure 3-16 shows a comparison image (absolute value of the difference between the fault and no fault images). Note the drastic change around the center of the track and the detection of the fault. Clearly the characteristic of two track segments are now visible. In the comparison, which is a magnitude comparison, the largest change is around the location of the open circuit. The darker areas indicate the larger voltage levels, the image software depicts the highest voltages in red and the lower voltages in light blue.

Figure 3-14: Straight Track Segment Prior To Fault (source: Cirlog)

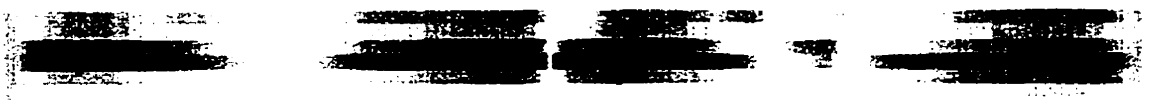


Figure 3-15: Straight Track Segment After Fault (source: Cirlog)

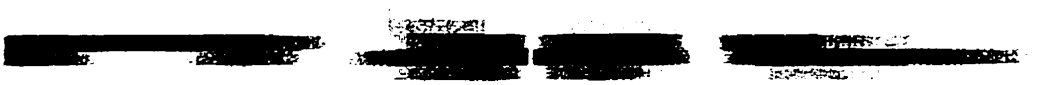


Figure 3-16: Image Comparison (source: Cirlog)

3.4 Sample Experimental Data

3.4.1 Straight Tack Segment

A brief description of experimental data collected from the CTS_128 prototype is given in this section of the report. All sample experimental data is courtesy of Cirlog Corporation. The characteristic for a scan of a straight track segment is given in Figure 3-17. The data was obtained by scanning, from left to right, the entire length of a 1 inch long track segment centered under a sensor. A null in the center of the scan of a straight track segment is characteristic of the Out-Of-Phase stimulators. When a track is exactly centered between the two stimulators the voltage on the track and thus the signal detected by the sensors is theoretically zero, due to the cancellation of field contributions to the

track by each stimulator (track voltage theoretically should be zero volts). The peaks correspond to the track positions when full stimulus from only one stimulator is applied to the track segment and the track is positioned such that its end is directly under the sensor. Figure 3-18 depicts the corresponding track positions in a simplified diagram that is not to scale and ignores the ground strips between the sensors and stimulators.

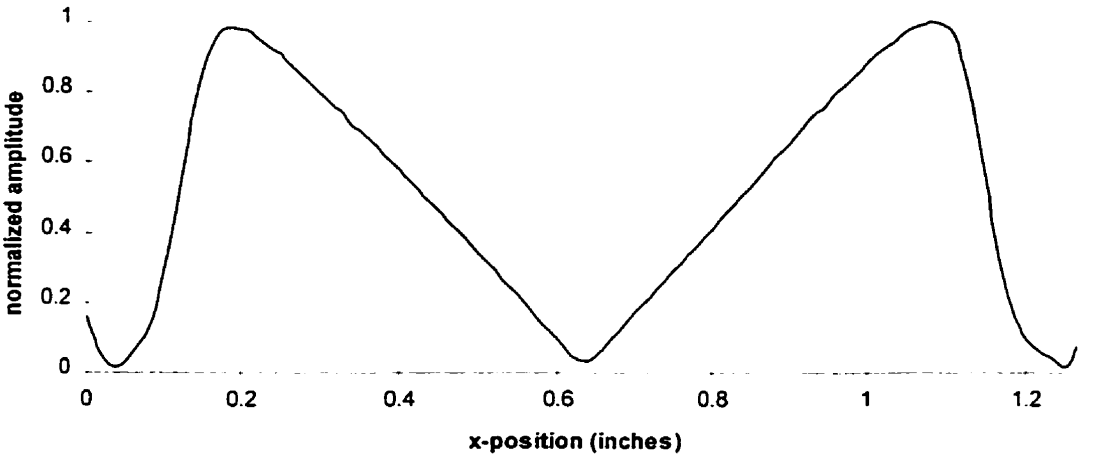
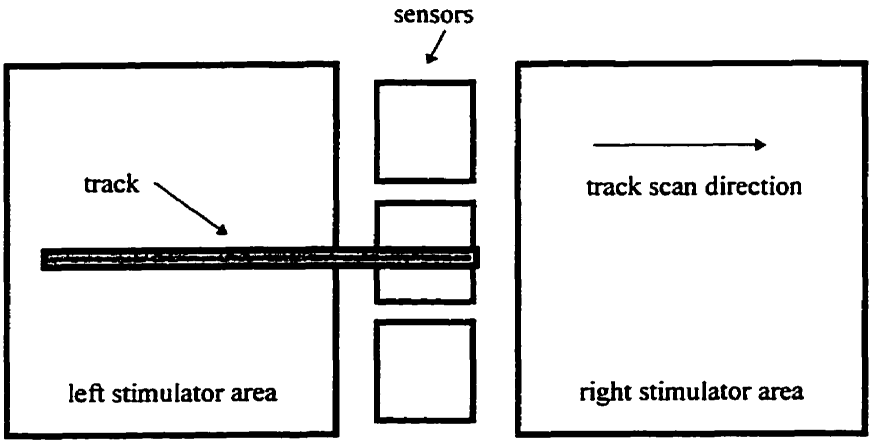


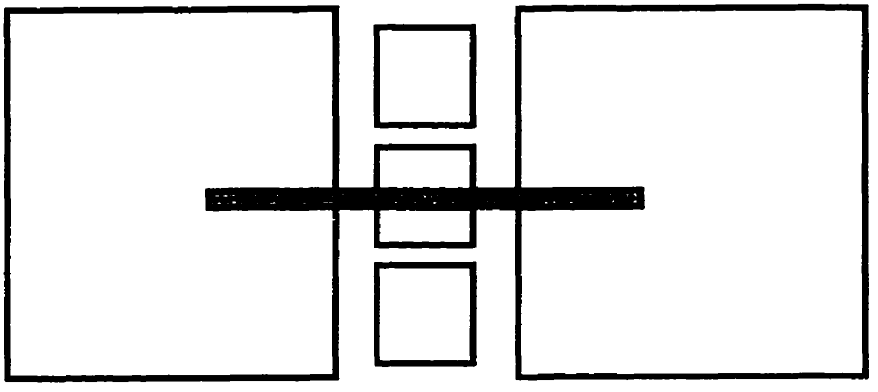
Figure 3-17: Scan Of A Straight Track

3.4.2 Adjacent Sensor Pickup

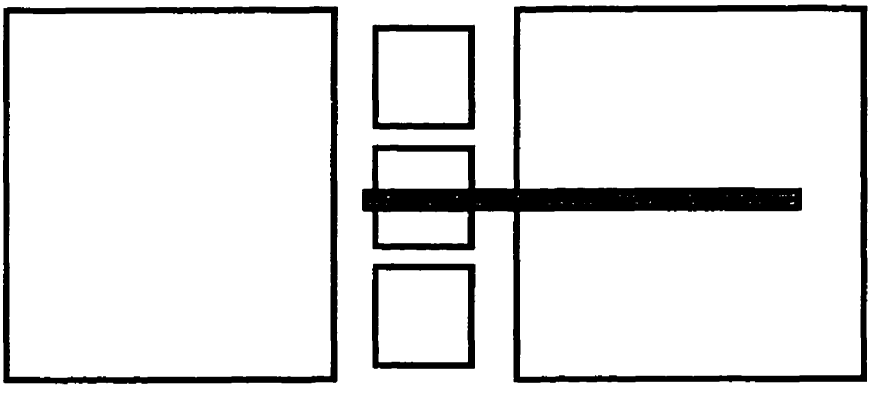
Figure 3-19 illustrates the signal picked up by a sensor even though a track is not located directly below the sensor. The electric field from the track to the sensor board spreads out and is detected by more than one sensor as illustrated in Figure 3-20. The data was obtained from scanning a 1 inch long straight track segment. Each sensor detects a signal from the BUT that may represent contributions from numerous traces.



(a) Left Peak Signal



(b) Center Null Signal



(c) Right Peak Signal

Figure 3-18: Corresponding Track Positions For Scan Peaks And Null

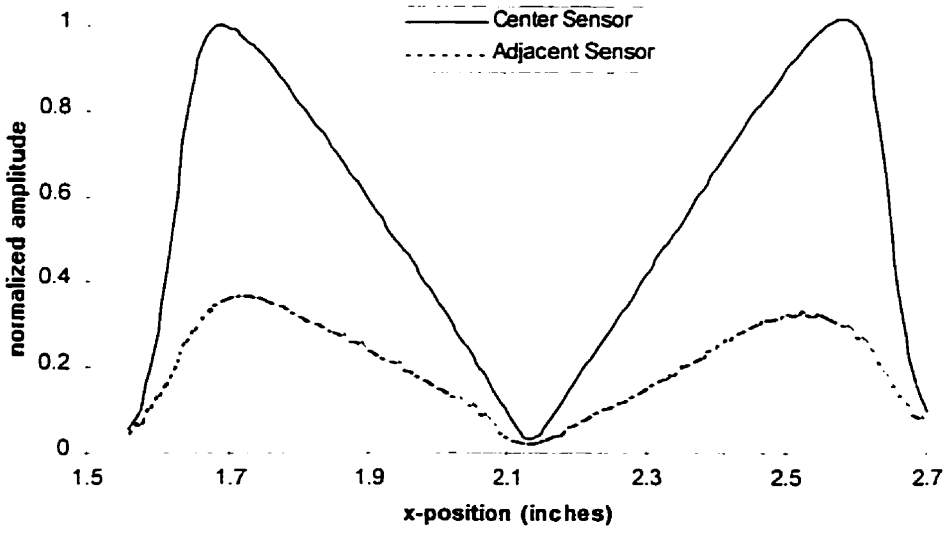


Figure 3-19: Adjacent Sensor Pickup

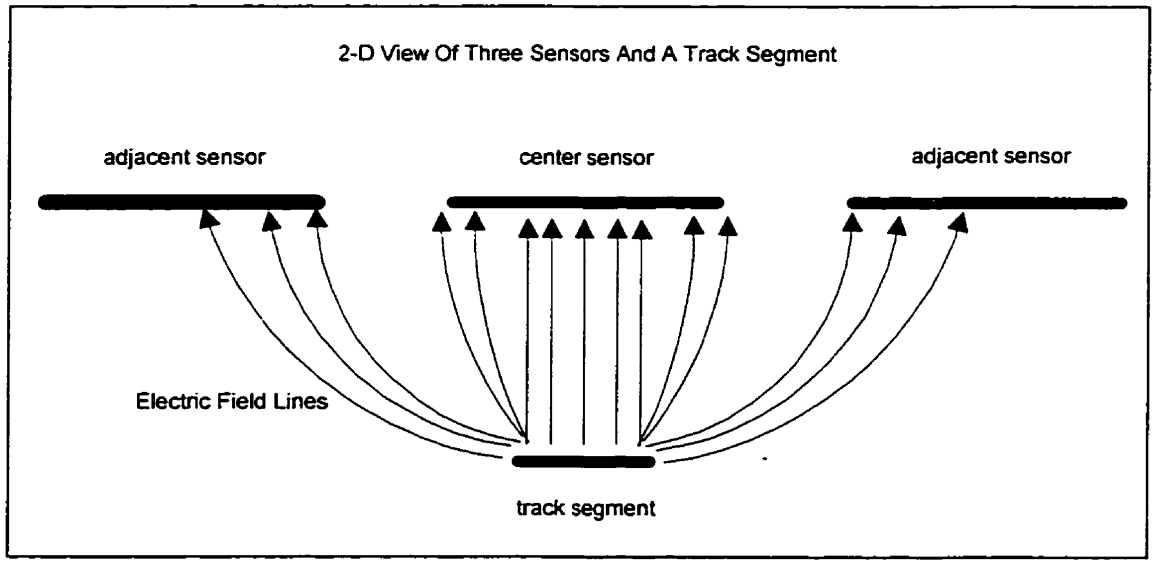


Figure 3-20: Detection Of A Track By Adjacent Sensors

3.4.3 Typical Data

Figure 3-21 illustrates data obtained from an random geometry, which is the type of characteristics obtained when scanning a test board. The sensor picks up the fields radiating from the tracks located directly below and from other tracks located near by. The combined field is detected by the sensors and the geometry of the traces on the BUT is not obvious from studying the data. As a result it is necessary to compare scan data against data from a known good board, or perhaps to a simulation of the BUT, to identify faults. For simple cases, like a straight track segment, an open circuit fault can easily be identified by studying the data.

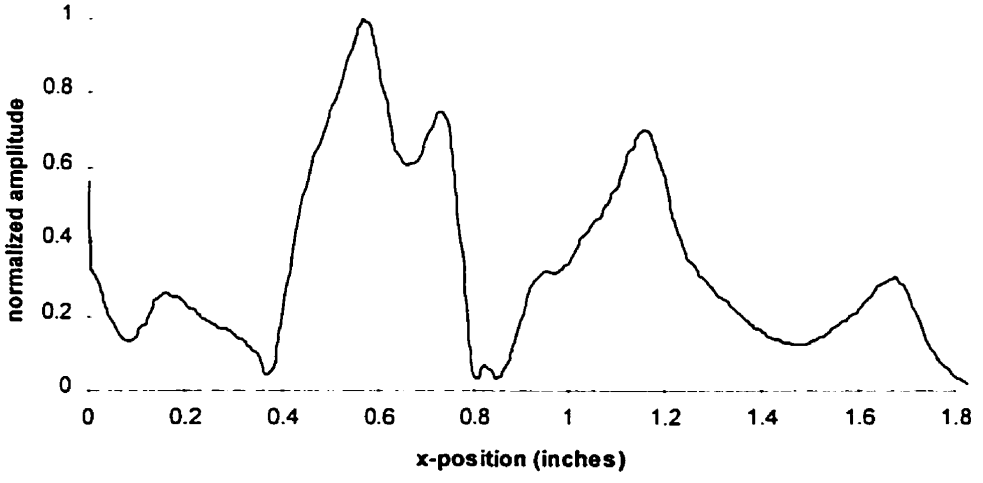


Figure 3-21: Typical Test Data

3.4.4 Open Circuit Data

The data collected is displayed graphically by lining up each row of sensor data and using a color scale to display the changing amplitudes as described in section 3.3.3.

The combined data collected from a scan of a test board is compared sensor row by sensor row to the data from a known good board to detect faults. Various comparison techniques are utilized including simple magnitude comparisons and a slope comparison technique. Figure 3-22 depicts the effect of an open circuit fault on the characteristic of the signal picked up by a sensor. In this case an open circuit was made on a 1 inch long track. The characteristic of the open circuit data appears with two nulls, thus indicating the presence of two straight track segments as expected. By introducing an open circuit, effectively two separate track segments are created. The signature of a magnitude comparison is also shown for reference.

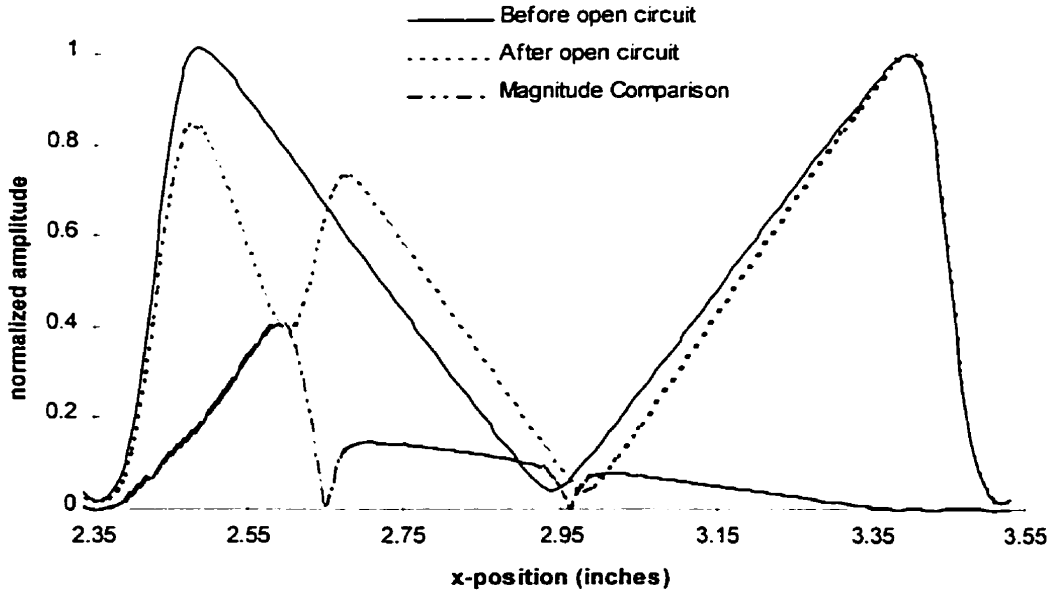


Figure 3-22: Open Circuit Fault

3.4.5 Rectification And Fault Sensitivity

Signal processing on the SSB includes rectification of the signals detected by the sensors and thus both halves of the scan are positive as seen in Figure 3-17. However, testing at Cirlog Corporation concluded that the rectification used in the CTS_128 electronics degrades fault detection characteristics by reducing the magnitude change in signal level changes caused by faults in some cases. If the signal were not rectified, the resulting characteristic shape for a straight track segment would appear as shown in Figure 3-23, for a track that is scanned from left to right and the left stimulator voltage is considered as the positive phase. The data for Figure 3-23 was obtained by changing the sign for half of the scan shown in Figure 3-17, which is equivalent to removing the rectification of the data.

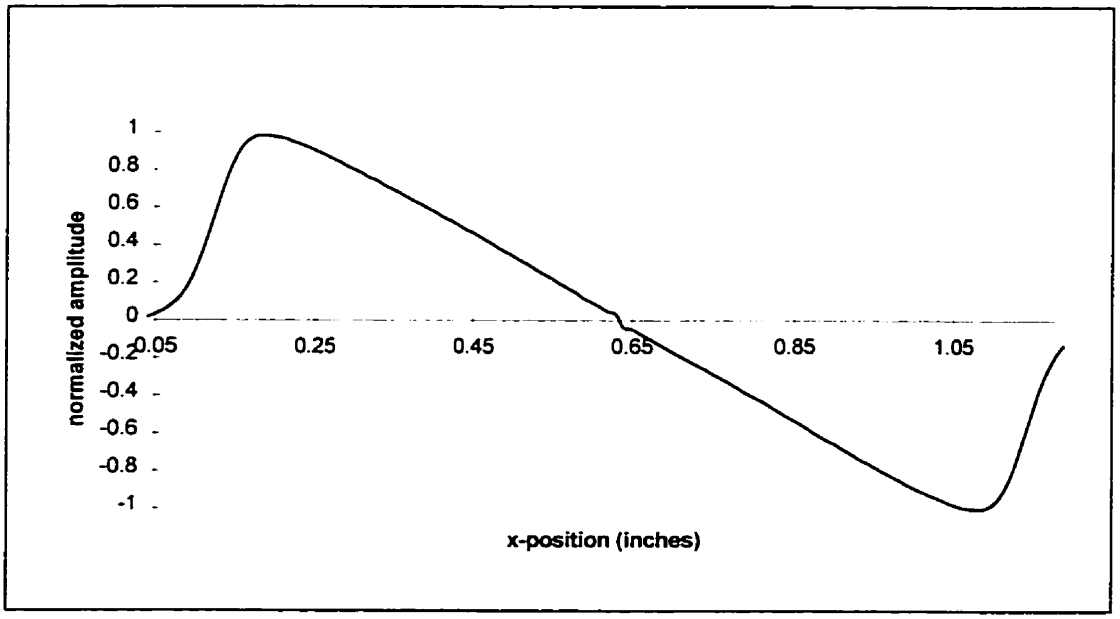


Figure 3-23: Characteristic Of A Straight Track Without Rectification

4. ELECTROSTATIC SIMULATION MODEL

Simulating the CTS technology is important for better understanding the technology in order to predict results. Furthermore, information from simulations aids in optimization of performance, defining the characteristics of a specific geometry, defining the effects of faults and determining the location of faults. The ultimate simulator would be able to generate the Gold Board data, which would replace the requirement of scanning a know good board for test comparisons.

Currently the simulations are used to demonstrate the characteristics of simple geometries and the effects of changing various parameters, such as spacing and dimensions of tracks and sensors. A simulation is performed by constructing an equivalent circuit model and then determining the values of the individual components in the model. The CTS_128 can be modeled using capacitors as shown in Figure 4-1. By calculation of the various capacitance values associated within the model and then applying the values to an equivalent circuit and using nodal or mesh analysis [22], parameters such as the current into the sensor can be calculated. Refer to Chapter 3 for the formulas to calculate the current into the sensor and the voltage on a track.

The capacitance between two conductors is defined as the ratio of the magnitude of the total charge on either conductor to the potential difference between the conductors[19] as given in Equation 4-1.

$$C_{ij} = \frac{Q_i}{V_i - V_j} \tag{4-1}$$

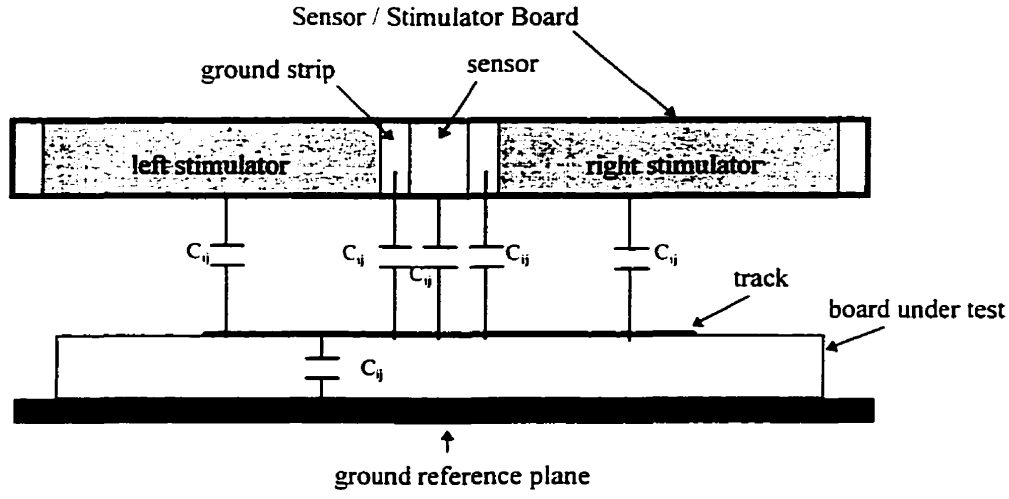


Figure 4-1: Contactless Test System Modeled Using Capacitors

Different methods can be used to determine the various capacitance values, including a parallel plate model, empirical formulas or solving a more complex electrostatic solution by considering Poisson's Equation. The various simulations discussed later, utilized the more accurate MoM technique (sections 4.3 to 4.5) to solve for the capacitance values between the track under test, the sensors and the ground strips. These capacitances corresponds to C_2 , C_3 , C_4 and C_6 as shown in Figure 3-5. Empirical formulas in section 4.2, which are a 2-D analytical method that model the capacitance per unit length and ignore the effects of coupling to other conductors (sensors and ground

strips), where utilized to calculate the capacitance between the track and stimulators. These capacitances correspond to C_1 and C_5 as shown in Figure 3-5. Empirical formulas were utilized to reduce computation time due to the size of the stimulators in comparison to the track and sensors. Furthermore, the more accurate results are required around the track to sensor area where the coupling to other conductors is accounted for using the MoM.

4.1 Parallel Plate

The simplest simulation is an approximation using a parallel plate model to determine the capacitance values, which neglects the effects of fringing fields and coupling to surrounding conductors. The parallel plate capacitance is given by Equation 4-2:

$$C = \frac{\varepsilon A}{d} \tag{4-2}$$

where ε is the permittivity or dielectric constant of the material, A is the area of one plate and d is the separation between plates. Figure 4-2 depicts a parallel plate capacitor.

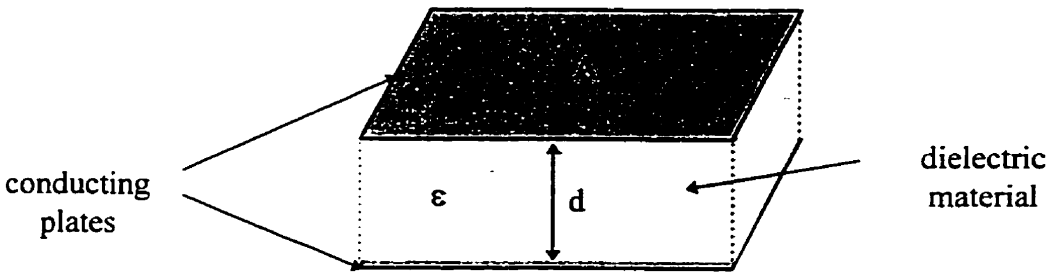


Figure 4-2: Parallel Plate Capacitor

4.2 Empirical Formulas

One technique to account for the effects of fringing fields, but not the coupling to surrounding conductors, is to use Empirical Formulas[23][24][25]. Empirical formulas are those derived from information gathered from practical experimentation. The following formulas take into account the fringing fields and are more accurate than the parallel plate model. Empirical Formulas for capacitance per unit length for a transmission line over a large ground plane (microstip) are given below in Equation 4-3, where the capacitance is in farads per meter. This formula accounts for fringing fields, but ignores the effects of coupling to other conductors (sensors and ground strips). Figure 4-3 depicts the transmission line over a large ground plane (microstrip).

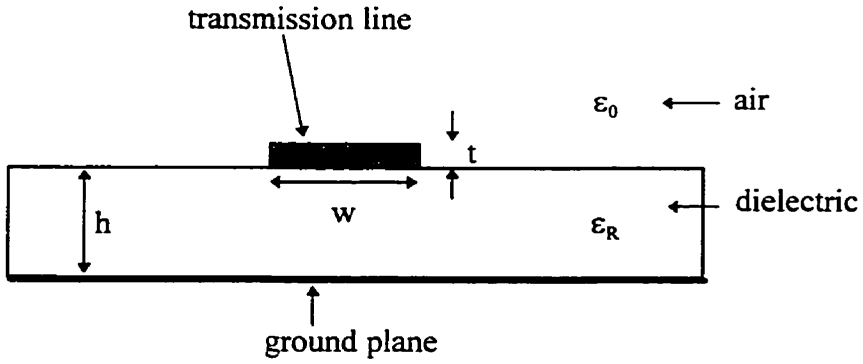


Figure 4-3: Transmission Line Over A Large Ground Plane (microstrip)

The empirical formulas given in Equation 4-3 were applied to the CTS geometry to calculate the capacitance between the track and stimulators, neglecting the effect of other conductors (ground strips, sensors). In order to model the CTS track to stimulator geometry as a microstrip, the stimulator was modeled as the ground plane and the

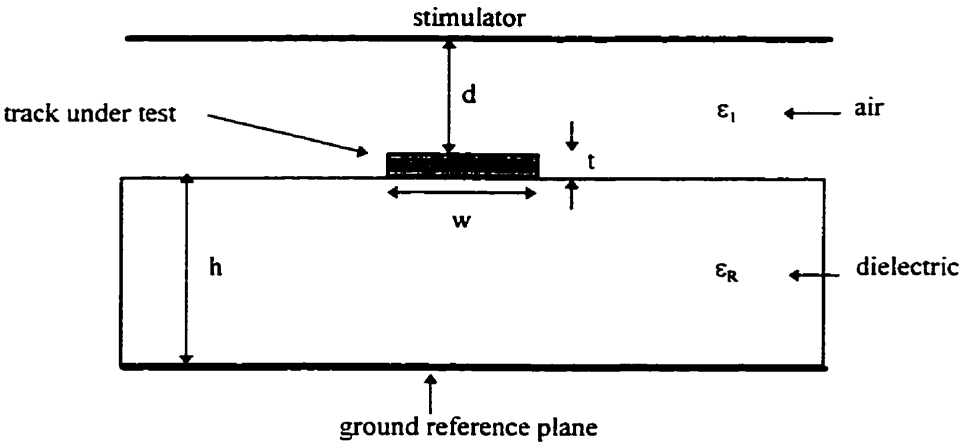
dielectric ε_r was assumed as ε_0 as shown in Figure 4-4B . Furthermore, the effect of the ground reference plane, as shown in Figure 4-4A, was neglected since the track to stimulator distance (d) is smaller than the distance from the track to the ground reference plane (h).

Microstrip empirical formulas valid for $t / h \leq 1/100$:

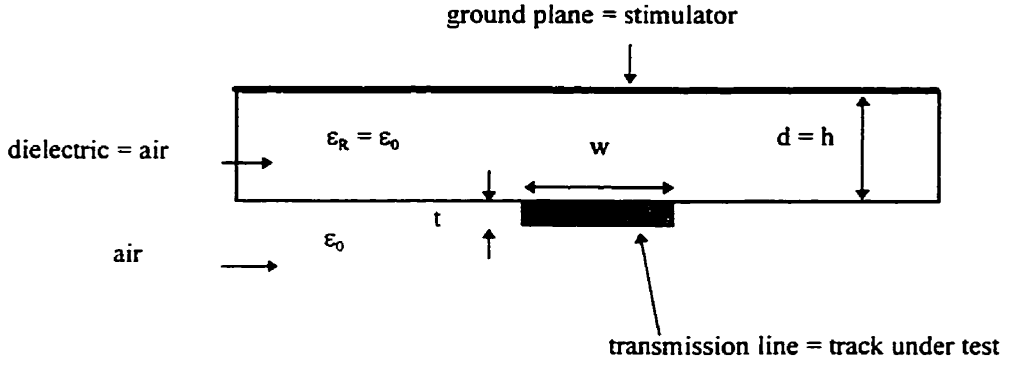
$$C = \begin{cases} \dfrac{2\pi \varepsilon_0 \varepsilon_{reff}}{\ln\left[\dfrac{8h}{w} + 0.25\dfrac{w}{h}\right]} \; ; \; \dfrac{w}{h} \leq 1 \\ \varepsilon_0 \varepsilon_{reff}\left[\dfrac{w}{h} + 1.393 + 0.667 \ln\left[\dfrac{w}{h} + 1.444\right]\right] \; ; \; \dfrac{w}{h} \geq 1 \end{cases} \qquad (4\text{-}3)$$

where $$\varepsilon_{reff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left(1 + 12\frac{h}{w}\right)^{-\frac{1}{2}}$$

A more accurate representation that could also model the ground reference plane, would have been to utilize the empirical formulas for a non-symmetric stripline. A non-symmetric stripline has a transmission line embedded between two ground planes that is closer to one ground plane. For the case where the track to stimulator distance equals the track to ground plan distance (d=h), an empirical formula for a symmetric stripline would be applicable [25]. Many Empirical Formulas can be found for different geometries, however the best simulation results are obtain by solving the governing field equations for a specific problem. For the CTS the electric fields are considered electrostatic in nature.



(A) 2-D VIEW OF TRACK TO STIMULATOR



(B) 2 -D MODEL OF TRACK TO STIMULATOR AS MICROSTIP

Figure 4-4: Applying Empirical Formulas To The CTS

4.3 Electrostatic Solutions

The CTS technology can be considered as an electrostatic problem [19][20]. In electrostatics there are no charges or electric fields within a conductor, but charge may appear on the conductor surface as a surface charge density. The electric charges are considered at rest and the electric fields do not change with time. Furthermore, there are no magnetic fields.

The physical quantities of interest in electrostatics are charge, electric field and the electrostatic potential. Poisson's equation relates these quantities and is given below in Equation 4-4 for a homogeneous region:

$$\nabla^2 \phi = - \frac{\rho}{\varepsilon} \tag{4-4}$$

In equation 4-4: ϕ is the electric potential, ρ is the volume density of free charges and ε is the permittivity of the medium, which is a constant if the medium is homogeneous. If volume charge density is zero ($\rho = 0$), but point charges, line charges and surface charge density are allowed to exist at singular locations as the field sources, then Poisson's equation reduces to Laplace's equation [19][20]:

$$\nabla^2 \phi = 0 \tag{4-5}$$

Poisson's equation and Laplace's equation are solved by applying the boundary conditions relating to the particular electrostatic problem.

The electrostatic problem for the CTS technology involves solving either Poisson's or Laplace's equation on the surface of the conductors subject to boundary conditions. An integral equation approach is utilized to solve the CTS. Solving the CTS electrostatic problem consists of solving a boundary value problem by transforming the differential operators defined in the domain to integral operators defined on the boundary. The resulting integral equations are solved using the Method Of Moments (MoM) [26], which reduces the integral equations to a system of linear algebraic equations in terms of

the unknowns. Section 4.4 and Figure 4-5 outline the formulation of the electrostatic solution in a homogeneous region (free space). Section 4.5 outlines the formulation of the electrostatic solution when considering dielectric layers.

Other techniques for solving electrostatic field problems, Finite Element Method (FEM) and Finite Difference Time Domain (FDTM). operate on the partial differential equations rather than integral equations [27][28].

4.4 Free Space Electrostatic Solution

An integral equation approach is utilized to solve the electrostatic problem and the Method Of Moments (MoM) is used to solve the resulting integral equation. The MoM procedure involves using a simple subsectioning and point-matching solution. Pulse subsectional basis functions are used to represent the charge density and Dirac Delta functions are used as weighting (testing) functions. The integral equations are obtained from assuming point charges as the sources of the electric field and solving Poisson's equation subjected to boundary conditions.

Solving the electrostatic CTS geometry, as shown in Figure 3-5, in a homogeneous region (free space) utilizing an integral equation approach and using the MoM technique is as follows:

1. Consider the following solution to Poisson's Equation (assuming point charges as sources of the electric field and using image theory [19] to account for ground planes):

$$\phi\,(x,y,z) = \iiint \frac{\rho(x',y',z')}{4\,\pi\,\varepsilon\;R}\,dx'dy'dz'$$

where (4-6)

$$R = \sqrt{(x-x')^2 + (y-y')^2 + (z-z')^2} - \sqrt{(x-x')^2 + (y-y')^2 + (z+z')^2}$$

R is the distance from the source point to the observation point minus the distance from the image point to the observation point.

2. MoM requires that the unknown charge density (ρ) be expanded as a linear combination of N terms.

$$\rho(x',y',z') = \alpha_1 f_1 + \alpha_2 f_2 + \alpha_N f_N = \sum_{n=1}^{N} \alpha_N f_N \qquad (4\text{-}7)$$

$$f_N = \begin{cases} 1 & \text{on conductor of interest} \\ 0 & \text{elsewhere} \end{cases}$$

Each α is an unknown constant and each f_N is a known function usually referred to as a basis or expansion function. A pulse basis function is chosen.

3. Using a subdomain approach, the geometry is subdivided into N patches and a pulse function is chosen as the basis function. The boundary condition that the electrostatic potential is 1 volt on the conductor is applied. The total charge on each segment is assumed to be concentrated at the center of the segment by using Dirac Delta weighting functions. The MoM is a numerical technique whose solution only satisfies the boundary conditions at discrete points. Between these points the boundary conditions may not be satisfied and the deviation is referred to as a residual. To minimize the residual in such a way that its overall average over the entire

structure approaches zero, the method of weighted residuals is utilized in conjunction with the inner product of the weighted residual and the response function (charge density). This forces the boundary conditions to be satisfied in an average sense over the entire surface. The use of Dirac Delta weighting functions is seen as the relaxation of boundary conditions so that they are enforced only at discrete points on the surface of the structure.

4. Substituting the charge density representation (Equation 4-7) into the integral equation (Equation 4-6) results in one equation in N unknowns:

$$\phi\,(x,y,z) = V(x,y,z) = 1 = \frac{1}{4\,\pi\,\varepsilon_0}\sum_{n=1}^{N}\frac{q_n}{\text{area}_n}\iiint\frac{1}{R}\,dx'dy'dz' \qquad (4\text{-}8)$$

Furthermore, since each patch is assumed to have no thickness since the charge is concentrated on the surface the integral part, which is the Free Space Green's Function, reduces to:

$$\iiint\frac{1}{R}\,dx'dy'dz' = \iint\frac{1}{R}\,dx'dy' = \frac{1}{R}\times(\text{area of the patch}) \qquad (4\text{-}9)$$

The Green's function gives the effect of the source charge on the potential at the observation point. When the source and observation point are the same, an approximate formula is used to solve the integration. The patch subsections are approximated by circular subsections of the same area for the primary contribution[26]. The image contribution is calculated as before.

$$\iint\frac{1}{R}\,dx'dy' \cong 4\,\pi\,(0.282)\sqrt{\text{area of patch}} - \frac{\text{area of patch}}{2\,z} \qquad (4\text{-}10)$$

5. By point matching or evaluating Equation 4-6 at N different points (observation points), N equations with N unknowns are obtained. Each equation has the following form:

$$1 = q_1 z_{11} + q_2 z_{12} + q_3 z_{13} + \ldots\ldots + q_N z_{NN} \qquad (4\text{-}11)$$

   The z elements are known and the q elements are the unknowns to be determined.

6. Matrix operations, LU decomposition and forward and backward substitution [29], are used to solve for the unknown charges, actually the unknowns are solved in charge density form and then converted to charge. Finally the capacitance values are determined by summation of the charge divided by the potential difference (1 volt).

A program was written in C code for performing the method described above and a sample program is given in Appendix D for reference. By running the program for a specified geometry, the capacitance values between the system of conductors are determined.

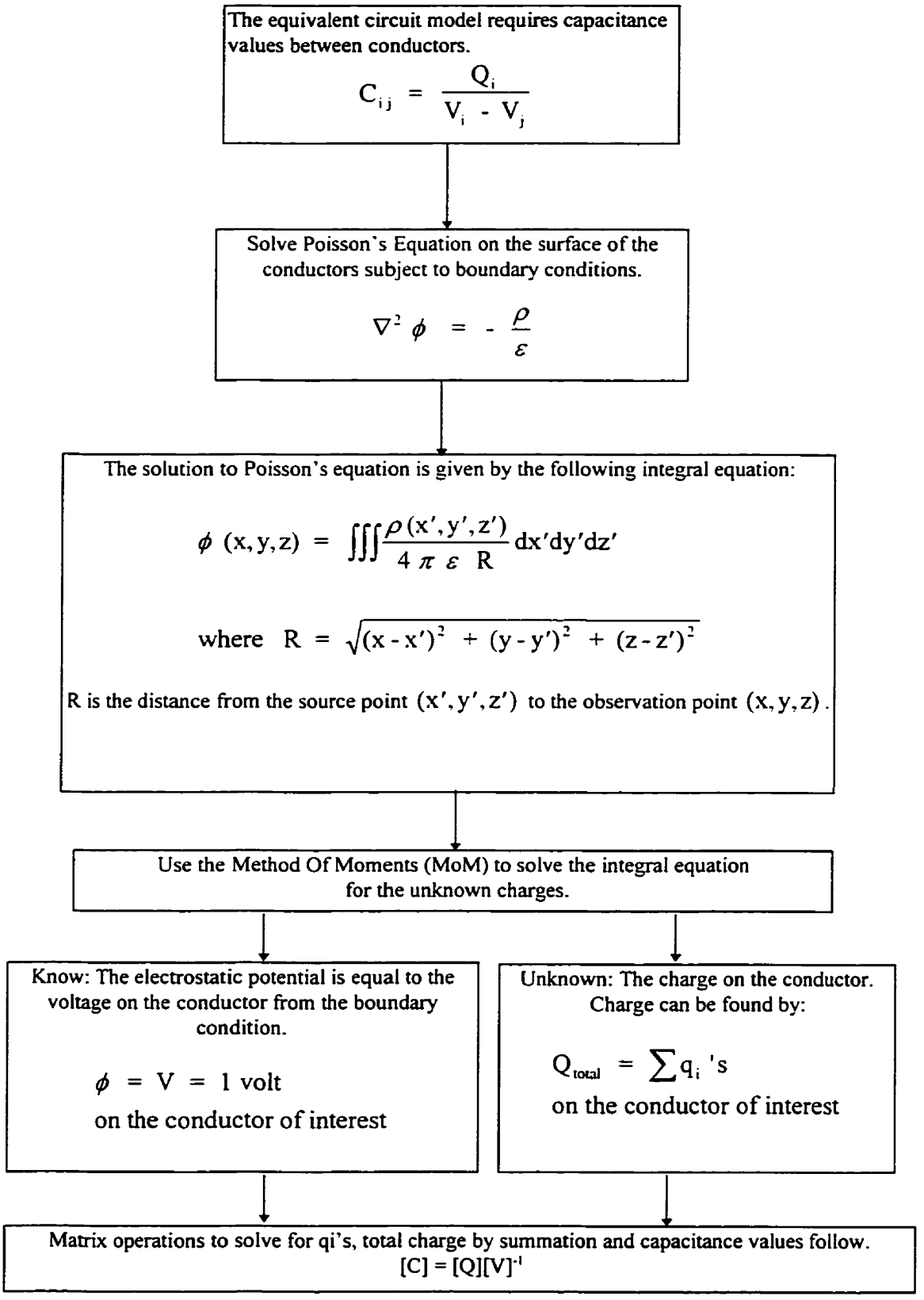


Figure 4-5: CTS Electrostatic Problem Solution - Homogeneous Region

4.5 Dielectric Region Simulations

The simulations described above accounted for the effects of fringing fields and the coupling between all the conductive elements. An important factor that was neglected is the effect of different dielectric regions in the geometry. In the above discussions, the region was assumed homogeneous in air (free space). Practically, different layers of dielectric should be modeled into the geometry to model the CTS more accurately. The dielectric layers should be considered:

- sensor board has a dielectric coating around the sensors and stimulators
- tracks on the board under test are covered by a dielectric coating
- board under test has a dielectric material separating it from the ground reference plane

In order in incorporate the effects of dielectric regions, a more complicated electrostatic boundary value problem must be solved. The resulting integral equations are more complicated than the integral equation solved in the free space case, Equation 4-6. However, the Method of Moments technique is used in the same way to solve the new more complicated integral equations.

By modeling the CTS geometry in free space many factors can be studied to gain insight into the general characteristics of different geometries. However, if simulations are required to generate data for actual test comparisons a better model incorporating the effects of the dielectric regions is probably required.

4.5.1 Dielectric Simulation Theory

This section of the report describes the theory for simulations when a dielectric layer is introduced into the CTS system of conductors geometry. Formulating the integral equations to include the effects of the dielectric layers requires solving a complicated electrostatic boundary value problem. Furthermore, depending on which dielectric layer is under consideration, the corresponding model or geometry must be solved. In considering the CTS technology, the dielectric layers that apply include the coating around the SSB PCB, the dielectric coating on the traces on the BUT and the dielectric material separating the BUT from the ground reference plane.

The geometries, as shown in Figure 4-6, Figure 4-9 and Figure 4-10, for which boundary value problem solutions are required, include the following:

- A charge q located over a grounded dielectric slab ($z > 0$)
- A charge q located within a grounded dielectric slab ($0 > z > -d$)
- A charge q located at the interface between the grounded dielectric slab and free space ($z = 0$)

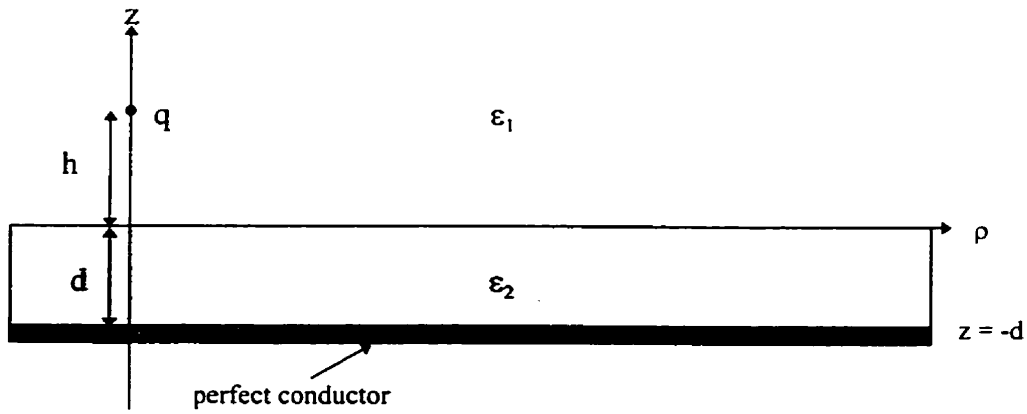


Figure 4-6: Charge q Located Over A Grounded Dielectric Slab

Figure 4-6 illustrates a perfect conductor located at z = -d and a dielectric interface located at z = 0. The region z >0 has a permittivity of ε_1 and the region 0 > z > -d has a permittivity of ε_2. A point charge (q) is located at (ρ,z) coordinate (0,h).

4.5.2 Formulation Of Integral Equations Including Dielectric Regions

The potential at any point in space due to the point charge can be found by solving Laplace's Equation subject to certain boundary conditions [19][20]. Using cylindrical coordinates (ρ,ϕ,z), the potential V(ρ,z) satisfies Laplace's equation (except right at the point charge):

$$\frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho \frac{\partial V}{\partial \rho}\right) + \frac{\partial^2 V}{\partial z^2} = 0 \qquad (4\text{-}12)$$

where the component $\frac{\partial}{\partial \phi} = 0$ due to azimuthal symmetry

A solution to the above equation for the case of a charge q located over a grounded dielectric slab, as shown in Figure 4-6, requires formulating equations for each of the two regions of dielectric material [30]. The potential can then be determined everywhere above the perfect conductor located at z = -d.

The appropriate form of the potential solutions is given by the following equations where the source charge is located at (0,h) and the observation point is (ρ,z).

For the region z > 0:

$$V(\rho, z) = \frac{q}{4\pi\varepsilon_1} \int_0^\infty \left[e^{-\lambda|z-h|} + R(\lambda)\, e^{-\lambda(z+h)} \right] J_o(\lambda\rho)\, d\lambda \qquad (4\text{-}13)$$

For the region 0 > z > -d:

$$V(\rho, z) = \int_0^\infty \left[A(\lambda)\, e^{-\lambda z} + B(\lambda)\, e^{\lambda z} \right] J_o(\lambda\rho)\, d\lambda \qquad (4\text{-}14)$$

These equations are formulated as follows. The fields of sources in the presence of disturbing bodies can be estimated by introducing images of the source (Image Theory) [19][30][31]. In considering the region z >0, an image charge must be located at z = -h . The image charge is assumed equal to Kq, where K is not determined yet. Figure 4-7 shows the location of the image charge.

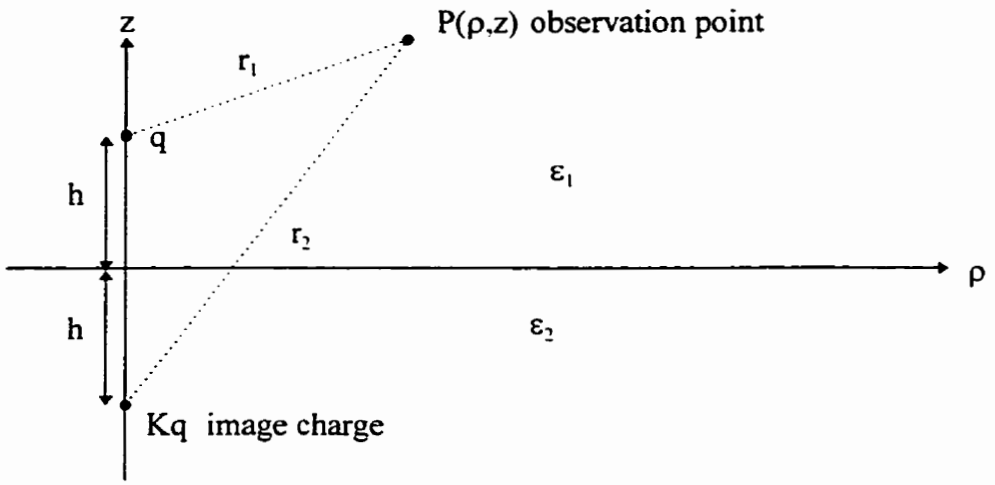


Figure 4-7: Image Charge Locations

NOTE: K = -1 for the lower region being a conductor and K = 0 for the lower region (z < 0) being the same as the upper region (z > 0).

The potential at any point in the region z > 0, due to the source point charge q can be given by:

$$V(\rho, z) = \frac{q}{4\pi\varepsilon_1}\left(\frac{1}{r_1} + K\frac{1}{r_2}\right)$$

where (4-15)

$$r_1 = \sqrt{\rho^2 + (|z-h|)^2}$$
$$r_2 = \sqrt{\rho^2 + (z+h)^2}$$

In considering the region 0 > z > -d, the charge q is viewed as having changed effective source strength equal to a value Tq, where T is not determined yet. The charge Tq is referred to as the modified source. Figure 4-8 shows the modified source charge.

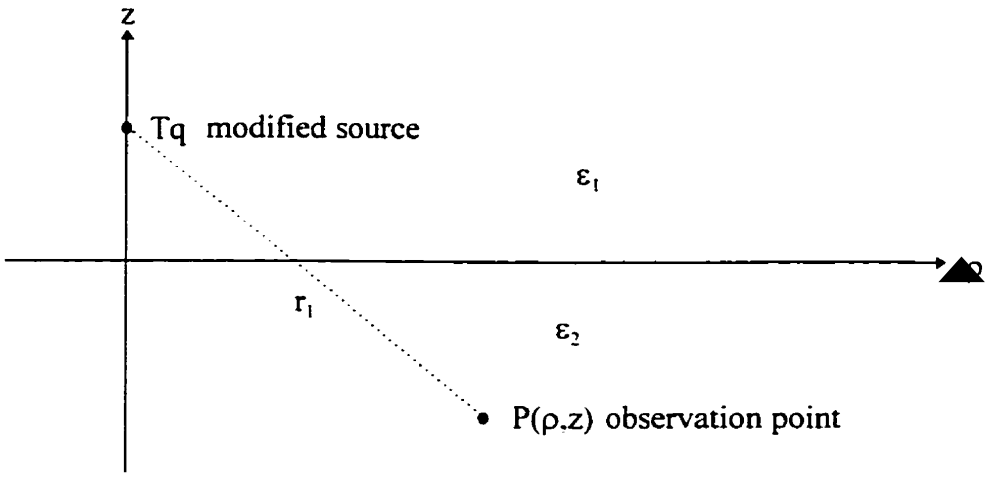


Figure 4-8: Modified Source Charge

The potential at any point in the region 0 > z > -d, due to the modified source point charge Tq can be given by:

$$V(\rho, z) = \frac{Tq}{4\pi\varepsilon_2}\frac{1}{r_1}$$

where (4-16)

$$r_1 = \sqrt{\rho^2 + (-(z-h))^2}$$

The unknown values T and K are determined by applying boundary conditions for the specific geometry under consideration.

By utilizing the Weber integral identity the potential equations are transformed to the form in Equations 4-13 and 4-14. The Weber integral identity is given by:

$$\frac{1}{\sqrt{(1 + y^2)}} = \int_0^\infty J_o(x)\, e^{-yx}\, dx \qquad (4\text{-}17)$$

for $y > 0$

4.5.2.1 Applying Boundary Conditions to the Potential Equations

Referring to equations 4-13 and 4-14, boundary conditions are applied to determine the unknown functions of λ (R,A and B) [30]. At the perfectly conducting base at $z = -d$ the boundary condition for a conductor surface indicates that the tangential electric field is equal to zero ($E_{tan} = 0$):

$$E_\rho \Big|_{z=-d} = -\frac{\partial V}{\partial \rho} \Big|_{z=-d} = 0 \qquad (4\text{-}18)$$

Applying this boundary condition to Equation 4-14 the following result is obtained:

$$A(\lambda) = -B(\lambda)\, e^{-2\lambda d} \qquad (4\text{-}19)$$

Thus, one unknown is eliminated and Equation 4-14 is reduced to:

$$V(\rho, z) = \int_0^\infty A(\lambda)\,(e^{-\lambda z} - e^{\lambda(z-2d)})\, J_o(\lambda\rho)\, d\lambda \qquad (4\text{-}20)$$

At the dielectric interface the boundary condition stating the tangential electric field is continuous applies:

$$E_{tan\,1} = E_{tan\,2}$$

where (4-21)

$$E_{tan} = -\frac{\partial V}{\partial \rho} \Big|_{z=0}$$

Applying this to Equations 4-13 and 4-20 the following result is obtained:

$$\frac{q}{4\pi\varepsilon_1}\left(e^{-\lambda h} + Re^{-\lambda h}\right) = A\left(1 - e^{2\lambda d}\right) \tag{4-22}$$

At the dielectric interface the boundary condition stating the normal flux density is continuous applies:

$$D_{N1} = D_{N2}$$

where (4-23)

$$D_N = \varepsilon E_N = \varepsilon \frac{\partial V}{\partial z}\Big|_{z=0}$$

Applying this to Equations 4-13 and 4-20 the following result is obtained:

$$\frac{q}{4\pi\varepsilon_1}\left(e^{-\lambda h} - Re^{-\lambda h}\right) = -A\left(1 + e^{2\lambda d}\right)\frac{\varepsilon_2}{\varepsilon_1} \tag{4-24}$$

Dividing Equation 4-23 by Equation 4-21 the following result is obtained:

$$R = -\left[\frac{1 - \frac{\varepsilon_1}{\varepsilon_2}\tanh(\lambda d)}{1 + \frac{\varepsilon_1}{\varepsilon_2}\tanh(\lambda d)}\right] \tag{4-25}$$

Now that the unknown functions in the potential equations have been determined, the integral equations are complete and ready to be solved using the Method Of Moments.

4.5.2.2 Arbitrary Source And Observation Point Configurations

The integral equations for the case of a charge q located above the dielectric slab were described in detail in the previous section. The equations for the other geometries are listed here for reference[31].

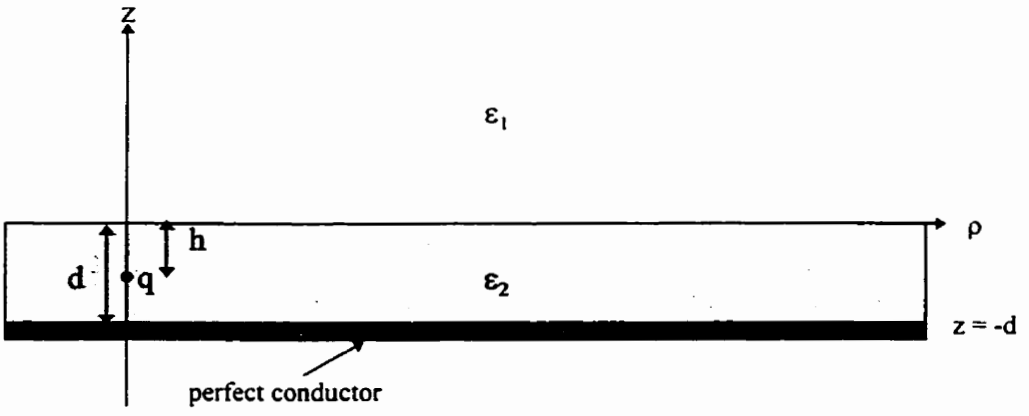


Figure 4-9: Charge q Located Within A Grounded Dielectric Slab

For a charge q located within the dielectric slab, as shown in Figure 4-9, the integral equations for the potential are given by:

For the region z > 0:

$$V(\rho, z) = \int_0^\infty \left[C(\lambda)\, e^{-\lambda z} \right] J_o(\lambda \rho)\, d\lambda \qquad (4\text{-}26)$$

For the region 0 > z > -d:

$$V(\rho, z) = \int_0^\infty \left[\frac{q}{4 \pi \varepsilon_2} e^{-\lambda |z-h|} \quad D(\lambda)\, e^{-\lambda z} + E(\lambda)\, e^{\lambda z} \right] J_o(\lambda \rho)\, d\lambda \qquad (4\text{-}27)$$

Upon applying the appropriate boundary conditions, the unknown values are found to be:

$$C = \frac{q}{4\pi\varepsilon_2}\left[\frac{2(\cosh(\lambda h)\tanh(\lambda d) + \sinh(\lambda h))}{1 + \frac{\varepsilon_1}{\varepsilon_2}\tanh(\lambda d)}\right] \tag{4-28}$$

$$D = \frac{q}{4\pi\varepsilon_2}\left[\frac{2(\frac{\varepsilon_1}{\varepsilon_2}\sinh(\lambda h) - \cosh(\lambda h))}{(1 + e^{2\lambda d})(1 + \frac{\varepsilon_1}{\varepsilon_2}\tanh(\lambda d))}\right] \tag{4-29}$$

$$E = \frac{q}{4\pi\varepsilon_2}\left[\frac{2(\cosh(\lambda h) - \frac{\varepsilon_1}{\varepsilon_2}\sinh(\lambda h)}{(1 + e^{-2\lambda d})(1 + \frac{\varepsilon_1}{\varepsilon_2}\tanh(\lambda d))} - e^{-\lambda h}\right] \tag{4-30}$$

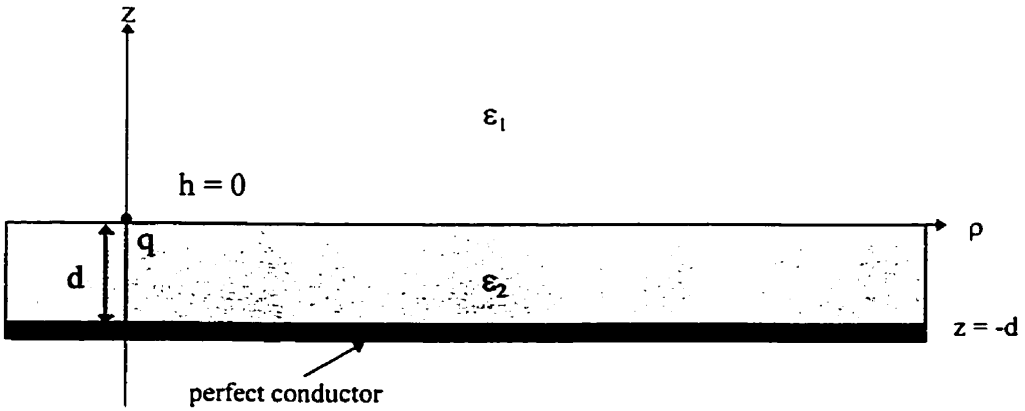


Figure 4-10: Charge q Located At The Interface

For the case of a point charge q located at the dielectric interface, as shown in Figure 4-10, the potential is given by:

For the region z > 0:

$$V(\rho, z) = \frac{q}{4\pi\varepsilon_1}\int_0^\infty \frac{2\frac{\varepsilon_1}{\varepsilon_2}}{\frac{\varepsilon_1}{\varepsilon_2} + \coth(\lambda d)} J_o(\lambda\rho)\, d\lambda \tag{4-31}$$

4.5.2.3 Approximation For Point Charge Above Dielectric Slab

An approximation which removes the integration term in Equation 4-13 is formulated by expanding the tanh terms by a Maclaurin series in Equation 4-25. A Maclaurin series is defined for f(x) as follows[32]:

$$f(x) = f(0) + f'(0)x + \frac{f''(0)x^2}{2!} + \frac{f'''(0)x^3}{3!} + \ldots. \tag{4-32}$$

Expanding the tanh terms in equation 4-24:

$$R \cong -\left[\frac{1 - \frac{\varepsilon_1}{\varepsilon_2}(\lambda d - \frac{2}{3!}(\lambda d)^3 + \ldots)}{1 + \frac{\varepsilon_1}{\varepsilon_2}(\lambda d - \frac{2}{3!}(\lambda d)^3 + \ldots)}\right] \tag{4-33}$$

Further reduction leads to:

$$R \cong -\left[1 - 2\frac{\varepsilon_1}{\varepsilon_2}(\lambda d)\right] \tag{4-34}$$

Considering a Maclaurin series for e^x:

$$e^x = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots. \tag{4-35}$$

For small x:

$$e^{x} \cong 1 + x \tag{4-36}$$

Leads to:

$$R \cong - e^{-2\frac{\varepsilon_1}{\varepsilon_2}(\lambda d)} \tag{4-37}$$

Applying this result to Equation 4-13 leads to:

$$V(\rho, z) \cong \frac{q}{4 \pi \varepsilon_1} \int_0^{\infty} \left[e^{-\lambda |z-h|} - e^{-2\frac{\varepsilon_1}{\varepsilon_2}(\lambda d)} e^{-\lambda (z+h)} \right] J_o(\lambda \rho) \, d\lambda \tag{4-38}$$

Or:

$$V(\rho, z) \cong \frac{q}{4 \pi \varepsilon_1} \int_0^{\infty} \left[e^{-\lambda |z-h|} - e^{-\lambda (z + h + 2\frac{\varepsilon_1}{\varepsilon_2}(\lambda d))} \right] J_o(\lambda \rho) \, d\lambda \tag{4-39}$$

Considering the Weber integral identity given in Equation 4-17, Equation 4-39 reduces to:

$$V(\rho, z) \cong \frac{q}{4 \pi \varepsilon_1} \left(\frac{1}{r_1} + \frac{1}{r_2} \right) \tag{4-40}$$

where:

$$r_1 = \left[\rho^2 + (z - h)^2 \right]^{\frac{1}{2}}$$

$$r_2 = \left[\rho^2 + \left(z + h + 2\frac{\varepsilon_1}{\varepsilon_2} d \right)^2 \right]^{\frac{1}{2}} \tag{4-41}$$

The above approximation is valid when $(\varepsilon_1/\varepsilon_2)d$ is sufficiently small, since for expanding e^x, x was assumed small. This approximation can be utilized to reduce the computation time for a simulation by replacing the time consuming numerical integration for the reflection term.

4.6 Accuracy Of Simulation Models

4.6.1 Discretization Errors

The accuracy of the free space MoM simulation technique described in section 4.4, is investigated by calculating the capacitance of a unit square conducting plate. By varying the number of subsections used to model the plate, the accuracy of the capacitance approximation improves. The capacitance of an object is the sum of capacitances of all its subsections plus the mutual capacitances between every pair of subsections. Again, the charge on each subsection is treated as if it were a point charge. This approximation is 3.8 per cent in error for adjacent subsections, and has less error for nonadjacent ones[26].

Figure 4-11 below, illustrates that as the number of subsections increases, the estimate of the true capacitance improves. A good estimate of the true capacitance for a unit square plate is 40 picofarads[26]. The results indicate that at least 6 Δl per side on the unit square plate are required for an error of 1.02%. Using 6 Δl per side corresponds to 36 patches in total to segment the unit square plate.

Selection of the discretization used for simulations of the CTS were based on choosing a convergent number of segments and making a trade off against simulation time. Further calculations to reduce the error for a given discretization could be made by determining the error capacitance and subtracting it from the calculated values [33].

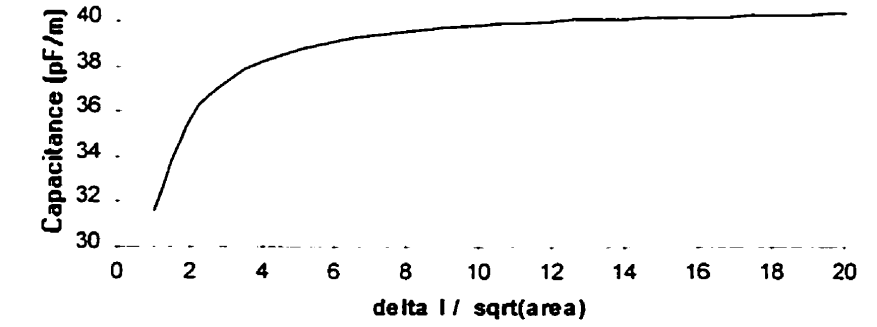


Figure 4-11: Capacitance Of A Unit Square Plate

4.6.1.1 Selection Of Discretization

The selection of discretization of the conductors in the CTS geometry was based on trial comparisons such as the example described below. A certain number of patches were chosen for each element (sensor, track etc.) and the capacitance values calculated, the number of patches was then increased and the capacitance results were compared to the previous results. In this manner, when the results converged, a discretization model was selected.

A geometry (representative of the CTS_128) consisting of three sensors (1.27 mm^2), two ground strips (0.762 mm x 4.527 mm) and one track (25.4 mm x 0.508 mm) was discretized until the results converged. Table 1 summarizes the discretization for

three trials. For all simulations the track was centered under the center sensor and the air gap was 0.254 mm. Figure 4-12 illustrates the convergence of the capacitance values that are required by the equivalent circuit model given in Figure 3-5.

Conductor	Case 1	Case 2	Case 3
all three sensors	Δl = 0.635 mm 8 patches / sensor	Δl = 0.254 mm 25 patches / sensor	Δl = 0.127 mm 100 patches / sensor
both ground strips	Δl = 0.381 mm 24 patches / strip	Δl = 0.254 mm 54 patches / strip	Δl = 0.127 mm 216 patches / strip
track	Δl = 0.508 mm 50 patches	Δl = 0.254 mm 200 patches	Δl = 0.127 mm 800 patches
Total Patches	**110**	**383**	**1532**

Table 1: Discretization Summary

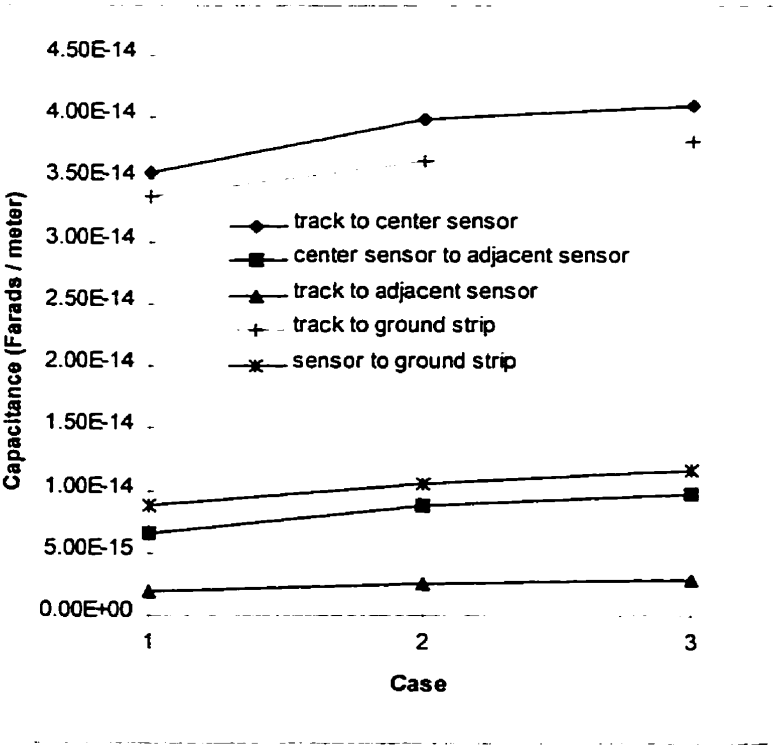


Figure 4-12: Convergence Of Capacitance Calculations

Most of the simulations conducted resulted in around 1000 patches in total, since 0.127 mm^2 patches are generally used to discretize the sensors, ground strips and the tracks.

4.6.2 Accuracy of Approximation For Point Charge Above Dielectric Slab

As stated in section 4.5.2.3, the approximation for a point charge above a dielectric slab is valid when $(\varepsilon_1/\varepsilon_2)d$ is sufficiently small. Here some sample calculations are made to compare the approximation to the solution obtained by integration. By utilizing the approximation computation time is greatly reduced as the matrix elements are filled faster. Only the image term (r_2) is calculated to demonstrate accuracy, which is the reflection term portion of the integral in equation 4-12. For all calculations the dielectric constant for the ε_1 region is 1.0 and for the ε_2 region it is 4.7. Figure 4-13 illustrates the accuracy of the approximation.

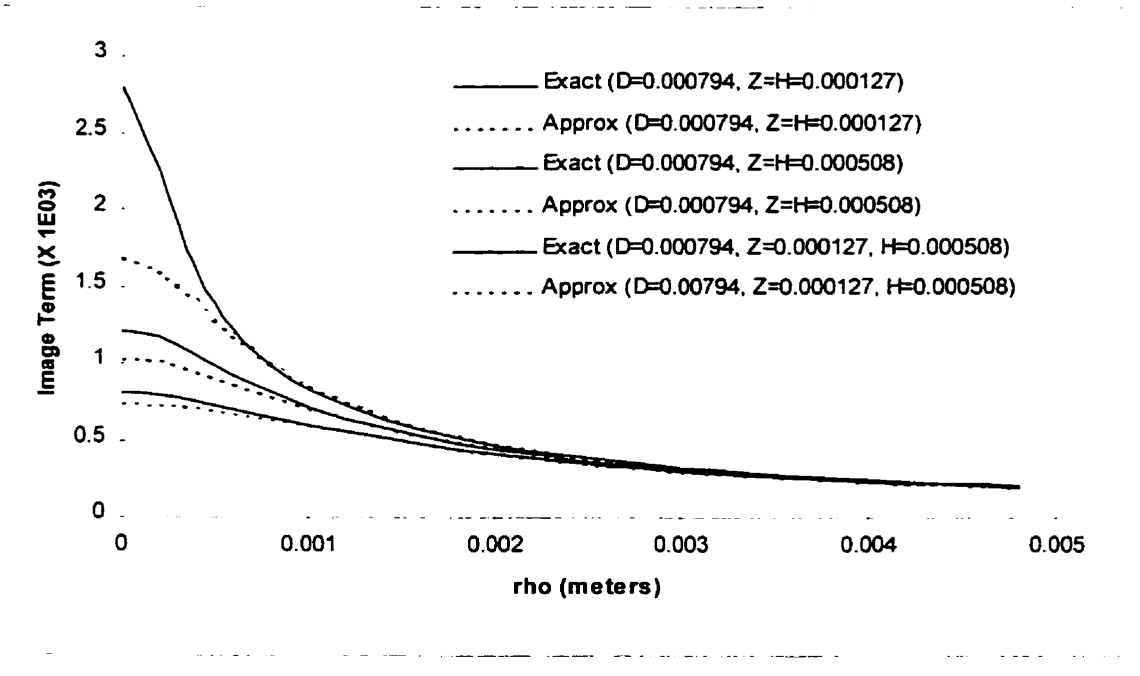


Figure 4-13: Accuracy - Approximation For Point Charge Above Dielectric Slab

Dimensions similar to that of the test BUT discussed earlier where used for the comparison. The BUT thickness, or the dimension D, was 7.94 X 10^{-4} meters. Source and observation point dimensions where also chosen similar to the geometry used in the simulations discussed later. Clearly the data in Figure 4-13 indicates that when the source and observation points are far apart the approximation is sufficient. Simulation computation time can be reduced as a result for similar source and observation point coordinates by using this approximation to replace numerical integration.

Figure 4-13 also indicates that the approximation is more accurate for larger values of source and observation z-axis coordinates. Furthermore, Figure 4-14 demonstrates that the approximation is more accurate when D is small, as expected. The results indicate the that the computation time of simulating the CTS can be reduced by utilizing the approximation for the reflection term with only a small increase in error for cases where the source and observation point are closer together and for thicker BUTs.

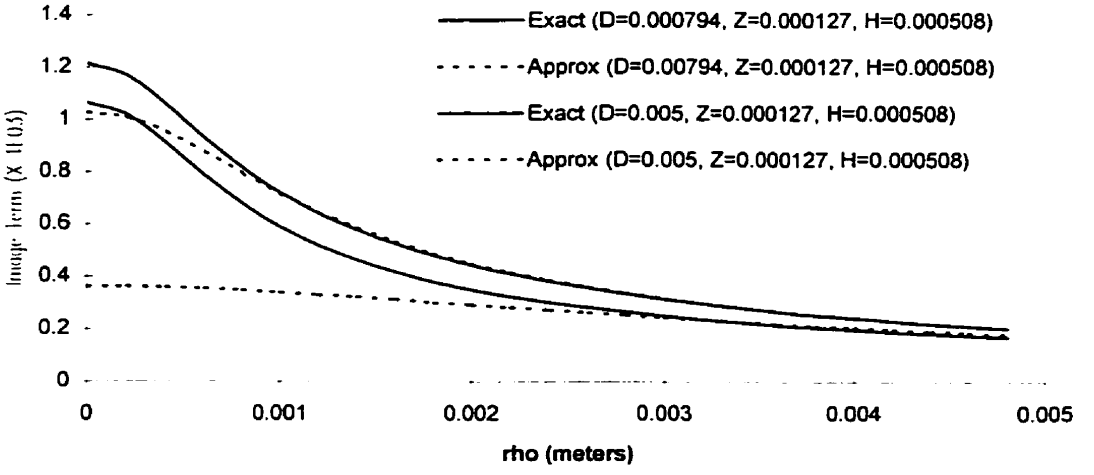


Figure 4-14: Improved Accuracy For Smaller D

4.7 Simulation Data Compared To CTS_128 Data

4.7.1 General Characteristics

For all the general characteristic simulations discussed in this section a program similar to that in Appendix D was used to simulate the CTS geometry, which follows the procedure outlined in section 4.4 for the MoM. The boundary elements (tracks, sensors etc.) were discretized into patches 0.127 mm^2 (5 $thou^2$), which resulted in close to 1000 patches for each simulation. All current magnitudes were calculated using Equation 3-6 without the frequency or voltage included, since all data was normalized. Some of the simulations used in the following sections are from work I conducted while employed at Cirlog Corportion.

4.7.1.1 Simulation Of Track Width Variation

The general characteristics associated with increasing the track width along the entire track length are investigated with the following simulation.

The air gap was set at 0.05 mm and the track length was 3.0 mm. The track widths simulated were 0.1 mm, 0.2 mm and 0.3 mm. The track was centered under the center sensor and scanned from left to right, with the initial position such that the right end of the track was aligned with the right end of the left stimulator (Figure 4-15). The geometry for this simulation was that of the CTS_880, which was the next generation of the CTS_128 with smaller patch sensors (0.6 mm^2). The step size for the scan simulation was 0.01 mm.

The technique outlined in section 4.4 was used to determine the capacitance values between the ground strips, sensors and track. Empirical formulas from section 4.2 were used to determine the capacitance between the stimulators and the track to reduce computation time. The equivalent circuit model shown in Figure 3-5 was used to determine the signal detected by the center sensor.

The result as illustrated in Figure 4-16 and Figure 4-17 below, show the magnitude increases as the track width increases and the characteristic shape of the track remains the same for a width increase along the entire length of the track. Normalization of the three trials shows that the characteristic shape remains generally the same. It should be noted that the track width did not exceed the sensor width of 0.6 mm.

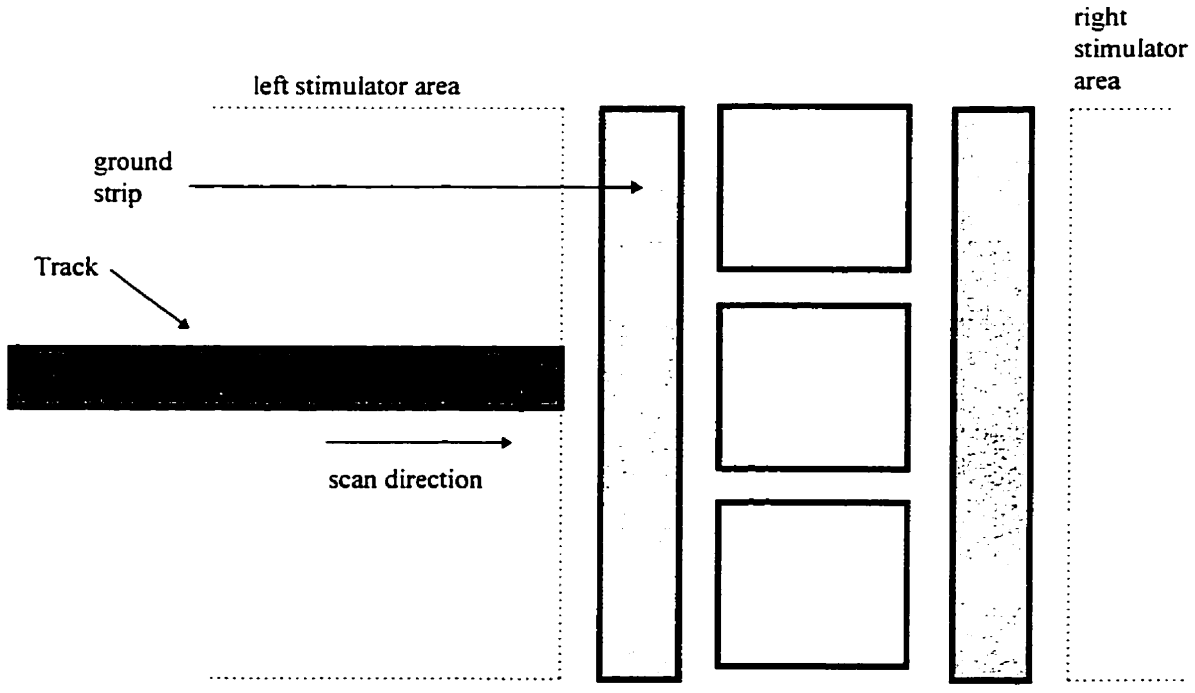


Figure 4-15: Track Width Simulation Initial Position (not to scale)

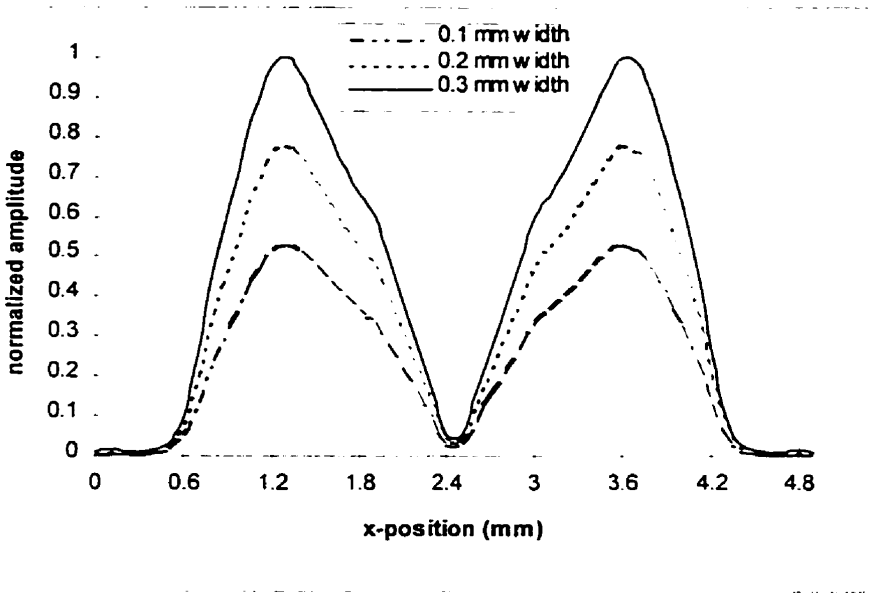


Figure 4-16: Varying Track Widths

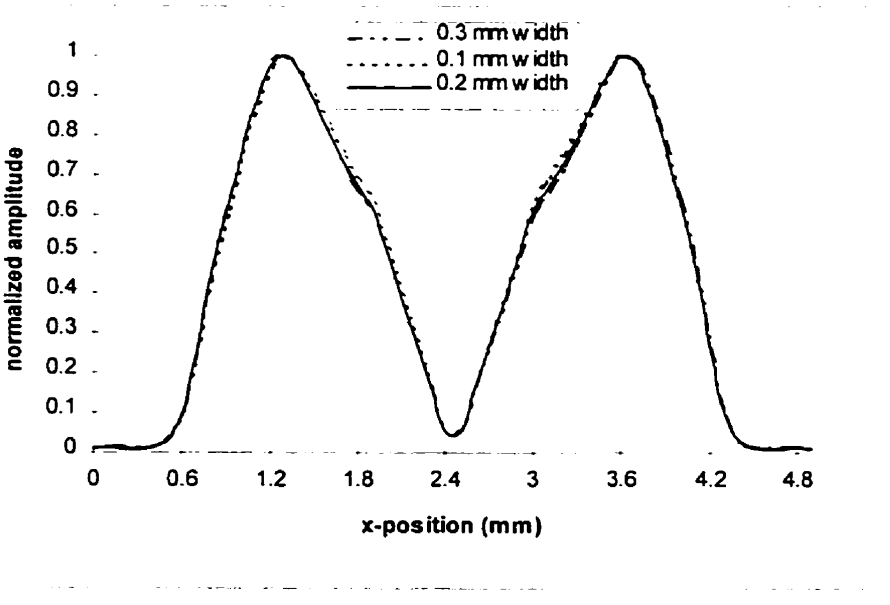


Figure 4-17: Varying Track Widths Normalized

4.7.1.2 Effect of Increasing The Air Gap On Fault Detection

The general characteristics associated with fault detection sensitivity are investigated with the following simulation. The air gap between the SSB and the board under test is varied and the sensitivity to fault detection is shown to degrade as the air gap increases. In other words, the closer the SSB is to the surface of the BUT, the better the fault detection sensitivity becomes.

The end of a track was removed to represent a fault and the air gap was varied from 0.005 inches to 0.02 inches. The track dimensions were 1 inch long by 0.01 inches wide, with the left end of the track reduced by 0.02 inches to simulate the fault. The track was centered under the center sensor and scanned from left to right, with the initial position such that the right end of the track was aligned with the right end of the left ground strip as shown in Figure 4-18. The dimensions are those of the CTS_128. The step size for the scan simulation was 0.025 inches.

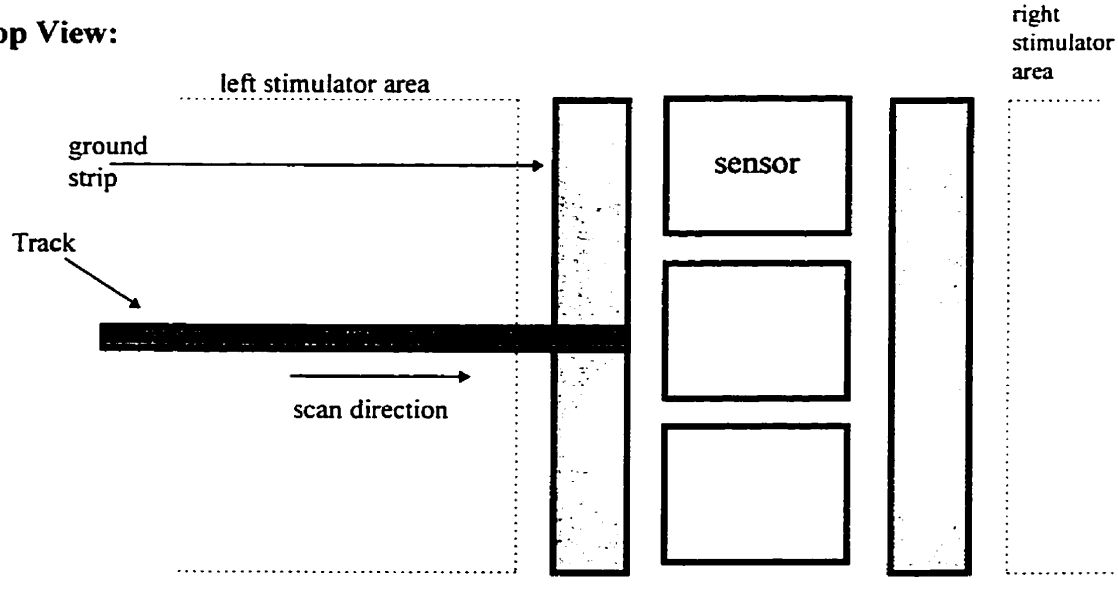


Figure 4-18: Varying Air Gap Simulation Geometry (not to scale)

The technique outlined in section 4.4 was used to determine the capacitance values between the ground strips, sensors and track. Empirical formulas were used to determine the capacitance between the stimulators and the track. The equivalent circuit model shown in Figure 3-5 was used to determine the signal detected by the center sensor. The results are shown below in Figure 4-19 and Figure 4-20. Fault magnitude will decrease as the air gap increases. Furthermore, the characteristic shape of the fault changes as information about the fault is lost as the air gap increases. Sensitivity to fault detection is lost as the air gap increases and the information can not be recovered by increasing the stimulator voltage or frequency. Thus, it is important to maintain the air gap between the BUT and the SSB as small as possible for the best possible fault detection results. These simulation results clearly illustrate the fact that the closer the SSB is to the BUT the better the fault detection sensitivity.

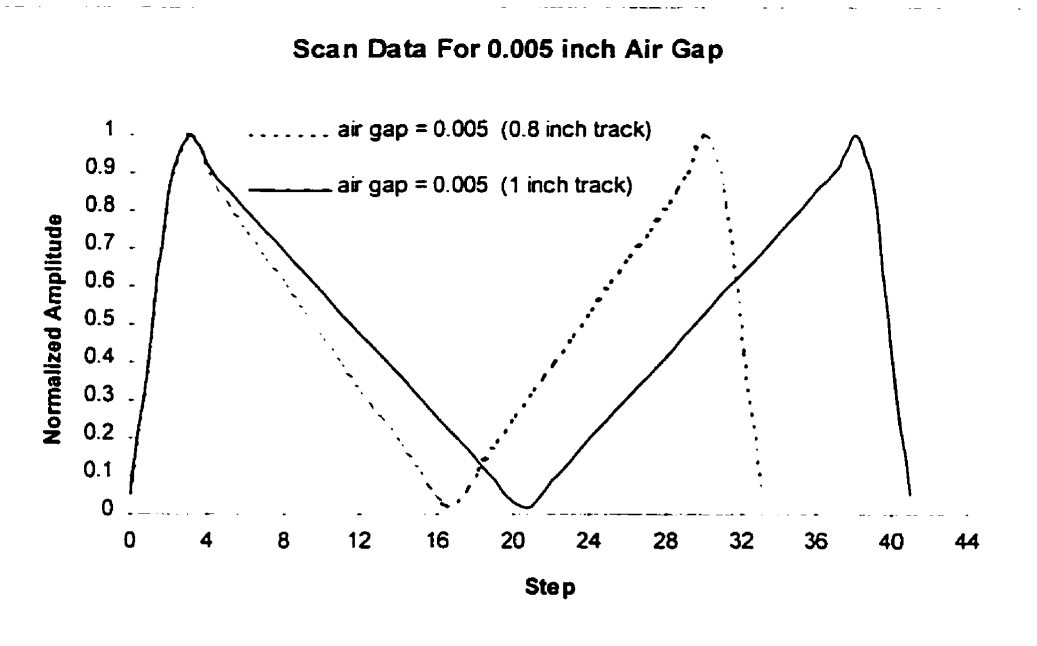


Figure 4-19: Air Gap Scan Data

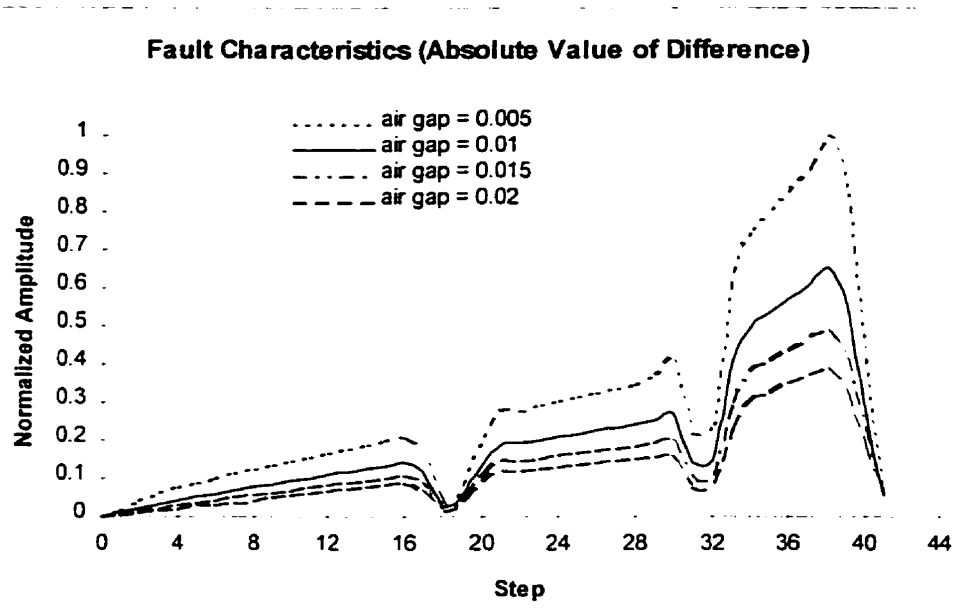


Figure 4-20: Varying Air Gap Fault Data

4.7.1.3 Simulation Of BUT Thickness Variation

The general characteristics associated with varying the BUT thickness are investigated with the following simulation. A single track 3.0 mm long and 0.1 mm wide with a large ground plane on the bottom side of the BUT is simulated with thickness values of 0.0625 mm, 0.125 mm, 0.25 mm and 1.25 mm. The air gap between the SSB and the BUT is fixed at 0.1 mm.

The track was centered under the center sensor and scanned from left to right, with the initial position such that the right end of the track was aligned with the right end of the left stimulator as shown in Figure 4-18 for the increasing air gap simulation. The dimensions used were those of the CTS_880, as in the simulation for track width variation. The step size for the scan simulation was 0.01 mm.

The technique outlined in section 4.4 was used to determine the capacitance values between the ground strips, sensors and track. Empirical formulas were used to determine the capacitance between the stimulators and the track. The equivalent circuit model shown in Figure 3-5 was used to determine the signal detected by the center sensor. The results are shown in Figure 4-21 and Figure 4-22.

In Figure 4-21, the results are normalized with respect to the peak signal detected for the 1.25 mm thick BUT. The results indicate that increasing the thickness of the BUT, or increasing the distance between the track and the large ground plane, results in a larger signal detected by the sensors.

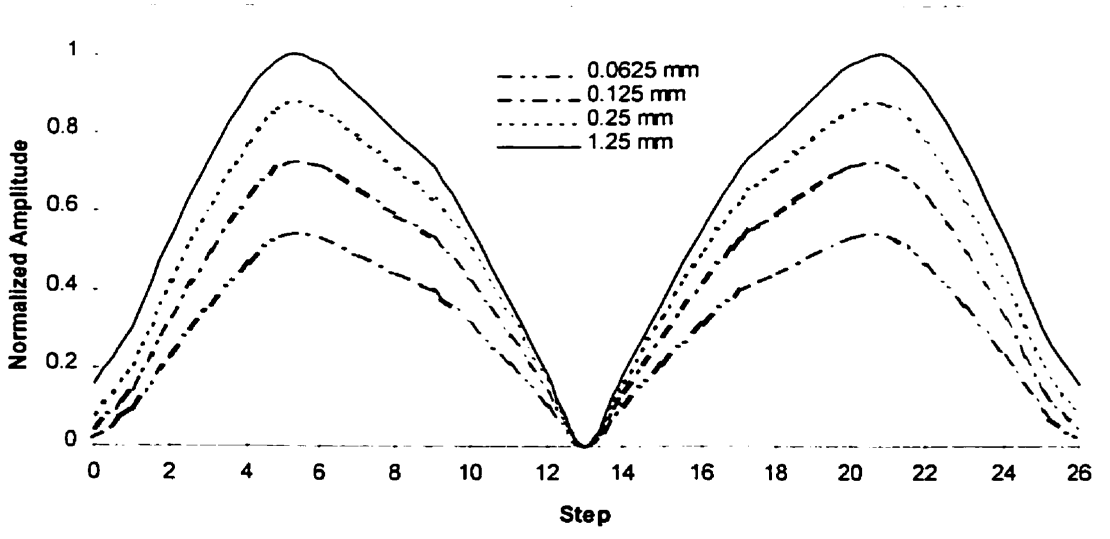


Figure 4-21: Varying BUT Thickness

In Figure 4-22, the results are normalized for each case with respect to the peak signal detected for that case. The results indicate that the characteristic shape of the detected signal remain relatively unchanged when the BUT thickness is varied.

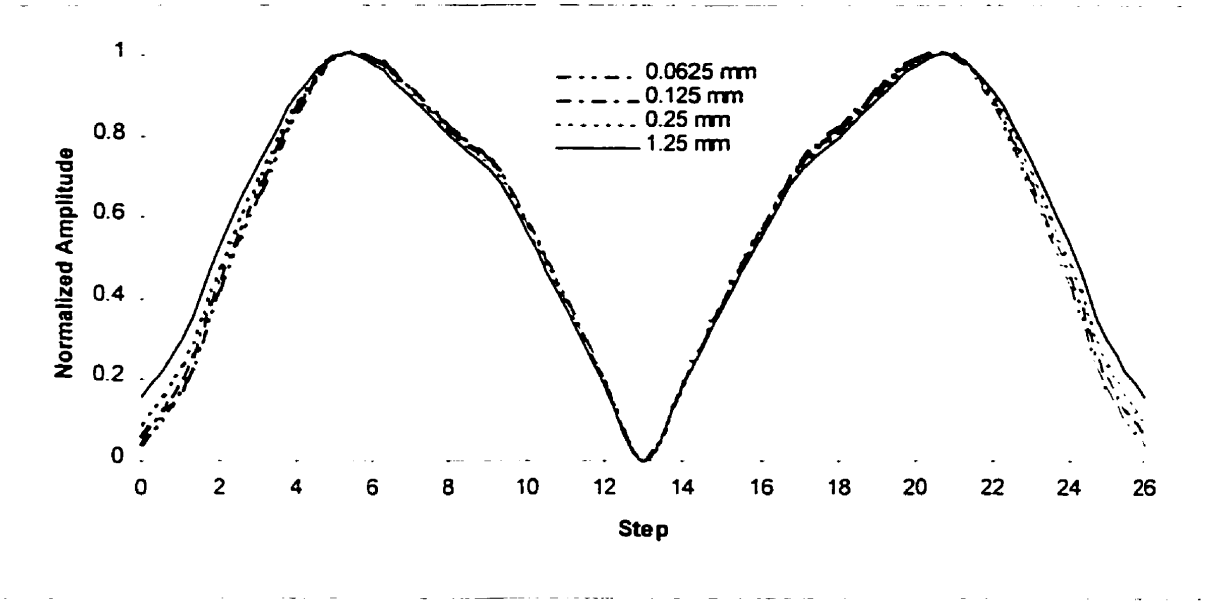


Figure 4-22: Varying BUT Thickness Normalized

4.7.2 Free Space Comparison

Experimental data collected from the CTS_128 is depicted in Figure 4-23 for a single straight track segment on a test BUT with a thickness of approximately 0.03 inches. The test BUT is similar to the one depicted in Figure 3-11. The gap between the BUT and SSB was varied and data collected at approximately 0.005, 0.01, 0.015 and 0.02 inches for four different scans. Track segment dimensions were approximately 1.0 inch by 0.01 inches and the track was approximately centered under sensor number 64, which corresponds to a ground strip width of approximately 0.03 inches. In order to make valid comparisons, the data was normalized to the peak value for the 0.005 inch air gap case. Normalization was utilized since calculation of the exact signal detected by the sensors, which requires considering many other factors as discussed later in this section, was not necessary.

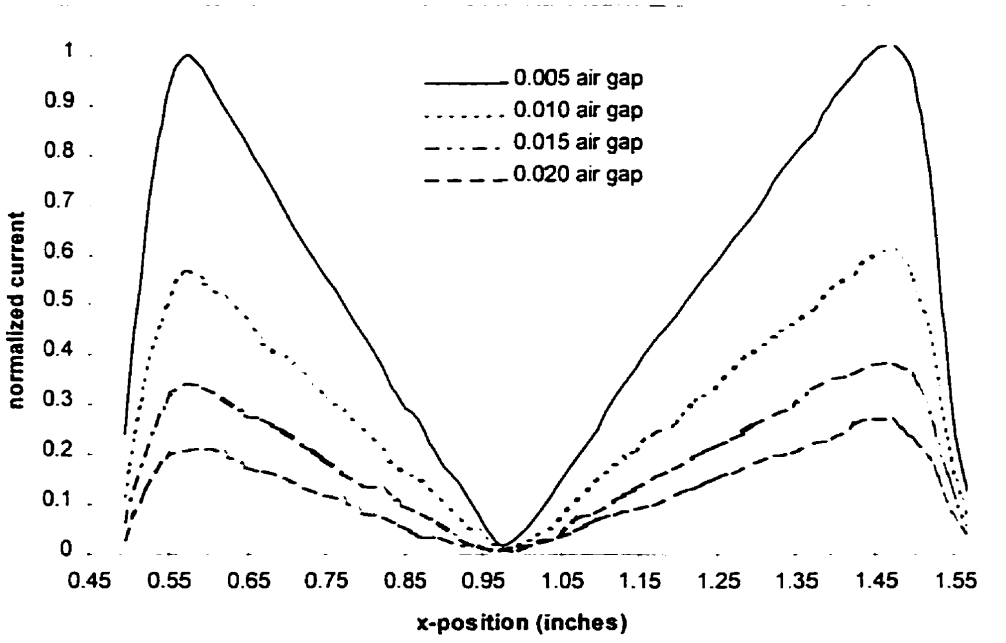


Figure 4-23: CTS_128 Experimental Data

A free space MoM simulation was performed for the same track segment geometry and the results, illustrated in Figure 4-24, were normalized to the peak of the 0.005 inch air gap case. A program similar to the sample in Appendix D was utilized with a geometry of three sensors, two ground strips and the track. Using 0.127 mm^2 patches to discretize the boundary elements resulted in a total of 1108 patches for the simulation. Empirical formulas from section 4.2 were used to calculate the capacitance from the track to the stimulators in order to reduce computation time. The current magnitude was calculated using Equation 3-6 without including the actual voltage and frequency values since normalization was utilized.

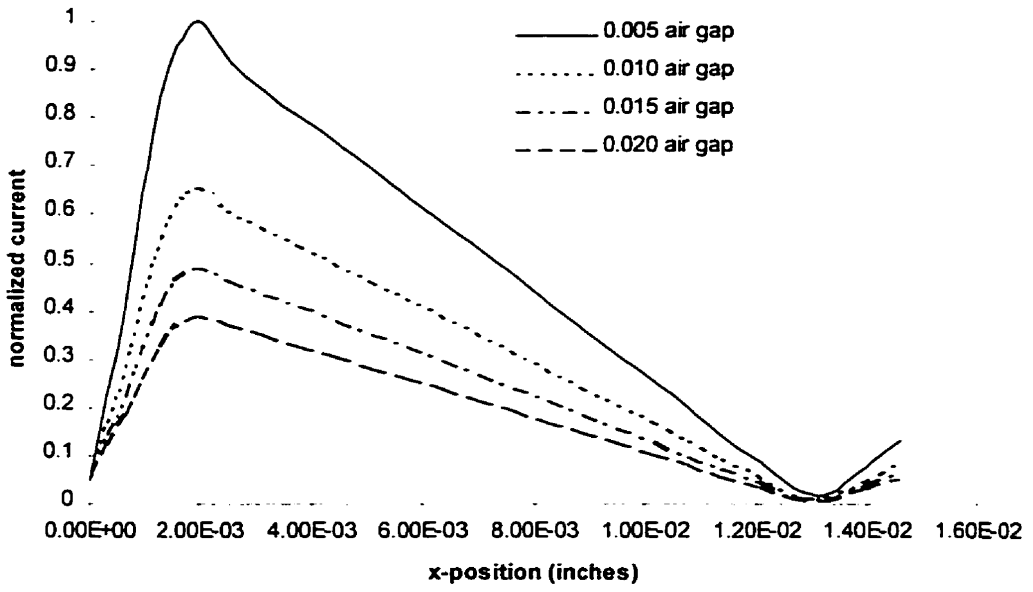


Figure 4-24: Free Space MoM Simulation Data

The following figures illustrate comparisons between the actual CTS_128 experimental data and the MoM free space simulation. Figure 4-25 is for the 0.005 inch air gap, Figure 4-26 is for the 0.010 inch air gap, Figure 4-27 is for the 0.015 inch air gap and Figure 4-28 is for the 0.020 inch air gap.

As a point of reference, the alignment of the data was based on matching up the peak signals for the 0.005 air gap case and all data was normalized with respect to the peak value for the 0.005 air gap case. That is, all the CTS_128 data was normalized with respect to the peak current for the CTS_128 0.005 air gap case and the simulation data was normalized with respect to the peak current for the simulation 0.005 air gap case.

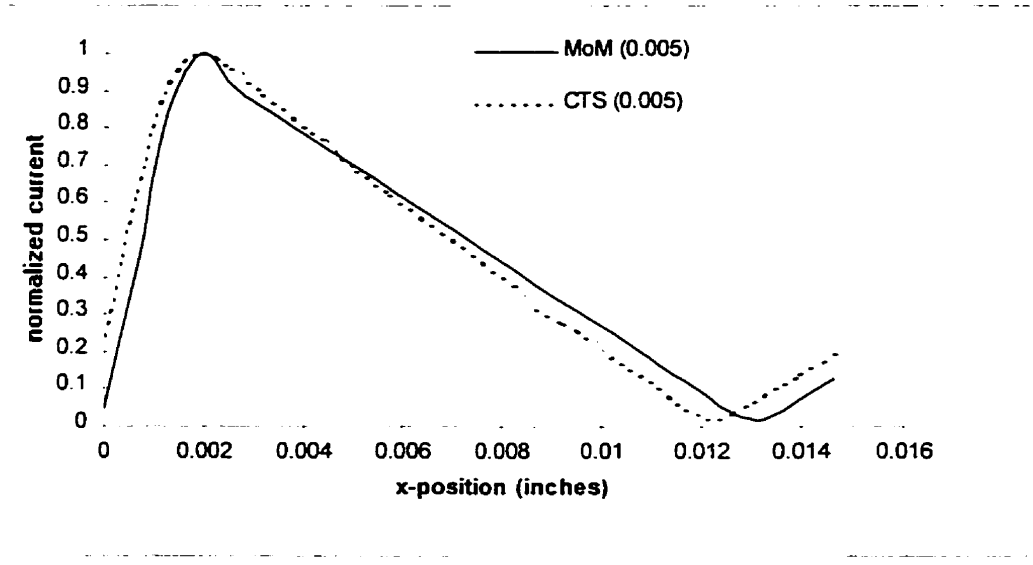


Figure 4-25: Experimental Data / Simulation (0.005 air gap)

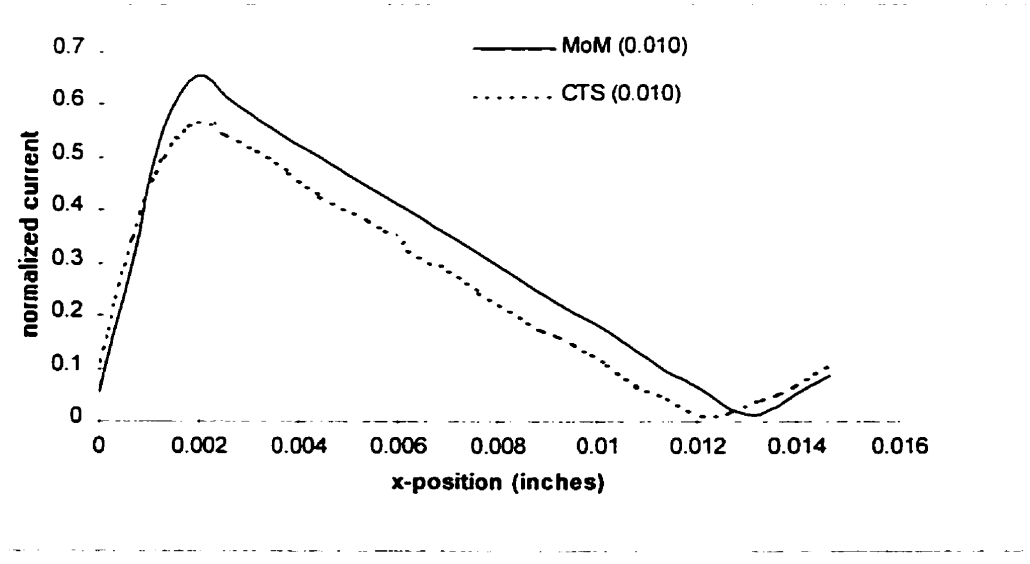


Figure 4-26: Experimental Data / Simulation (0.010 air gap)

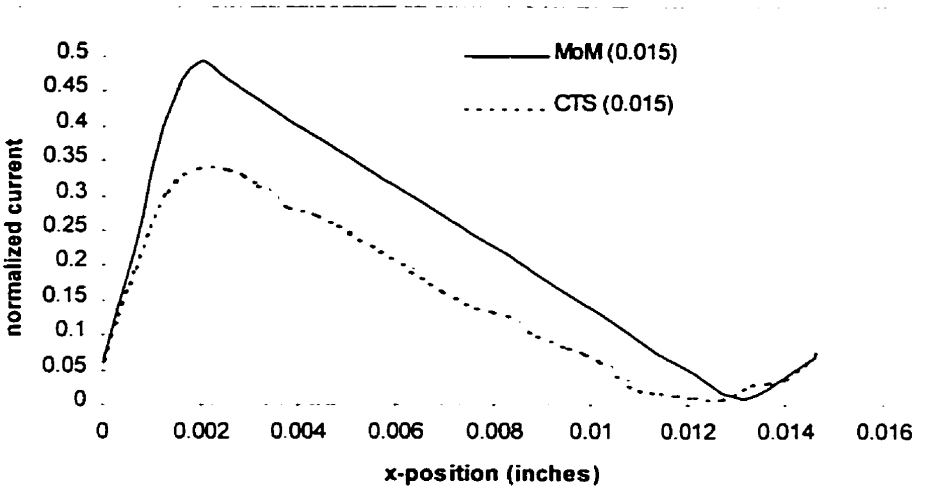


Figure 4-27: Experimental Data / Simulation (0.015 air gap)

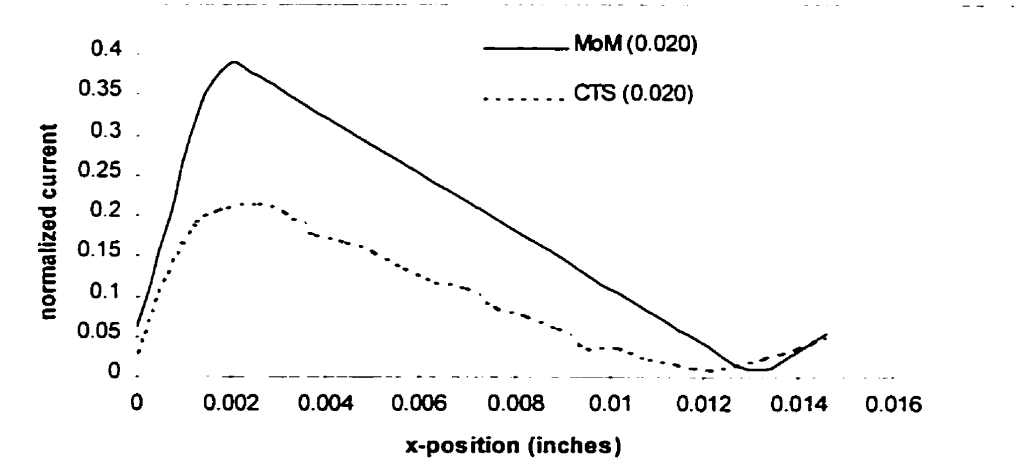


Figure 4-28: Experimental Data / Simulation (0.020 air gap)

Alignment was difficult due to the lack of an exact positioning system on the CTS_128 prototype. As evident in all the figures, the center null does not match up when aligned on the peak value. Furthermore, Figure 4-23 shows that the scan data is not symmetrical as expected for a straight track segment and the sensitivity of detecting the

track is greatly reduced as the air gap is increased and the signal level is degraded by electrical noise.

Never the less the simulation data illustrated above, when considering the changes due to increasing the air gap, show good agreement with the experimental data to within about 10 %. The percentage change in signal level detected by the sensor falls off as the air gap changes as illustrated in Figure 4-29, Figure 4-30 and Figure 4-31. Figure 4-29 depicts experimental data percentage change in signal detected as the air gap changes for two cases, 0.005 to 0.010 inches and 0.010 to 0.015 inches. Figure 4-30 shows the same for the simulation data and Figure 4-31 shows the difference between experimental and simulation results. Notice that the results are closer for the smaller air gap comparison since the sensitivity to signal detection falls off at larger gaps for the CTS_128 as mentioned earlier and the difference is even more evident around the null area since the signal level detected is smallest at the null (theoretically zero).

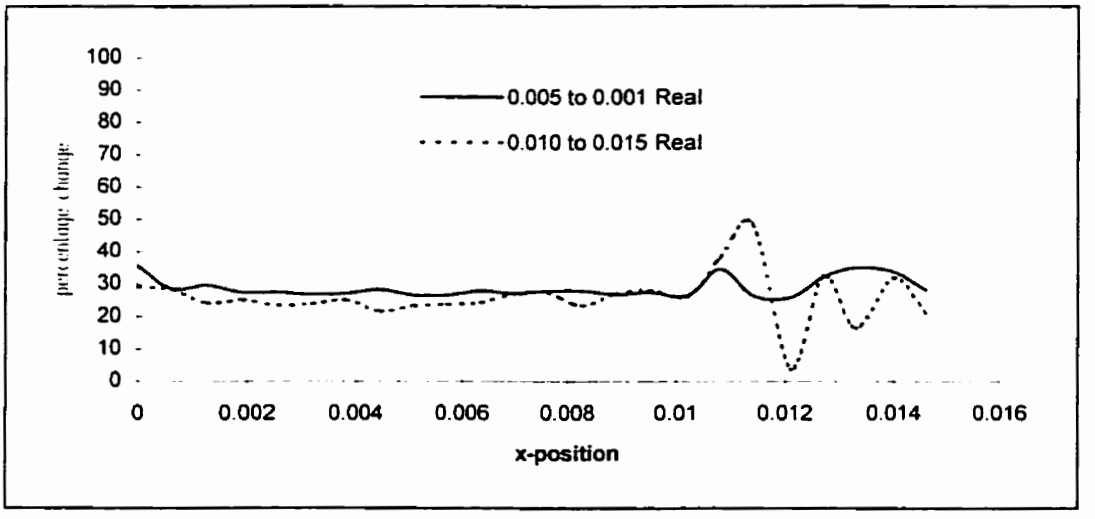


Figure 4-29: CTS_128 Percentage Change As Air Gap Increases

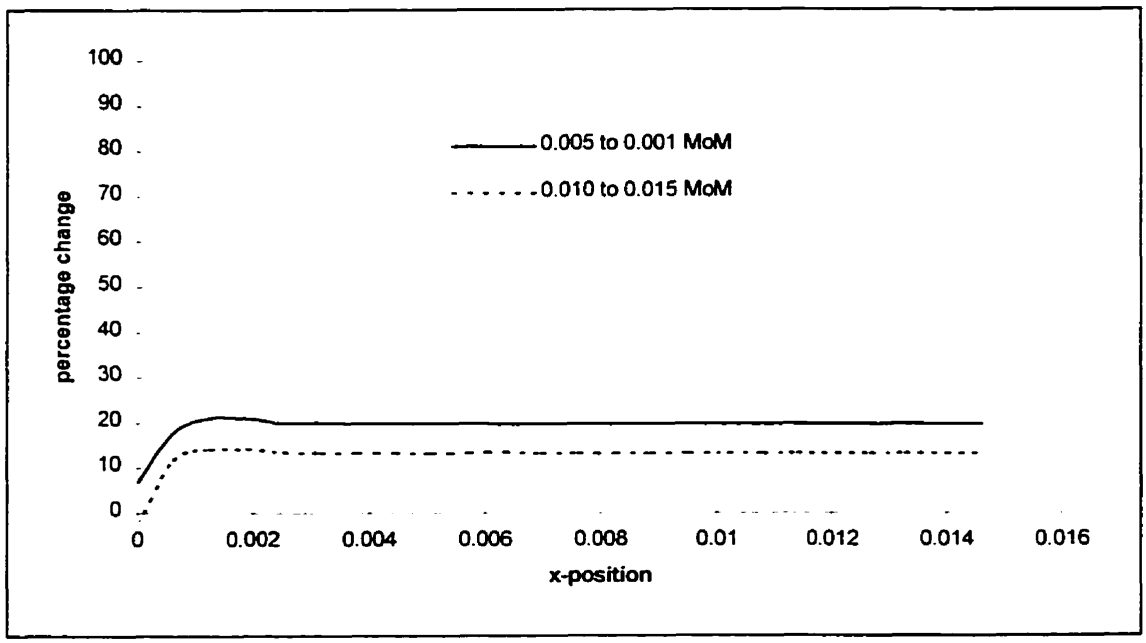


Figure 4-30: Simulation Percentage Change As Air Gap Increases

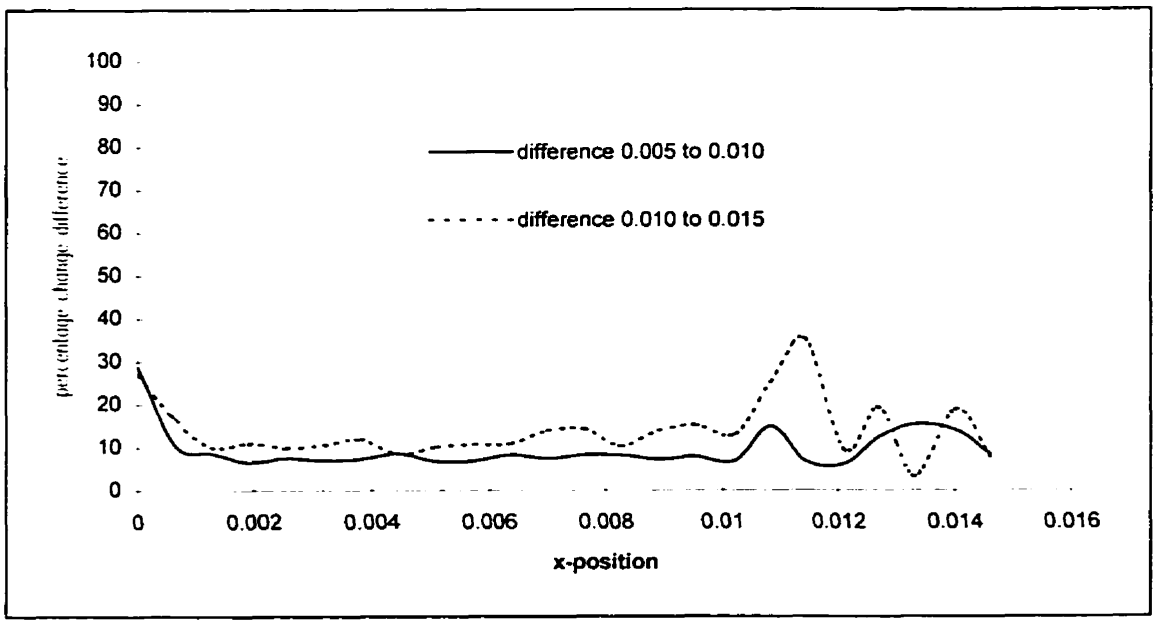


Figure 4-31: Difference Between CTS_128 Data And Simulation Data

4.7.3 Exact Value Calculations

Simulating the CTS_128 to the exact values detected by the SSB is very difficult for many reasons:

- exact air gap between the SSB and the BUT is unknown
- exact starting point for experimental scan is unknown
- various dielectric layers exist
    - SSB coating
    - coating on BUT
    - layer between BUT and ground reference plane
- warpage of SSB, BUT and CTS_128 frame
- thickness variations in SSB and BUT
- SSB power planes (should be considered in simulation)
- signal processing by the SSB electronics
- test BUT track dimension variations

The model required to simulate the CTS experimental results directly would require utilizing the dielectric simulation techniques outlined earlier in section 4.5 and later in section 4.7.3.3.

4.7.3.1 Electrostatic Simulation vs Parallel Plate Simulation

To illustrate the improvement in simulation results gained by solving the electrostatic problem a simple comparison was made. The parallel plate model as defined in section 4.1 is compared to the free space MoM technique and to experimental data. MoM and experimental data from the 0.010 air gap case described in the previous section

was utilized. Figure 4-32 illustrates the comparison for half the scan and Figure 4-33 zooms in on the peak value and area where the track is first detected by the sensor.

Figure 4-33 illustrates that the electrostatic solution is more accurate than the parallel plate model (since it accounts for fringing fields as the track approaches the sensor and the effect of other conductors). In the parallel plate technique the sensor will not detect the track unless it is directly underneath the sensor. Incorporating dielectric layers into the simulation would further improve the simulation results and shape the simulation curves to better agreement with actual data. Incorporating dielectric layers into the simulation is also critical if the exact values are required.

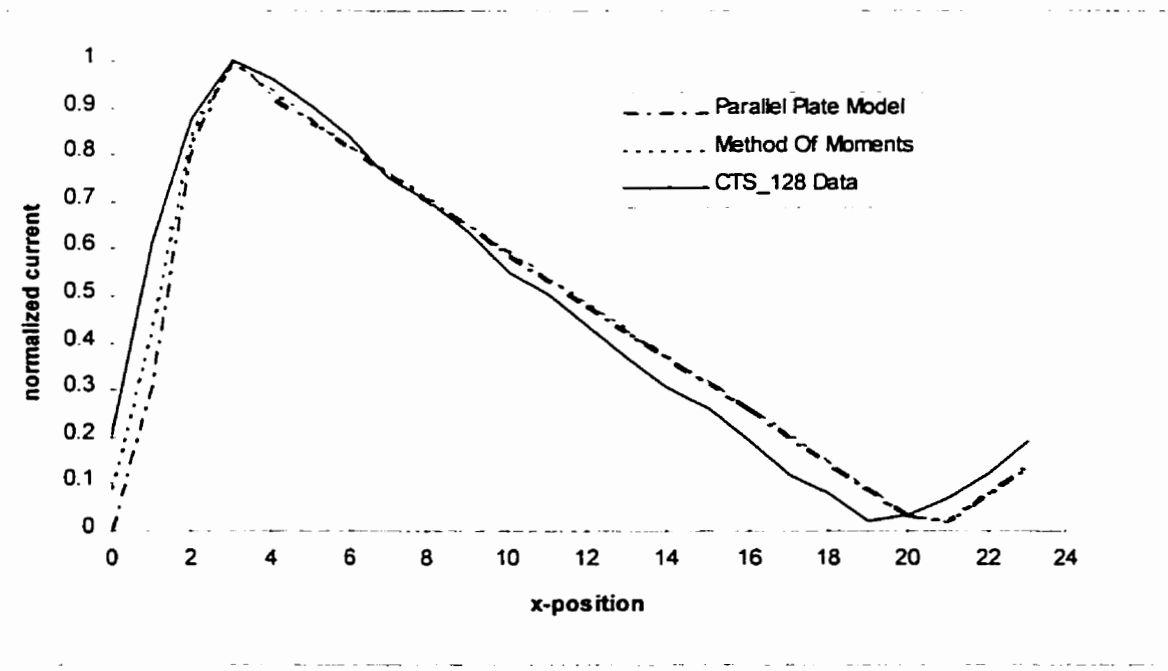


Figure 4-32: Improving Simulation Results Full View

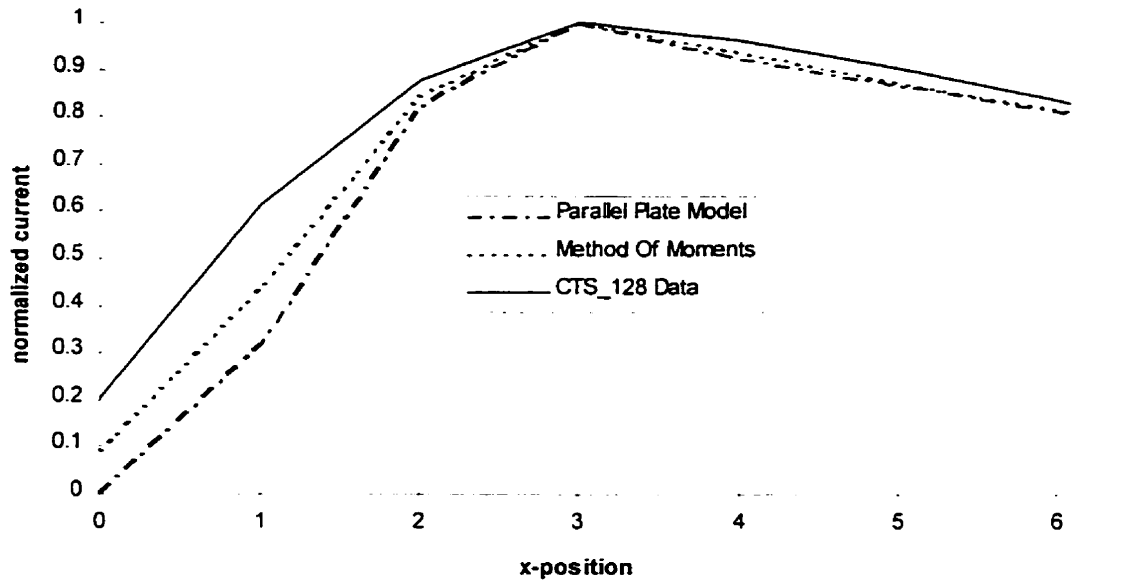


Figure 4-33: Area Sensor First Detects Track

4.7.3.2 Incorporating Effect Of BUT Dielectric Layer Into Simulations

As mentioned in section 4.7.3, the difference between the free space MoM simulation and the experimental data is a result of many factors. One factor that will not change the characteristic shape of the detected signal is the dielectric layer between the BUT and the ground reference plane. It does however reduce the signal detected by the sensors and is thus only important when considering exact value calculations.

A simple simulation was conducted to illustrate the reduced signal pickup. The geometry for the simulation is illustrated in Figure 4-34. The straight track segment was 0.4 inches by 0.01 inches and the BUT thickness was 0.03 inches thick with a dielectric constant of 4.7. The gap between the track and sensor was 0.01 inches. The initial

position of the track was with the right edge of the track lined up with the right edge of the left ground strip.

The geometry was first simulated in free space and then was simulated including the dielectric of BUT. A program written in C code utilized the approximate formula given in Equation 4-39 and Equation 4-30 for the case when the source and observation points were on the dielectric interface as illustrated in Figure 4-10. Both simulations utilized empirical formulas from section 4.2 to calculate the capacitance between the track and stimulators. A sample of the C code is given is Appendix E. The program utilizes Gauss-Legendre Integration [29] for numerical integration.

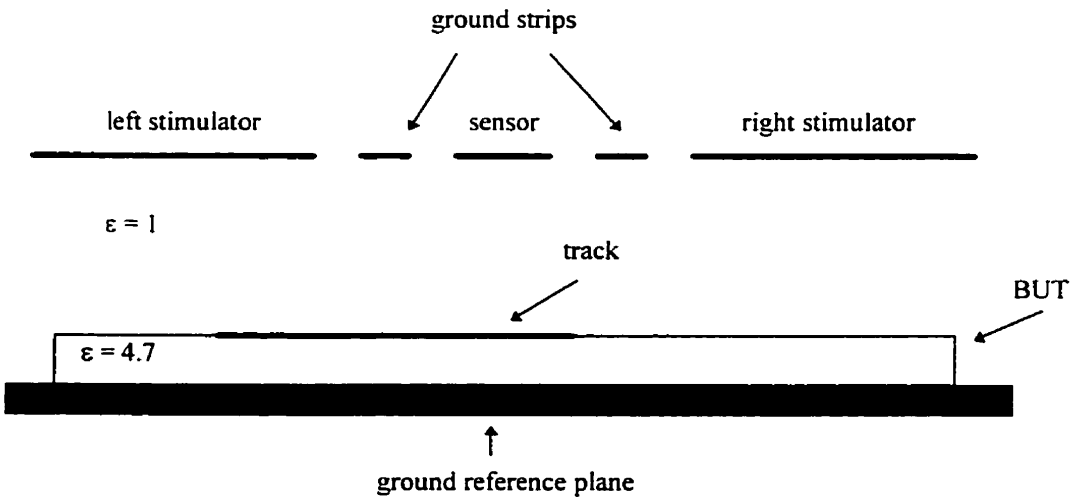


Figure 4-34: BUT Dielectric Simulation Geometry

A partial scan of the track is shown in Figure 4-35. Clearly the effect of the dielectric layer between the track and ground reference plane is a reduction in the signal detected by the sensor. The current depicted in the Figure 4-35 factors in the voltage and frequency of the SSB, but no amplification factors due to the SSB electronics. That is,

Equation 3-6 was used to calculate the current magnitude with a frequency of 200 kHz and a simulator voltage of 10V. The normalization of each case to its peak value illustrated in Figure 4-36 shows that the characteristic shape does not change due to the BUT dielectric layer.

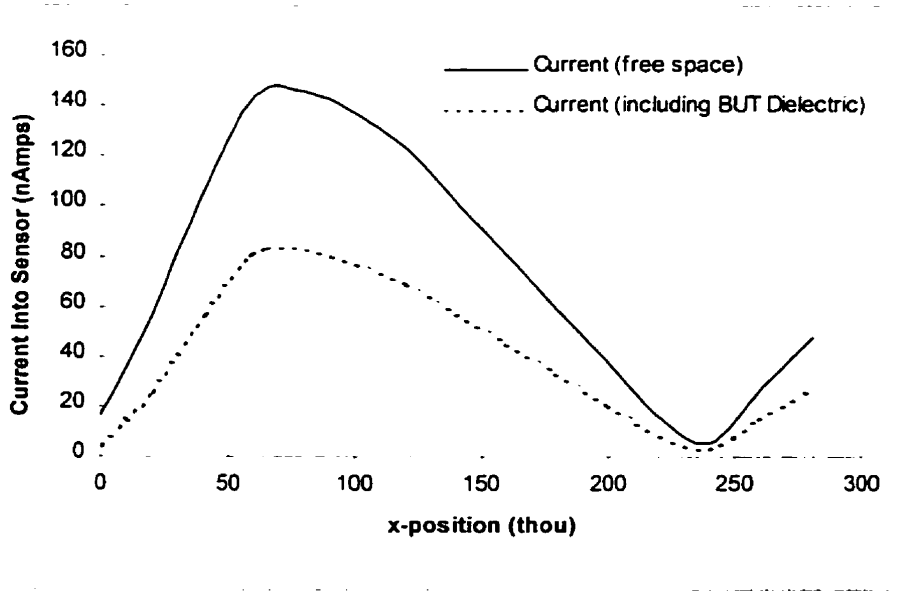


Figure 4-35: Reduced Signal Detection Due To BUT Dielectric Layer

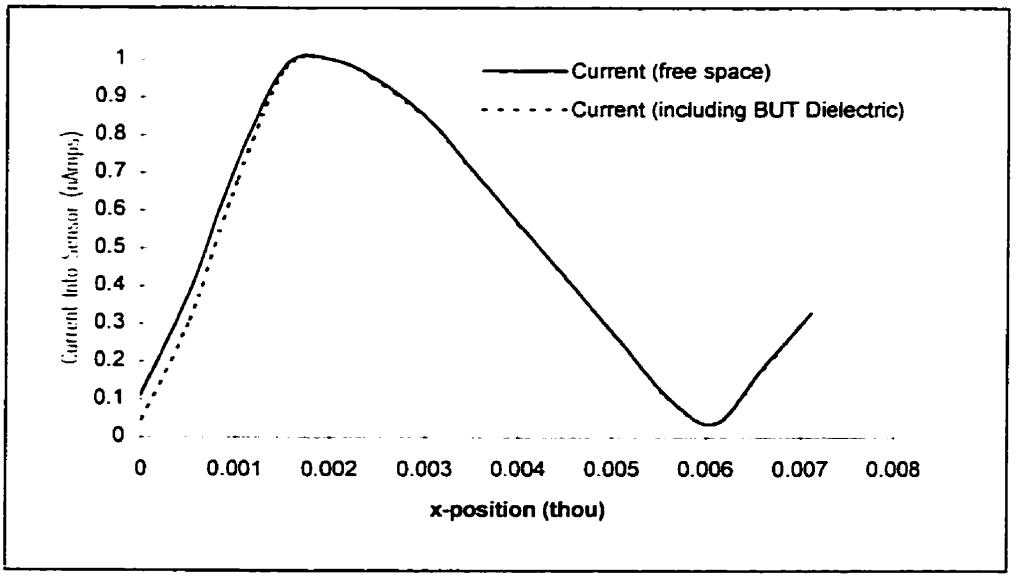


Figure 4-36: Signals Detected Normalized

4.7.3.3 Incorporating Multiple Dielectric Layers

In order to improve simulation results further and study the effects of the dielectric layers in the SSB, a more complex simulation model is required to model the CTS. In the simulation techniques presented earlier, the Green's function was represented as an infinite series obtained from image theory for a two layered dielectric. Using image theory for a three layered dielectric results in a Green's Function represented as a doubly infinite series of images. The image theory technique is impractical when considering more than three dielectric layers since for N dielectric layers, the resulting Green's function consists of an N-1 multiple infinite series [34].

The Complex Images technique [34] results in a Green's function consisting of four terms, including an effective source term plus three complex image terms. This Green's function makes it practical to find the simulated images of point charges in multi-layered media and gives an error of approximately 0.1% when compared to the infinite series Green's function [34]. Furthermore, the Complex Image technique still results in only four terms when extended for more than three dielectric layers.

The Complex Images technique is as follows:

1. Consider the spatial domain potential function that satisfies Poisson's equation, Equation 4-4, and all the boundary conditions.
2. Find the spectral domain potential function by taking the Fourier transform[35] of the spatial domain potential function on x and y. This results in a spectral function of the form:

$$\tilde{\Phi} = \frac{q_0}{2\,\varepsilon_0 \varepsilon_{rs}\,\alpha} \quad F(\alpha, z, z_0) \tag{4-42}$$

The relative dielectric constant of the layer in which the point charge (q_0) is located is ε_{rs}. The spatial coordinates z and z_0 in $F(\alpha, z, z_0)$ are usually fixed in numerical computations to avoid a sophisticated expression for the spatial potential function for arbitrary z and z_0.

3. Find the limit of $\alpha \to \infty$ of the spectral domain potential function F( ) portion:

$$\lim_{\alpha \to \infty} F(\alpha, z, z_0) = F_0 \tag{4-43}$$

4. Subtract the limit from the original F( ) portion and match the remainder with a short sum of decaying exponential functions. The complex coefficients are determined using Prony's method, which is a technique of modeling data as a linear combination of exponentials, and N is the number of exponential terms (2 to 5 generally):

$$F(\alpha, z, z_0) - F_0 \cong \sum_{i=1}^{N} a_i e^{b_i \alpha} \tag{4-44}$$

5. Substituting Equation 4-44 into Equation 4-42 the spectral domain potential function.

6. Take the Inverse Fourier transform and the result is the spatial domain potential in terms of complex images:

$$\Phi \cong \frac{q_0}{4\,\pi\,\varepsilon_0\,\varepsilon_{rs}}\left(F_0\,\frac{1}{r_0} + \sum_{i=1}^{N}\frac{a_i}{r_i}\right)$$

where

$$r_0 = \sqrt{(x-x_0)^2 + (y-y_0)^2}$$

$$r_i = \sqrt{(x-x_0)^2 + (y-y_0)^2 + b_i^2} \tag{4-45}$$

As an example, performing the Complex Images technique for a point charge above a grounded dielectric slab as shown in Figure 4-6 is as follows:

1. The spectral domain potential is given by:

$$\tilde{\Phi} = \frac{q_0}{2\,\varepsilon_0\,\alpha}\left(e^{-\alpha|z-z_0|} + \frac{K - e^{-2\alpha h}}{1 - Ke^{-2\alpha h}}\,e^{-\alpha(z-z_0)}\right)$$

for

$z \geq 0$ and $z_0 \geq 0$ (4-46)

where

$$K = \frac{(1-\varepsilon_r)}{(1+\varepsilon_r)}$$

2. Taking the Limit:

$$\lim_{\alpha \to \infty} \frac{K - e^{-2\alpha h}}{1 - Ke^{-2\alpha h}} = K \qquad (4\text{-}47)$$

3. Subtracting the limit and matching remainder with a short sum of decaying exponential functions:

$$\frac{K - e^{-2\alpha h}}{1 - Ke^{-2\alpha h}} - K \cong \sum_{i=1}^{N} a_i e^{b_i \alpha} \qquad (4\text{-}48)$$

4. Substituting Equation 4-48 into Equation 4-46 and taking the Inverse Fourier transform results in the spatial domain potential function:

$$\Phi \cong \frac{q_0}{4\,\pi\,\varepsilon_0}\left(\frac{1}{r_0} + \frac{K}{r_0} + \sum_{i=1}^{N}\frac{a_i}{r_i}\right)$$

where (4-49)

$$r_i = \sqrt{(x-x_0)^2 + (y-y_0)^2 + (z+z_0-b_i)^2}$$

The Complex Images technique could be utilized to further investigate the CTS technology by extending simulations to include multiple dielectric layers as shown in Figure 4-37, with arbitrary source and observation points.

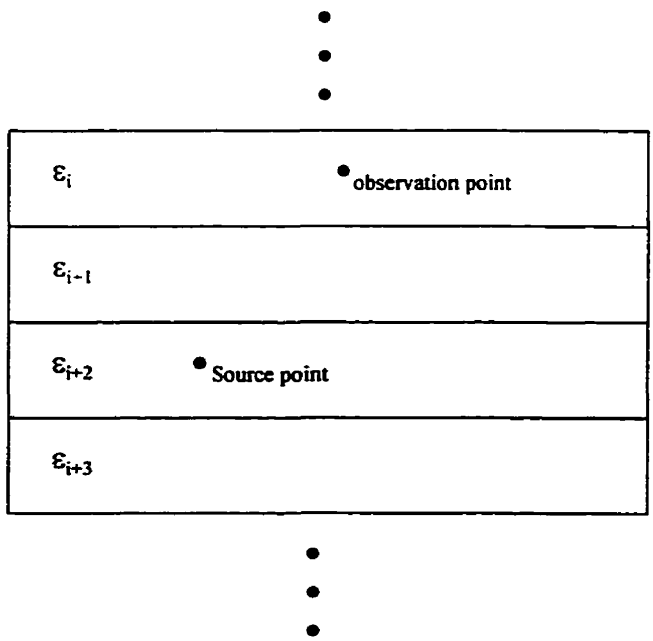


Figure 4-37: Multiple Dielectric Layer Geometry

5. CONCLUSIONS

The utilization of electric field simulations in the research and development of the CTS technology provided useful insight into general characteristics such as track width variations, air gap variations and BUT thickness variations. The free space electrostatic Method of Moments simulations demonstrated that the magnitude of the signal detected by the sensors increased with increased width of the traces on the BUT. Furthermore the simulation demonstrated that the characteristic shape of the scan data remains unchanged for width variations along the entire length of a trace. Sensitivity to fault detection was shown to degrade with an increase in the air gap between the BUT and the SSB. The signal detected was known to degrade with an increased air gap, however the simulation verified that the fault characteristic shape changed and that information about the fault was lost. BUT thickness variation simulations demonstrated that the characteristic shape of the data for a constant air gap was not effected by the proximity of the ground reference plane. However, the magnitude of the signal detected by the sensors decreased as the BUT thickness decreased, or as the ground reference plane comes closer to the SSB.

An important finding from simulations about the CTS_128 prototype was that the utilization of rectification during sensor signal processing degraded sensitivity to fault

detection in certain situations. Future designs should not rectify the signals detected by the sensors during processing in order to improve fault detection sensitivity.

Comparing free space electrostatic MoM simulation data to experimental data from the CTS_128 prototype illustrated that improvements are required for comparing simulation data to experimental data with respect to the characteristic shape. The improvements gained by utilizing the electrostatic MoM simulation technique was demonstrated by comparisons to the characteristic shapes of a parallel plate simulation, MoM simulation and experimental data. Furthermore, the free space MoM comparison to experimental data stressed a requirement for a more accurate positioning system in order to establish correct simulation parameters. The exact location of the beginning of a scan, centering of a track and the air gap between the SSB and the BUT are only approximations with the current CTS_128 prototype.

In order to calculate the exact values detected by the sensors a more complicated model should be considered that incorporates various dielectric layers such as the SSB coating, coating on the BUT and the dielectric layer between the BUT and the ground reference plane. Other factors to consider would include warpage of the SSB, BUT and CTS_128 frame and thickness variations in SSB and BUT.

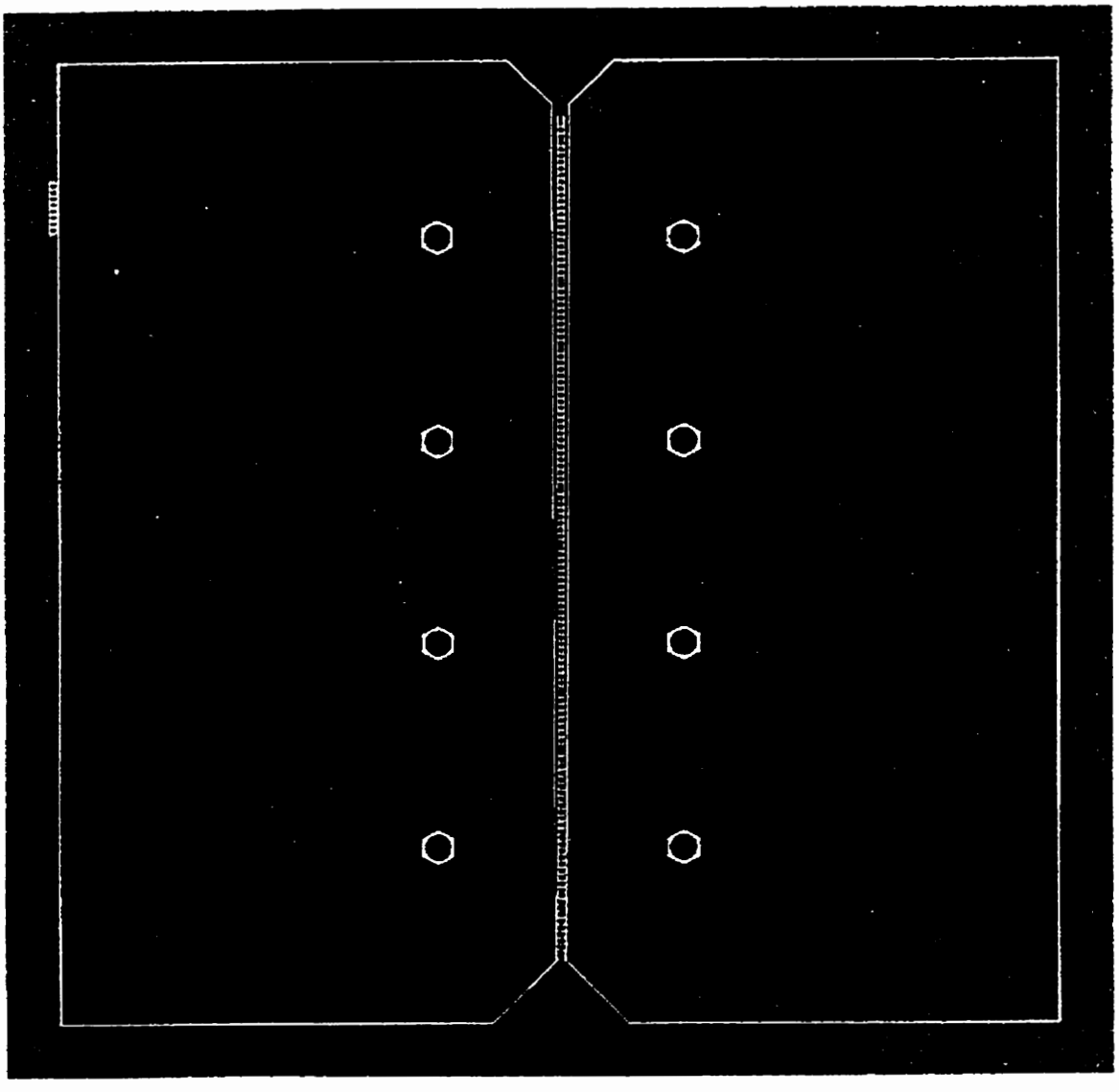
A MoM simulation incorporating the dielectric layer between the SSB and the ground reference plane demonstrated that the dielectric had no effect on the characteristic shape of the signal detected by the sensors, but that the magnitude of the signal detected degrades.

A multi-layer dielectric simulation technique such as the Complex Images technique should be utilized to study the dielectric effects internal to the SSB.

The CTS technology developed by Cirlog Corporation has many advantages over existing bare PCB test systems resulting from the contactless nature of the technology. Simulation techniques such as the MoM integral equation approach will play an important role in the continuing research and development of the technology. The problems associated with testing loaded PCBs or Multi-Chip Modules (MCMs) will be difficult to overcome for the technology in it current form due to issues such as sensitivity to variations in component heights and placement, positioning and sensor resolution. Simulations can be utilized to investigate different sensor / stimulator designs for optimization prior to building the prototype for applications such as loaded PCB testing or MCM testing. Furthermore, simulations could be utilized to generate "Gold Board" data required for manufacturing defect analysis in a future test system that would replace the requirement of a known good board.

APPENDIX A

CTS_128 Sensor Stimulator Board (not to scale) (source: Cirlog)

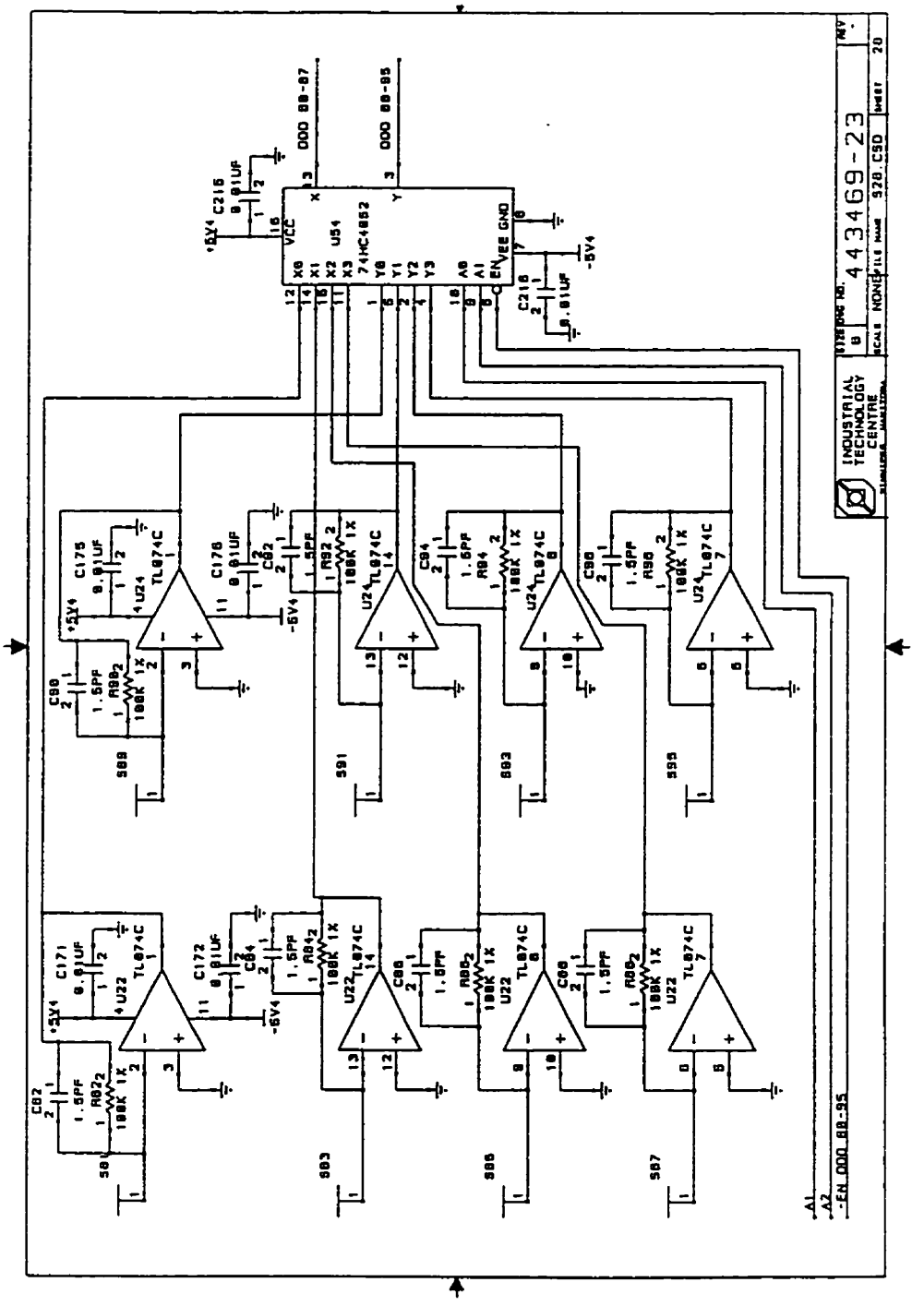


Dimensions:

- SSB approximately 10 inches X 10 inches (25.4 cm X 25.4 cm)
- Sensors 0.05 inches square (1.27 mm^2)
- Center to center spacing 0.06 inches (1.52 mm)
- Ground strip width (at middle of SSB) 0.03 inches (0.76 mm)

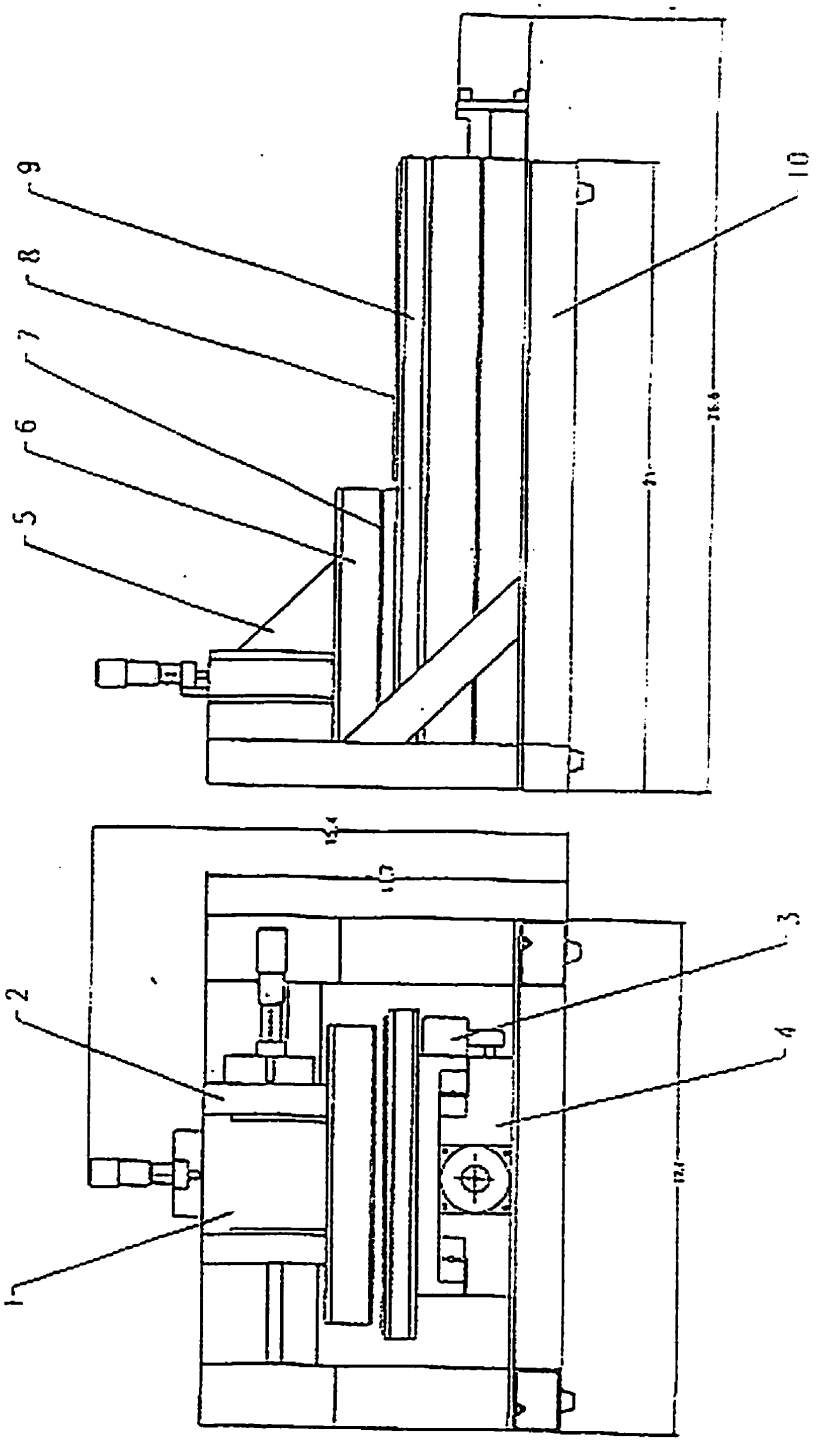
APPENDIX B

Sensor Front End Schematic (source: Cirlog)



APPENDIX C

CTS_128 Mechanical Frame Dimensions (source: Cirlog)



APPENDIX D

Homogeneous Region C Code Sample

```
/* ********************************************************************** */
/* ********************************************************************** */
/*          Simulation Of The CTS_128 PCB Test System          */
/*                    Using a Free Space MoM Technique                    */
/*     Capacitance calculations using the "Method of Moments"    */
/* This program calculates the capacitances between a number of conductors */
/*                                                                        */
/*            Implementation of THE METHOD OF MOMENTS             */
/*                                                                        */
/*   The method of moments is used for solving integral       */
/*   equations, discretization by some method is used to reduce the    */
/*   problem to a set of linear equations.                    */
/*                                                                        */
/*   This program uses square patches for segmentation and treats the  */
/*   charge on each segment as a point charge with uniform charge      */
/*   density over each segment.                               */
/*                                                                        */
/*   It solves the integral equations numerically by writing N        */
/*   equations in N unknowns.  Matrix inversion is required to solve   */
/*   the equations.                                           */
/*                                                                        */
/*     PROCEDURE:                                                 */
/*                                                                        */
/*       (1) Set up the coordinates of all elements             */
                    /* Set up the position of all the elements: sensor,stimulators etc    */
                    /* The xpos, ypos and zpos set up the geometry for the simulation and  */
                    /* the for loops set up the dimensions of the various elements.       */
/*       (2) Calculate the Impedance Matrix (factor out 4*PI*e0)   */
/*       (3) Invert the Impedance Matrix                          */
/*       (4) Set up the voltage and chargedensity matrix          */
/*       (5) Sum up the total chargedensity and convert to total   */
/*           charge by multiplying by the area of each segment     */
/*           *** also multiply by (4*PI*e0) factor left out of     */
/*           the Impedance Matrix.                                */
/*       (6) Capacitance = total charge / voltage                 */

/*            Calculate the Capacitance matrix             */
/*    The capacitance is calculated by the following formula:     */
/*                                                                */
/*              C = summation of all q's / voltage          */
/*                   # of q's = n                          */
/*                  voltage = 1 volt                       */
/*  (specify the voltage at the center of each patch to be 1 volt) */
/*                                                                */
/* This result can be interpereted as stating that the capacitance of */
/* an object is the sum of the capacitances of all its subsections   */
/* plus the mutual capacitances between every pair of subsections.   */
/* ********************************************************************** */

#include <stdio.h>
#include <math.h>
#include <malloc.h>

#define tiny 1.0e-20
#define PI 3.141592654
#define e0 8.854e-12
#define LIM 1200
#define EPS 3.0e-11

/* ********************************************************************** */
/*                  Prototypes                          */
```

```
/* ********************************************************************** */

void  ludcmp(int n,int *indx,float *d);
void  lubksb(int n,int *indx,float *col);
void  nrerror(char error_text[]);
float *vector(int nl,int nh);
void  free_vector(float *v,int nl,int nh);
void  exit(int);


/* ********************************************************************** */
/*              global variable declarations                  */
/* ********************************************************************** */

float *Impedance[LIM],*InvImpedance[LIM];
float xpos[LIM], ypos[LIM], zpos[LIM];
int  n;

/* ********************************************************************** */
/*                    Main Program                        */
/* ********************************************************************** */

void main(void)
{

   float Voltage[LIM],d,col[LIM],CENTER, ChargeDensity[LIM],dx,dx2,dy,dy2,capacitance,rootn;
   float area[LIM], dseg, distance,totalcap, TrackLength, dz, dz2,zsquare, dzimage, dzimage2,atot,qtot;
   float del, TrackHeight, dTrackSeg,z1,BUTthickness,z2,AirGapthickness, ShiftTrackRight,xoffset,yoffset;
   float capmat[10][10],dSensorSeg, avalue, SensorWidth, CenterTrack, ShiftTrackLeft,
   float SensToTrackHeight, dStripSeg,CENTERY, float dStimulatorSeg,TrackWidth ;
   int  indx[LIM],i,j,k,nc, NumSegments[8],nn,ncond, Sensor_X_axis;
   FILE *fptr;

   del = 0.001;          /* convert mm to meters */

   BUTthickness = 0.79375 * del;  /*31.25 thou * 0.0254 = 0.79375 mm BUT */
   AirGapthickness = 0.254 * del; /* 10 thou air gap = 10*0.0254 = 0.254 mm air gap  */
   z1 = BUTthickness;
 /* z2= space between track and sensor/stimulator copper */
   z2 = AirGapthickness;

   ncond=6;              /* The # of elements (sensors,tracks, etc.*/

/* ********************************************************************** */
/* Dynamic memory allocation for Impedance[][] the impedance matrix and  */
/* InvImpedance[][] the inverse impedance matrix.                */
/* ********************************************************************** */

    for(i=1;i<=LIM;i++)
    {
    Impedance[i] = (float *) malloc( LIM*sizeof(float));
    if (Impedance== NULL)
       printf("Insufficient memory available. \n");
    }

    for(i=1;i<=LIM;i++)
    {
    InvImpedance[i] = (float *) malloc( LIM*sizeof(float));
    if (InvImpedance== NULL)
       printf("Insufficient memory available. \n");
    }

/* Track Dimensions */
TrackLength = 25.4 * del;  /* 25.4 mm track length */
TrackWidth = 0.254*del;   /* 0.254 mm wide track */

xoffset= 0.0 * del;     /* scanning the track in x-direction */
```

```
yoffset= 0.0*del;     /* centered under center sensor */

fptr=fopen("file.txt","w");
if(fptr==NULL)
  { printf("can't open file");
  }

fprintf(fptr,"Trackheight, C11,C12,C13,C14,C15,C16,");
fprintf(fptr,"C21,C22,C23,C24,C25,C26,");
fprintf(fptr,"C31,C32,C33,C34,C35,C36,");
fprintf(fptr,"C41,C42,C43,C44,C45,C46,");
fprintf(fptr,"C51,C52,C53,C54,C55,C56,");
fprintf(fptr,"C61,C62,C63,C64,C65,C66,");
fprintf(fptr,"C71,C72,C73,C74,C75,C76,");
fprintf(fptr,"C81,C82,C83,C84,C85,C86 \n");

/* do loop for step scan */
while(xoffset<= ( 14.605 * del))   /* 575 thou distance stepped = 575 * 0.0254 = 14.605mm x-direction */
{
/* ************************************************************************ */
/*                    COORDINATES                                           */
/* (1) center sensor  (2) track  (3) top sensor  (4) bottom sensor  (5) right ground strip  (6) left ground strip   */
/* The track is centered in middle of the center sensor (y-direction)       */
/* Then shifted to where desired!                                           */
/* The xpos, ypos and zpos set up the geometry for the simulation and       */
/* the for loops set up the dimensions of the various elements.             */
/* ************************************************************************ */

   n = 0;        /* start the counter at 0  */

/* ************************************************************************ */
/*            (1) position of Center Sensor                                 */
/* ************************************************************************ */
   dSensorSeg = 0.127 * del; /* Size of segmentation for Sensor. = (5thou = 0.127mm)*/
   SensorWidth = 1.27 * del; /* sensor is 50 thou square = 1.27mm X 1.27mm */
   Sensor_X_axis = (int)((SensorWidth/dSensorSeg)+0.5);  /* 10 */

/* sensor dimensions */
    for (i=1;i<=Sensor_X_axis;i++)  /* 1.27 mm long(x-direction) */
    {
      for (j=1;j<=10;j++)   /* 1.27 mm  wide(y-direction) */
      {
      n += 1;
      xpos[n]=(i-1)*dSensorSeg - 4.5*dSensorSeg;  /* (10-1)/2 = 4.5 */
      ypos[n]=(j-1)*dSensorSeg - 4.5*dSensorSeg; /* (10-1)/2 = 4.5 */
      zpos[n]= z2+z1;
      area[n]=dSensorSeg*dSensorSeg;
      }
    }
   /* The number of segments for the sensor. */
   NumSegments[1]=n;
fprintf(fptr,"Sensor Segments = %6d \n",NumSegments[1]);
printf("Sensor Segments = %6d  \n",NumSegments[1]);

/* ************************************************************************ */
/*                (2) position of Track                                     */
/* ************************************************************************ */
  dTrackSeg = 0.127 * del;   /* Size of segmentation for track. (5thou = 0.127mm) */
/* offset is used to scan the track in the x-direction. */
/* Track Dimensions:                */
/*  1.0 inches long = 25.4mm,  200 segments (x-direction)  */
/*  10 thou wide = 0.254mm wide, 2 segments (y-direction)  */

    for (i=1;i<=(int)((TrackLength/dTrackSeg)+0.5);i++)
    {
      for (j=1;j<=(int)((TrackWidth/dTrackSeg)+0.5);j++)
      {
```

```
        n += 1;
                CENTER = ((TrackLength/dTrackSeg)-1.0)/2.0;
  /* line up right edge of track with right edge of left gnd strip */
 /* shift to the right = 35 thou + tracklength/2 */
 /* in this case 35 thou + 1000/2  = 535 thou shift right */
 /* 535 * 0.0254 = 13.589 mm shift */
        xpos[n]=(i-1)*dTrackSeg - CENTER*dTrackSeg - 13.589*del + xoffset;
        CENTERY = ((TrackWidth/dTrackSeg)-1.0)/2.0;
        ypos[n]=(j-1)*dTrackSeg - CENTERY*dTrackSeg + yoffset;
        zpos[n]= z1;
        area[n]=dTrackSeg*dTrackSeg;
        }
      }
    NumSegments[2]=n-NumSegments[1];/* The number of segments for Track */
fprintf(fptr,"Track Segments = %6d \n",NumSegments[2]);
printf("Track Segments = %6d  \n",NumSegments[2]);

/* ********************************************************************** */
/*          (3) position of Top Sensor(center to center = 1.524mm)     */
/********************************************************************** */
/* sensor dimensions */
     for (i=1;i<=Sensor_X_axis;i++)  /* 1.27 mm long(x-direction) */
     {
       for (j=1;j<=10;j++)   /* 1.27 mm  wide(y-direction) */
       {
       n += 1;
       xpos[n]=(i-1)*dSensorSeg - 4.5*dSensorSeg;
                                                  /*shift up*/
       ypos[n]=(j-1)*dSensorSeg - 4.5*dSensorSeg + 1.524*del;
       zpos[n]= z2+z1;
       area[n]=dSensorSeg*dSensorSeg;
       }
     }
   /* The number of segments for the sensor. */
   NumSegments[3]=n-NumSegments[1]-NumSegments[2];
fprintf(fptr,"Sensor Segments = %6d \n",NumSegments[3]);
printf("Sensor Segments = %6d  \n",NumSegments[3]);

/* ********************************************************************** */
/*              (4) position of Bottom Sensor                  */
/* ********************************************************************** */
/* sensor dimensions */
     for (i=1;i<=Sensor_X_axis;i++)  /* 1.27 mm long(x-direction) */
     {
       for (j=1;j<=10;j++)   /* 1.27 mm  wide(y-direction) */
       {
       n += 1;
       xpos[n]=(i-1)*dSensorSeg - 4.5*dSensorSeg;
                                                  /*shift down*/
       ypos[n]=(j-1)*dSensorSeg - 4.5*dSensorSeg - 1.524*del;
       zpos[n]= z2+z1;
       area[n]=dSensorSeg*dSensorSeg;
       }
     }
   /* The number of segments for the sensor. */
   NumSegments[4]=n-NumSegments[1]-NumSegments[2]-NumSegments[3];
fprintf(fptr,"Sensor Segments = %6d \n",NumSegments[4]);
printf("Sensor Segments = %6d  \n",NumSegments[4]);

/* ********************************************************************** */
/*          (5) position of right ground strip                */
/* ********************************************************************** */
   dStripSeg = 0.127 *del;  /* size of segmentation for strips  same as sensors*/
/* strip dimensions */
     for (i=1;i<=6;i++)  /* 0.762 mm long(x-direction)= 6 segments */
     {
       for (j=1;j<=34;j++)   /* 4.318 mm wide(y-direction)= 34 segments */
```

```
        {
        n += 1;
                                        /* right shift = 25 + 15 +10 = 50 thou (1.27mm) */
        xpos[n]=(i-1)*dStripSeg - 2.5*dStripSeg + 1.27*del;
        ypos[n]=(j-1)*dStripSeg - 16.5*dStripSeg;
        zpos[n]= z2+z1;  /* at same level as sensor/stimulator */
        area[n]=dStripSeg*dStripSeg;
        }
      }
    /* The number of segments for the right strip. */
    NumSegments[5]=n-NumSegments[1]-NumSegments[2]-NumSegments[3]
            -NumSegments[4];
fprintf(fptr,"Ground Strip Segments = %6d \n",NumSegments[5]);
printf("Ground Strip Segments = %6d  \n",NumSegments[5]);

/* ********************************************************************** */
/*          (6) position of left ground strip                             */
/* ********************************************************************** */
/* strip dimensions */
     for (i=1;i<=6;i++)  /* 0.762 mm long(x-direction)= 6 segments */
     {
       for (j=1;j<=34;j++)   /* 4.318 mm wide(y-direction)= 34 segments */
       {
       n += 1;
/* left shift = 25 + 15 +10 = 50 thou (1.27mm) */
       xpos[n]=(i-1)*dStripSeg - 2.5*dStripSeg - 1.27*del;
       ypos[n]=(j-1)*dStripSeg - 16.5*dStripSeg;
       zpos[n]= z2+z1;  /* at same level as sensor/stimulator */
       area[n]=dStripSeg*dStripSeg;
       }
     }
    /* The number of segments for the left strip. */
    NumSegments[6]=n-NumSegments[1]-NumSegments[2]-NumSegments[3]
            -NumSegments[4]- NumSegments[5];
fprintf(fptr,"Ground Strip Segments = %6d \n",NumSegments[6]);
printf("Ground Strip Segments = %6d  \n",NumSegments[6]);
fprintf(fptr,"Total Segments = %6d \n",n);
printf("Total Segments = %6d  \n",n);

/* ********************************************************************** */
/*              Calculate the Impedance Matrix                            */
/* ********************************************************************** */
   for(i=1; i<=n; i++)              /* outter loop */
   {
   Impedance[i][i] = 4.0*PI*0.282*sqrt((float) area[i])
              - area[i]/(2.0*zpos[i]);

     for(j=1; j<=n; j++)            /* inner loop */
     {
       if (j<i)                  /* symmetric matrix */
       {
       dx = xpos[i] - xpos[j];
       dy = ypos[i] - ypos[j];
       dz = zpos[i] - zpos[j];
       dzimage = zpos[i] + zpos[j];
       dx2 = dx*dx;
       dy2 = dy*dy;
       dz2 = dz*dz;
       dzimage2 = dzimage*dzimage;
       distance = (1.0/sqrt((float)(dx2+dy2+dz2)))
              - (1.0/sqrt((float)(dx2+dy2+dzimage2)));
       Impedance[i][j] = area[j]*distance;
       Impedance[j][i] = area[i]*distance;
       }
     }
   }
```

```
/* ********************************************************************** */
/*                MATRIX INVERSION Routine                */
/* ********************************************************************** */
    ludcmp(n,indx,&d);
    for(j=1;j<=n;j++) {
     for(i=1;i<=n;i++) col[i] = 0.0;
       col[j] =1.0;
     lubksb(n,indx,col);
     for(i=1;i<=n;i++) InvImpedance[i][j]=col[i];
     }

for (nc=1;nc<=ncond;nc++)
{
nn=0;

/* Set up the Voltages first. */

    for(i=1;i<=ncond;i++)
    {
      for(j=1;j<=NumSegments[i];j++)
      {
      nn+=1;
      Voltage[nn]=0.0;
      if(i==nc) Voltage[nn]=1.0;
      }
    }
/* Set up the Charges. */
/* Actually the charge density is calculated */
    for(i=1;i<=n;i++)
    {
    ChargeDensity[i]=0.0;
        for(j=1;j<=n;j++)
        {
        ChargeDensity[i]+=InvImpedance[i][j]*Voltage[j];
        }
    }
/* Calculate capacitances. charge = charge density x area  Don't forget the 4*PI*e0 factor */
    nn=0;
    for(i=1;i<=ncond;i++)
    {
    atot=0.0;
    qtot=0.0;
    for(j=1;j<=NumSegments[i];j++)
       {
       nn+=1;
       qtot+= ChargeDensity[nn]*area[nn]*(4.0*PI*e0);
       atot+=area[nn];
       }
    capmat[nc][i]=(fabs(qtot));     /* C=q/V (the voltage = 1 volt) */
    }
}

/* Print out the Capacitances */
fprintf(fptr,"%5.4e,",xoffset);
 for (i=1;i<=ncond;i++)
   {
     for(j=1;j<=ncond;j++)
     {
     fprintf(fptr,"%9.8e,",capmat[i][j]);
     printf("%9.8e,",capmat[i][j]);
     }
   }
/* free up memory */
    for (i=1; i<=LIM;i++)
    {
    free (Impedance[i]);
    free (InvImpedance[i]);
```

```
   }
xoffset += 0.635*del;  /* 25 thou = 25 *0.0254 = 0.635mm step size x-direction  */
} /*end x-scan while*/   /* tracklength = 1000 . + 70 thou edge to edge = 1070 thou or 27.178mm* /
                         /* which means to scan half the track = 535 thou or 13.589mm*/
                        /* 25 thou step size means 21.4 steps or 23 to go just over half*/
printf("FINISHED PROGRAM. \n");
fclose(fptr);
}

/* ********************************************************************** */
/*                    FUNCTIONS                          */
/* ********************************************************************** */

/* ludcmp */
/* LU Decomposition from Numerical Recipes in C */
void ludcmp(int n,int *indx,float *d)
{
   int i,imax,j,k;
   float big,dum,sum,temp;
   float vv[LIM];
   void nrerror(),free_vector();
   *d=1.0;                    /* no row interchange yet */
   for (i=1;i<=n;i++)
   {
      big=0.0;
     for (j=1;j<=n;j++)
         if ((temp=fabs(Impedance[i][j])) > big ) big=temp;
      if (big == 0.0) nrerror("Singular matrix in routine LUDCMP");
       /* no nonzero largest element */
      vv[i]= 1.0/big;          /* Save the scaling */
   }
   for (j=1;j<=n;j++)
          /* This is the loop over columns of Crout's method. */
   {
      for (i=1;i<j;i++)
      {
         sum=Impedance[i][j];
         for (k=1;k<i;k++) sum -= Impedance[i][k]*Impedance[k][j];
         Impedance[i][j]=sum;
      }
      big=0.0;
          /* Initialize for the search for largest pivot element*/
    for (i=j;i<=n;i++)
      {
         sum=Impedance[i][j];
         for (k=1;k<j;k++)
            sum -= Impedance[i][k]*Impedance[k][j];
         Impedance[i][j]=sum;
          if ( (dum=vv[i]*fabs(sum)) >= big)
          {
            big=dum;
            imax=i;
          }
      }
      if (j != imax)           /* Do we need to interchange rows? */
      {
       for (k=1;k<=n;k++)       /* Yes do so..*/
         {
            dum=Impedance[imax][k];
            Impedance[imax][k]=Impedance[j][k];
            Impedance[j][k]=dum;
         }
         *d = -(*d);        /* ..and change the prity of d*/
         vv[imax]=vv[j];    /* Also interchange the scale factor. */
      }

      indx[j]=imax;
```

```
        if (Impedance[j][j] == 0.0) Impedance[j][j]=tiny;
        /* If the pivot element is zero the matrix is singular (at least to the precision */
        /* of the algorithm). For some applications on singular matrices, it is desirable */
        /* to substitute tiny for zero.*/
      if (j != n)      /* now finally, divide by the pivot element.*/
        {
           dum=1.0/(Impedance[j][j]);
         for (i=j+1;i<=n;i++) Impedance[i][j] *= dum;
        }   }}
/* lubksb */
/* Routine for forward and backward substitution for solving */
/* a set of n linear equations. From Numerical Recipes in C  */

void lubksb(int n,int *indx,float col[])
{
    int i,ii=0,ip,j;
    float sum;
         /* when ii is set to a positive n, it will become the index of the first nonvanishing element of col.  We now do the forward
substitution, the only new wrinkle is to unscramble the permutation as we go. */
    for (i=1;i<=n;i++)
    {
        ip=indx[i];
        sum=col[ip];
        col[ip]=col[i];

        if (ii)
            for (j=ii;j<=i-1;j++) sum -= Impedance[i][j]*col[j];
        else if (sum) ii=i;
 /* A nonzero element was encountered, so from now on we will have to do the sums in the loop above */
        col[i]=sum;
    }
    for (i=n;i>=1;i--)   /* now we do the back substitution */
    {
        sum=col[i];
        for (j=i+1;j<=n;j++) sum -= Impedance[i][j]*col[j];
        col[i]=sum/Impedance[i][i];
        /* store a component of the solution vector x. */
    }
}

/* Utility Routines From Numerical Recipes in C  */
/* nrerror */
/* standard error handler */
void nrerror(char error_text[])
{
    void exit();
    fprintf(stderr," numerical Recipes run-time error!...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

/* vector */
/* Allocates a float vector with range [nl..nh]*/
float *vector(int nl,int nh)
{
    float *v;
    v=(float *)malloc((unsigned) (nh-nl+1)*sizeof(float));
    if (!v) nrerror("allocation failure in vector()");
    return v-nl;
}

/* free_vector */
/* Frees a float vector allocated by vector()*/
void free_vector(float *v,int nl,int nh)
{
    free((char*) (v+nl));}
```

APPENDIX E

BUT Dielectric Layer C Code Sample

```
/* ********************************************************************** */
/*          Simulation Of The CTS_128 PCB Test System                     */
/*          BUT Dielectric Layer Incorporated into model                  */
/*           Aproximate Formula used for reflection term                  */
/*              and Full Integration used for z=h=0                       */
/* ********************************************************************** */
#include <stdio.h>
#include <math.h>
#include <malloc.h>

#define tiny 1.0e-20
#define PI 3.141592654
#define e0 8.854e-12
#define LIM 200
#define LIM3 200
#define EPS 3.0e-11

/* ********************************************************************** */
/*                              Prototypes                                */
/* ********************************************************************** */

void ludcmp(int n,int *indx,float *d);
void lubksb(int n,int *indx,float *col);
void nrerror(char error_text[]);
float *vector(int nl,int nh);
void free_vector(float *v,int nl,int nh);
void exit(int);
void gauleg(double x1,double x2,double x[LIM],double w[LIM],int n);
double besslj0(double x);

/* ********************************************************************** */
/*            global variable declarations                                */
/* ********************************************************************** */

float *Impedance[LIM],*InvImpedance[LIM],xpos[LIM], ypos[LIM], zpos[LIM];
int  n;

/* ********************************************************************** */
/*                Main Program                                            */
/* ********************************************************************** */

void main(void)
{
  float Voltage[LIM],d,col[LIM],ebsalon, PrimaryImpedance, ReflectionImpedance;
  float CHARGEDENSITY[LIM],dx,dx2,dy,dy2,capacitance,rootn, float Alamda, Blamda;
  float area[LIM], dseg, distance,totalcap, TrackLength,dz, dz2,zsquare, dzimage, dzimage2,atot,qtot, del;
  float TrackHeight, dTrackSeg,z1,BUTthickness,z2,AirGapthickness, ShiftTrackRight,xoffset,yoffset;
  float capmat[10][10],dSensorSeg, avalue, SensorWidth, CenterTrack, ShiftTrackLeft;
  float  dStripSeg, TrackWidth, CENTERY, CENTER, newvalue, error, integration, TotalIntegration;
  float IntegrationOfSection,e1,e2,H,D,Z,rho, dStimulatorSeg, Fx[LIM3], value, ,r1,r2,approx,Difference;
  float ydist,xmin,xmax,dxfn[LIM3],dxr,rdist,xtest,dxst[LIM3] A, B, R, a, b, dv , Sum ,Value ,sumr ,test
  int numtot,ns,nstep[LIM],nt,nmax,nPointGaussLegendre;
  int indx[LIM],i,j,k,nc, NumSegments[10],nn,ncond, Sensor_X_axis, m, pp, jj, jjj, qq, r, t, p;
  double x[LIM3],w[LIM3],x1,x2;
  FILE  *fptr;

  del = 0.001;    /* convert mm to meters */

/* assign the dielectric constants for the two regions */
  e1 = 1.0*e0;
```

```
   e2 = 4.7*e0;  /* Dielectric constant of BUT is 4.7 */

   BUTthickness = 0.79375 * del; /*  31.25 thou*0.0254=.79375mm BUT , which will equal D later*/
   AirGapthickness = 0.254 * del; /* 10 thou air gap = 10*0.0254 = 0.254 mm air gap  */
   z1 = 0.0 ;                    /* track will be at z=zero interface  */
   z2 = AirGapthickness;          /* z2= space between track and sensor/stimulator copper */
   D = BUTthickness;             /* since z[pos] track = 0 */

   ncond=4;                      /* The # of elements - sensors,tracks, etc.*/

/* memory allocations */
    for(i=1;i<=LIM;i++)
    {
    Impedance[i] = (float *) malloc( LIM*sizeof(float));
    if (Impedance== NULL)
       printf("Insufficient memory available. \n");
    }
    for(i=1;i<=LIM;i++)
    {
    InvImpedance[i] = (float *) malloc( LIM*sizeof(float));
    if (InvImpedance== NULL)
       printf("Insufficient memory available. \n");
    }

/* output file allocation */
fptr=fopen("dielec.txt","w");
if(fptr==NULL)
  { printf(" can't open file");
  }

                /* ***************** */
                /*  Geometry Setup     */
                /* ***************** */
/* 400 THOU (approx 1cm) X 10 THOU */
TrackLength = 10.16 *del;  /* 400 x 0.0254 = 10.16 mm track length */
TrackWidth = 0.254*del;   /* 0.254 mm wide track */

/* Track Scanning variables */
xoffset= 0.0*del;     /* 60  thou (1.524mm)right puts track right edge at sensor right edge */
yoffset= 0.0*del;     /* centered under center sensor */

fprintf(fptr,"Trackheight, C11,C12,C13,C14,");
fprintf(fptr,"C21,C22,C23,C24,");
fprintf(fptr,"C31,C32,C33,C34,");
fprintf(fptr,"C41,C42,C43,C44 \n");

/* do loop for step scan */
while(xoffset<= ( 6.1 *del))   /* 240 thou distance stepped = 240 *0.0254 = 6.096mm x-direction */
{
   n = 0;        /* start the counter at 0  */

/* ********************************************************************** */
/*            (1) position of Center Sensor                */
/* ********************************************************************** */
   dSensorSeg = 0.254 * del; /* Size of segmentation for Sensor. = (10thou = 0.254mm)*/
   SensorWidth = 1.27 * del; /* sensor is 50 thou square = 1.27mm X 1.27mm */
   Sensor_X_axis = (int)((SensorWidth/dSensorSeg)+0.5); /* 10 */
/* sensor dimensions */
    for (i=1;i<=Sensor_X_axis;i++) /* 1.27 mm long(x-direction) */
    {
      for (j=1;j<=5;j++)  /* 1.27 mm  wide(y-direction) */
      {
      n += 1;
      xpos[n]=(i-1)*dSensorSeg - 2.0*dSensorSeg;          /* (5-1)/2 = 2 */
      ypos[n]=(j-1)*dSensorSeg - 2.0*dSensorSeg;          /* (5-1)/2 = 2 */
      zpos[n]= z2 ;                                       /* location of top elements */
      area[n]=dSensorSeg*dSensorSeg;
```

```
      }
    }
NumSegments[1]=n;                    /* The number of segments for the sensor. */
fprintf(fptr,"Sensor Segments = %6d \n",NumSegments[1]);
printf("Sensor Segments = %6d  \n",NumSegments[1]);

/* ********************************************************************** */
/*                (2) position of Track                                   */
/* ********************************************************************** */
  dTrackSeg = 0.254 * del;   /* Size of segmentation for track. (10thou = 0.254mm) */
/* xoffset is used to scan the track in the x-direction. */
    for (i=1;i<=(int)((TrackLength/dTrackSeg)+0.5);i++)
    {
      for (j=1;j<=(int)((TrackWidth/dTrackSeg)+0.5);j++)
      {
      n += 1;
            CENTER = ((TrackLength/dTrackSeg)-1.0)/2.0;
 /* line up right edge of track with right edge of left grnd strip */
/* shift to the right = 35 thou + tracklength/2 */
/* in this case 35 thou + 400/2  = 235 thou shift right */
/* 235 * 0.0254 = 5.969 mm shift */
      xpos[n]=(i-1)*dTrackSeg - CENTER*dTrackSeg - 5.969*del + xoffset;
      CENTERY = ((TrackWidth/dTrackSeg)-1.0)/2.0;
      ypos[n]=(j-1)*dTrackSeg - CENTERY*dTrackSeg + yoffset;
      zpos[n]= z1;                                          /* track is at z1 = 0  */
      area[n]=dTrackSeg*dTrackSeg;
      }
    }
   NumSegments[2]=n-NumSegments[1];                  /* The number of segments for Track */
fprintf(fptr,"Track Segments = %6d \n",NumSegments[2]);
printf("Track Segments = %6d  \n",NumSegments[2]);

/* ********************************************************************** */
/*          (3) position of right ground strip                            */
/* ********************************************************************** */
   dStripSeg = 0.254 * del;  /* size of segmentation for strips  same as sensors*/
/* strip dimensions */
    for (i=1;i<=3;i++)  /*  10 thou long(x-direction)= 3 segment */
    {
      for (j=1;j<=5;j++)   /* 50 thou wide(y-direction)= 5 segments */
      {
      n += 1;
                          /* right shift = 25 + 15 +10 = 50 thou (1.27mm) */
      xpos[n]=(i-1)*dStripSeg - 1.0*dStripSeg + 1.27*del;  /* (3-1)/2 = 1 for xpos shifting  */
      ypos[n]=(j-1)*dStripSeg - 2.0*dStripSeg;
      zpos[n]= z2;  /* at same level as sensor/stimulator */
      area[n]=dStripSeg*dStripSeg;
      }
    }
NumSegments[3]=n-NumSegments[1]-NumSegments[2];
fprintf(fptr,"Ground Strip Segments = %6d \n",NumSegments[3]);
printf("Ground Strip Segments = %6d  \n",NumSegments[3]);

/* ********************************************************************** */
/*          (4) position of left ground strip                             */
/* ********************************************************************** */
/* strip dimensions */
    for (i=1;i<=3;i++)
    {
      for (j=1;j<=5;j++)
      {
      n += 1;
/* left shift = 25 + 15 +10 = 50 thou (1.27mm) */
      xpos[n]=(i-1)*dStripSeg - 1.0*dStripSeg - 1.27*del;
      ypos[n]=(j-1)*dStripSeg - 2.0*dStripSeg;
      zpos[n]= z2;  /* at same level as sensor/stimulator */
      area[n]=dStripSeg*dStripSeg;
```

```
    }
  }
NumSegments[4]=n-NumSegments[1]-NumSegments[2]-NumSegments[3];
fprintf(fptr,"Ground Strip Segments = %6d \n",NumSegments[4]);
printf("Ground Strip Segments = %6d  \n",NumSegments[4]);
fprintf(fptr,"Total Segments = %6d \n",n);
printf("Total Segments = %6d  \n",n);

/* ********************************************************************** */
/*              Calculate the Impedance Matrix                            */
/* ********************************************************************** */

for(i=1; i<=n; i++) /* LOOP#3 */     /* outter loop */
{
/* for the self term  ALL cases including z=h=0 are covered by this approximation!*/
Impedance[i][i] = 4.0*PI*0.282*sqrt((float) area[i])- area[i]/((2.0*zpos[i]) + (2.0*(BUTthickness)));
/* this is for the image term.  ie don't require reflection term for i,i     */

          for(j=1; j<=n; j++)  /* LOOP#2 */              /* inner loop */
          {

                    if (j<i) /* LOOP #1  */             /* symmetric matrix */
                    {
                    /* calculate the primary contribution first */
                    dx = xpos[i] - xpos[j];
                    dy = ypos[i] - ypos[j];
                    dz = zpos[i] - zpos[j];
                    dx2 = dx*dx;
                    dy2 = dy*dy;
                    dz2 = dz*dz;
                 distance = (1.0/sqrt((float)(dx2+dy2+dz2)));
          /*    H =  height to source charge      */
          /*    D =  dielectric thickness         */
          /*    Z =  obsevation z-coordinate      */
          /*    rho =  observation rho-coordinate (OR distance from SOURCE) */
                    H = zpos[i];
                    Z = zpos[j];
          /* rho for the source is always at zero  therefore rho = sqrt (x[j]-x[i])**2 + (y[j]-yi])**2)*/
                    rho=fabs(sqrt(((xpos[j]-xpos[i])*(xpos[j]-xpos[i]))+((ypos[j]-ypos[i])*(ypos[j]-ypos[i]))));
                    if((zpos[i]< 1.0e-50) && (zpos[j]<1.0e-50))
                    {
                    PrimaryImpedance = 0.0;
                    integration=0.0;
                    IntegrationOfSection = 0.0;
                    nPointGaussLegendre = 12;       /* 12 point guass legendre */
                    rdist = rho;                    /* observation coord on x-axis*/
        ydist = (float) fabs(Z) + H;
        xmax = 1.0e+04;                  /* integration upper limit */
          xmin = D/100.0;       /* integration lower limit */
/* Truncate Integrand */
                    if(xmin>(xmax/10.0))   /* xmin must be <= xmax/10 */
                    {
                    xmin = xmax/10.0;
                    }
                    numtot = 1;          /* one segment assumed */
                    dxfn[numtot] = xmax;   /* dxfn[] stores the sementation coords */
                    if(rdist < (20.0/dxfn[numtot]))    /* if rdist < (20.0/xmax) */
                    {                         /* one segment only      */
                    dxr = dxfn[1] / 128.0;
                         for(p=1;p<=numtot;p++)
                         {
                         dxfn[numtot-p+8] = dxfn[numtot-p+1];
                         }
                    dxfn[1]=dxr;
                    dxfn[2]=2.0*dxr;
                    dxfn[3]=4.0*dxr;
                    dxfn[4]=8.0*dxr;
```

```
dxfn[5]=16.0*dxr;
dxfn[6]=32.0*dxr;
dxfn[7]=64.0*dxr;
numtot=numtot + 7;
}
else
{
/*  More Than One Segment Required  rdist > 20/xmax  */
dxr = 20.0/rdist;          /* between xmin and xmax need to segment */
xtest = 0.0;               /* the integrand into more than one seg  */
      while(xtest<xmin)
      {
      xtest += dxr;
      }
      while(xtest<dxfn[numtot])
      {
      numtot+=1;
      dxfn[numtot]=dxfn[numtot-1];
      dxfn[numtot-1]=xtest;
      xtest+=dxr;
            }
dxr = dxfn[1] / 128.0;      /* same code as above for segmenting the */
      for(pp=1;pp<=numtot;pp++)  /* first segment into 8 segments        */
      {
      dxfn[numtot-pp+8] = dxfn[numtot-pp+1];
      }
dxfn[1]=dxr;
dxfn[2]=2.0*dxr;
dxfn[3]=4.0*dxr;
dxfn[4]=8.0*dxr;
dxfn[5]=16.0*dxr;
dxfn[6]=32.0*dxr;
dxfn[7]=64.0*dxr;
numtot=numtot + 7;
}  /* end of if else statement */
/* **********  Sectioning the integrand is completed ********** */
/* set nstep */
for(jj=1;jj<=numtot;jj++)
{
nstep[jj]=1;
}
nstep[1] = 2;  /* the first section is choped into 2 immediately */
/* **********Pass each section one at a time to be Intergrated********** */
dxst[1]=0.0;               /* starting point of integration x1  */
for(jjj=2;jjj<=numtot;jjj++)
{
dxst[jjj]=dxfn[jjj-1];
}
/* Initialize variables to zero */
IntegrationOfSection = 0.0;
TotalIntegration = 0.0;
for(ns=1;ns<=numtot;ns++)   /* 600 one section at a time */
{
a = dxst[ns];
b = dxfn[ns];
nt = nstep[ns];
/* Termination Criteria */
error = 1.0e-05;   /* original value = 1.0e-05 */
nmax = 2;          /* end the sectioning of integrand section after nmax  */
test = 0.0;
            for(k=1;k<=nmax;k++)  /* 620 */
            {
            dv = (b-a)/nt;
            Sum = 0.0;
            x1=a;
            x2=a+dv;
     for(qq=1;qq<=nt;qq++)   /* 650 */
```

```
                                {
                                /* GAUSS LEGENDRE INTEGRATION  set up wi's and xi's */
                                gauleg(x1,x2,x,w,nPointGaussLegendre);
                            /* determine the function value at the xi's store the resulting f(x's) in Fx[] */
                                    for(r=1;r<=nPointGaussLegendre;r++)
                                    {
                                    value = x[r];  /* get the abscissas */
                        Fx[r]=((2.0*(e1/e2))/((e1/e2)+(1/(tanh((float)(value*D))))))*besslj0(value*rho);
                                    } /* for qq to nt loop */
                                    /* Calculate Integration - sum up the weight X f(abscissas)  */
                                Value =0.0;
                                for(t=1;t<=nPointGaussLegendre;t++)
                                {
                                Value += Fx[t]*w[t];
                                }
                                Sum += Value;
                                x1=x2;
                                x2=x2+dv;
                                    }
                                if(k==1)      /* test is evaluated after k>1 */
                                {
                                sumr = Sum;
                                nt=nt*2;
                                }
                                else
                                {
                                test=fabs(Sum-sumr)/fabs(sumr);
                                    if(test<error)
                                    {
                                    break;
                                    }
                                sumr=Sum;
                                nt=nt*2;  /* Increment nt*2 if no BREAK */
                                } /* end of else*/
                        }
            /* If BREAK OCCURED here */
                        IntegrationOfSection = Sum;
                        integration += IntegrationOfSection;
                        } /* end of for(ns =1 to numtot) */
/*Assigning the IMPEDANCE ELEMENTS */
                        ReflectionImpedance = area[j]*integration; /* symmetric matrix */
/* note for the case when z=h=0 the primary was set to zero since programed so that the relection term is the whole impedance term
*/
                        Impedance[i][j] = PrimaryImpedance + ReflectionImpedance;
                        Impedance[j][i] = PrimaryImpedance + ReflectionImpedance;

                        IntegrationOfSection = 0.0;
                        integration = 0.0;
                        sumr = 0.0;
                        Sum = 0.0;
                        }
/* ELSE it is the APPROXIMATE FORMULA  all cases other than z=h=0*/
                        else
                        {
                        PrimaryImpedance = area[j]*distance;
                        /* primary term = r1 = sqrt((rho*rho) + (Z-H)*(Z-H));  done above */
                        r2 = sqrt((rho*rho) + (Z+H+2*(e1/e2)*D)*(Z+H+2*(e1/e2)*D));
                        ReflectionImpedance = area[j]* (-1.0*(1.0/r2)); /* symmetric matrix */
                        /* Assigning the IMPEDANCE ELEMENTS */
                        Impedance[i][j] = PrimaryImpedance + ReflectionImpedance;
                        Impedance[j][i] = PrimaryImpedance + ReflectionImpedance;
                        }
                    } /* LOOP#1 ENDS */
                } /* LOOP#2 ENDS  */
} /* LOOP#3 ENDS  */

/* ********************************************************************** */
```

```
/*              MATRIX INVERSION Routine                */
/* ************************************************************** */

ludcmp(n,indx,&d);
for(j=1;j<=n;j++) {
            for(i=1;i<=n;i++) col[i] = 0.0;
      col[j] =1.0;
            lubksb(n,indx,col);
            for(i=1;i<=n;i++) InvImpedance[i][j]=col[i];
}
for (nc=1;nc<=ncond;nc++)
{
nn=0;
/* Set up the Voltages first. */
            for(i=1;i<=ncond;i++)
            {
                  for(j=1;j<=NumSegments[i];j++)
                  {
            nn+=1;
                  Voltage[nn]=0.0;
            if(i==nc) Voltage[nn]=1.0;
                  }
            }
            /* Set up the Charges. Actually the charge density is calculated */
            for(i=1;i<=n;i++)
            {
            CHARGEDENSITY[i]=0.0;
            for(j=1;j<=n;j++)
                  {
                  CHARGEDENSITY[i]+=InvImpedance[i][j]*Voltage[j];
                  }
            }
/* Calculate capacitances. charge = charge density x area  Don't forget the 4*PI*e1 factor -- not e0 */
nn=0;
            for(i=1;i<=ncond;i++)
            {
            atot=0.0;
            qtot=0.0;
                  for(j=1;j<=NumSegments[i];j++)
                  {
            nn+=1;
                  qtot+= CHARGEDENSITY[nn]*area[nn]*(4.0*PI*e1);  /*NOTE - e0 changes to e1 */
                  atot+=area[nn];
                  }
            capmat[nc][i]=(fabs(qtot));     /* C=q/V (the voltage = 1 volt) */
            }
} /* END of For nc loop */

/* Print out the Capacitances */
fprintf(fptr,"%9.8e,",xoffset);
for (i=1;i<=ncond;i++)
{
            for(j=1;j<=ncond;j++)
      {
      fprintf(fptr,"%9.8e,",capmat[i][j]);
      }
}
xoffset += 0.508*del;  /* 20 thou = 20 *0.0254 = 0.508mm step size x-direction  */
} /*end x-scan while*/   /* tracklength = 400 . + 70 thou edge to edge = 470 thou **/
                        /* which means to scan half the track = 235 thou or 5.969mm*/
                        /* 20 thou step size means 11.75 steps or 12 to go just over half*/
/* free up memory */
for (i=1; i<=LIM;i++)
{
free (Impedance[i]);
free (InvImpedance[i]);
}
```

```
fclose(fptr);
fprintf(fptr,"END OF PROGRAM! \n");
printf("END OF PROGRAM! \n");
}
/* ********************************************************************** */
/*                    FUNCTIONS                                           */
/* ********************************************************************** */
/* ludcmp */
/* LU Decomposition from Numerical Recipes in C */
void ludcmp(int n,int *indx,float *d)
{
   int i,imax,j,k;
   float big,dum,sum,temp;
   float vv[LIM];
   void nrerror(),free_vector();

   *d=1.0;                    /* no row interchange yet */
   for (i=1;i<=n;i++)
   {
      big=0.0;
     for (j=1;j<=n;j++)
         if ((temp=fabs(Impedance[i][j])) > big ) big=temp;
      if (big == 0.0) nrerror("Singular matrix in routine LUDCMP");
       /* no nonzero largest element */
      vv[i]= 1.0/big;          /* Save the scaling */
   }
   for (j=1;j<=n;j++)
          /* This is the loop over columns of Crout's method. */
   {
      for (i=1;i<j;i++)
      {
         sum=Impedance[i][j];
         for (k=1;k<i;k++) sum -= Impedance[i][k]*Impedance[k][j];
         Impedance[i][j]=sum;
      }
      big=0.0;
          /* Initialize for the search for largest pivot element*/
    for (i=j;i<=n;i++)
      {
         sum=Impedance[i][j];
         for (k=1;k<j;k++)
            sum -= Impedance[i][k]*Impedance[k][j];
         Impedance[i][j]=sum;
          if ( (dum=vv[i]*fabs(sum)) >= big)
          {
            big=dum;
            imax=i;
          }
      }
      if (j != imax)           /* Do we need to interchange rows? */
      {
       for (k=1;k<=n;k++)      /* Yes do so..*/
         {
            dum=Impedance[imax][k];
            Impedance[imax][k]=Impedance[j][k];
            Impedance[j][k]=dum;
         }
         *d = -(*d);       /* ..and change the prity of d*/
         vv[imax]=vv[j];    /* Also interchange the scale factor. */
      }
      indx[j]=imax;
      if (Impedance[j][j] == 0.0) Impedance[j][j]=tiny;
/* If the pivot element is zero the matrix is singular (at least to the precision of the algorithm). For some applications on singular
matrices, it is desirable to substitute tiny for zero.*/
      if (j != n)      /* now finally, divide by the pivot element.*/
        {
```

```
            dum=1.0/(Impedance[j][j]);
          for (i=j+1;i<=n;i++) Impedance[i][j] *= dum;
        }
    }
}

/* lubksb */
/* Routine for forward and backward substitution for solving a set of n linear equations */
void lubksb(int n,int *indx,float col[])
{
    int i,ii=0,ip,j;
    float sum;
/* when ii is set to a positive n, it will become the index of the first nonvanishing element of col.  We now do the forward
substitution, the only new wrinkle is to unscramble the permutation as we go. */
    for (i=1;i<=n;i++)
    {
        ip=indx[i];
        sum=col[ip];
        col[ip]=col[i];
        if (ii)
            for (j=ii;j<=i-1;j++) sum -= Impedance[i][j]*col[j];
        else if (sum) ii=i;
                        /* A nonzero element was encountered, so from
                          now on we will have to do the sums in the
                          loop above */
        col[i]=sum;
    }
    for (i=n;i>=1;i--)   /* now we do the back substitution */
    {
        sum=col[i];
        for (j=i+1;j<=n;j++) sum -= Impedance[i][j]*col[j];
        col[i]=sum/Impedance[i][i];
        /* store a component of the solution vector x. */
    }
}

/* Utility Routines From Numerical Recipes in C  */
/* nrerror */
/* standard error handler */
void nrerror(char error_text[])
{
    void exit();
    fprintf(stderr," numerical Recipes run-time error!...\n");
    fprintf(stderr,"%s\n",error_text);
    fprintf(stderr,"...now exiting to system...\n");
    exit(1);
}

/* vector */
/* Allocates a float vector with range [nl..nh]*/
float *vector(int nl,int nh)
{
    float *v;
    v=(float *)malloc((unsigned) (nh-nl+1)*sizeof(float));
    if (!v) nrerror("allocation failure in vector()");
    return v-nl;
}

/* free_vector */
/* Frees a float vector allocated by vector()*/
void free_vector(float *v,int nl,int nh)
{
    free((char*) (v+nl));
}

void gauleg(double x1,double x2,double x[LIM],double w[LIM],int n)
```

```
/* Given the lower and upper limits of integration x1 and x2, and given n  this routine returns arrays x[1...n] and w[1...n] containing
the abscissas and the weights of the Gauss-Legendre n-point quadrature formula*/
{
double z1,z,xm,xl,pp,p3,p2,p1;
int m,j,i;
   m = (n+1)/2;       /* the roots are symmetric in the interval so  */
   xm = 0.5*(x2+x1);   /* we only have to find half of them */
   xl = 0.5*(x2-x1);
   for (i=1;i<=m;i++)  /* loop over the desired roots */
   {
      z=cos(3.141592654*(i - 0.25)/(n + 0.5));
/* starting with above approx to the ith root, we enter the main loop of refinement by Newton's method*/
      do
      {
         p1 = 1.0;
         p2 = 0.0;
         for (j=1;j<=n;j++)  /* loop up the recurrence relation   */
         {                   /* to get the Legendre polynomial    */
         p3 = p2;            /* evaluated at z.                   */
         p2 = p1;
         p1 = ((2.0*j-1.0)*z*p2-(j-1.0)*p3)/j;
         }
/* p1 is now the desired Legendre polynomial. We next compute pp, its derivative, by a standard relation involving also  p2, the
polynomial of one lower order.                  */
         pp = n*(z*p1-p2)/(z*z-1.0);
         z1 = z;
         z = z1-p1/pp;       /* Newton's method */
      }
      while (fabs(z-z1) > EPS);
      x[i] = xm-xl*z;      /* Scale the root to the desired interval */
      x[n+1-i] = xm+xl*z;  /* and put in its symmetric counterpart.  */
      w[i] = 2.0*xl/((1.0-z*z)*pp*pp); /* compute the weight  */
      w[n+1-i] = w[i];              /* and its symmetric counterpart */
   } /* end for loop */
} /* end of gauleg*/

double besslj0(double x)
{
   double ax,z, xx,y,ans,ans1,ans2;
   if ((ax=fabs(x)) < 8.0)
   {
    y=x*x;
ans1=57568490574.0+y*(-13362590354.0+y*(651619640.7+y*(-11214424.18+y*(77392.33017+y*(-184.9052456)))));
ans2=57568490411.0+y*(1029532985.0+y*(9494680.718+y*(59272.64853+y*(267.8532712+y*1.0))));
    ans=ans1/ans2;
   }
   else
   {
    z=8.0/ax;
    y=z*z;
    xx=ax-0.785398164;
ans1=1.0+y*(-0.1098628627e-2+y*(0.2734510407e-4+y*(-0.2073370639e-5+y*0.2093887211e-6)));
ans2 = -0.1562499995e-1+y*(0.1430488765e-3+y*(-0.6911147651e-5+y*(0.7621095161e-6-y*0.934935152e-7)));
    ans=sqrt(0.636619772/ax)*(cos(xx)*ans1-z*sin(xx)*ans2);
   }
/*   printf("returning ans = %9.8f\n",ans);  */
   return (ans);
}
```

REFERENCES

[1] Dominic Haigh, "Using AOI and ICT To Increase Yields", *EE-Evaluation Engineering*, pp. 19-22, February 1998.

[2] Micheal T. Freeman and David Foust, " Flying Prober Simplifies Testing," *Test and Measurement World*, pp. 17-20, November 1997.

[3] Everett/Charles Contact Products, "An Overview of Bare PCB Testing, " *Electronic Manufacturing*, January 1988.

[4] Stan Runyon, "Back To Board Basics", *Electronic Engineering Times*, pp.78, July 18, 1994.

[5] Peter de Bruyn Kops, "Implementing Boundary Scan Testing", *EE-Evaluation Engineering*, May 1997.

[6] Sun Microsystems, "Introduction To JTAG and Boundary Scan", *Microelectronics Whitepaper*, January 1997.

[7] Texas Instruments, "Optimizing Fault Detection For Boundary Scan Testing", *Texas Instruments Information*, 1997.

[8] Brian Laine, "Answers To Common Questions About MDAs", *EE-Evaluation Engineering*, August 1996.

[9] Gerald Jacob, "PCs and Vectorless Test Provide More For Less", *EE-Evaluation Engineering*, June 1997.

[10] Gerald Jacob, "Scaleable Board ATE and Vectorless Test Gain Momentum", *EE-Evaluation Engineering*, February 1996.

[11] Paul O'Shea, "Shrinking Size And Higher Density Require Advanced Inspection Techniques", *EE-Evaluation Engineering*, pp. 24-34, February 1998.

[12] Jon Titus, "X-Ray Systems Reveal Hidden Defects", *Test & Measurement World*, pp. 29-36, February 1998.

[13] Stephen Rooks, "X-Ray Laminography Evaluates Solder Joints", *Test & Measurement World*, February 1996.

[14] Bruce Bolliger, "X-Ray Laminography Improves Functional-Test Yield", *Evaluation Engineering*, November 1996.

[15] Dan Romanchik, "New Technology, Tester for SMT Board Test," *Test and Measurement World*, pp. 85, April 1993.

[16] GenRad Inc., "Junction Xpress A New Wave In Opens Detection", *GenRad Information*, 1997.

[17] Dan Romanchik, "Board Testers Let You Choose The Features You Need", *Test & Measurement World*, pp. 69, December 1993.

[18] Warren L. Stutzman, *Antenna Theory and Design*, Wiley, New York, 1987.

[19] William H. Hayt, *Engineering Electromagnetics*, McGraw Hill, New York, 1989.

[20] David K. Cheng, *Field and Wave Electromagnetics*, Addison Wesley, New York, 1989.

[21] Raymond A. Serway, *Physics For Scientists & Engineers*, Saunders College Publishing, Toronto, 1986.

[22] J. David Irwin, *Basic Engineering Circuit Analysis*, Macmillan, New York, 1990.

[23] K.C.Gupta, R. Garg, and I.J. Bahl, *Microstrip Lines and Slotlines*, Artech House, Dedham, 1979.

[24] K.C.Gupta, R. Garg, and R.Chadha, *Computer-Aided Design Of Microwave Circuits*, Artech House, Dedham, 1981.

[25] Harold A. Wheeler, "Transmission-Line Properties of a Strip Line Between Parallel Planes", *IEEE Transactions on Microwave Theory and Techniques*, vol 26, pp. 866-876, November 1978.

[26] Roger F. Harrington, *Field Computations By Moment Methods*, Macmillan, New York, 1968.

[27] Richard C. Booton, *Computational Methods for Electromagnetics and Microwaves*, Wiley Interscience, New York, 1992.

[28] Eikichi Yamashita, *Analysis Methods For Electromagnetic Wave Problems*, Artech House, Boston, 1991.

[29] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Verterling, *Numerical Recipes in C The Art of Scientific Computing*, Cambridge University Press, Cambridge New York, 1990.

[30] James R. Wait, *Electromagnetic Wave Theory*, Harper & Row, New York, 1985.

[31] T.S. Forzely, "Non-invasive Diagnostics of Microwave Integrated Circuits Using a Scanning Near Field Probe", *M.Sc Thesis*, University of Manitoba, Dept. of Electrical and Computer Engineering, Decemeber 1992.

[32] D.W. Trim, *Engineering Mathematics*, Ruskin, 1989.

[33] Erik H. Lenzing and James C. Rautio, "A Model for Discretization Error in Electromagnetic Analysis of Capacitors", *IEEE Transactions on Microwave Theory and Techniques*, vol 46, pp. 162-165, February 1998.

[34] Y. Leonard Chow, Jian Jun Yang, and Gregory E. Howard, "Complex Images for Electromagnetic Field Computation in Multilayered Media," *IEEE Transactions on Microwave Theory and Techniques*, vol 39, pp. 1120-1125, July 1991.

[35] Dennis G. Zill, *Differential Equations With Boundary Value Problems*, PWS-KENT Publishing, Boston, 1989.

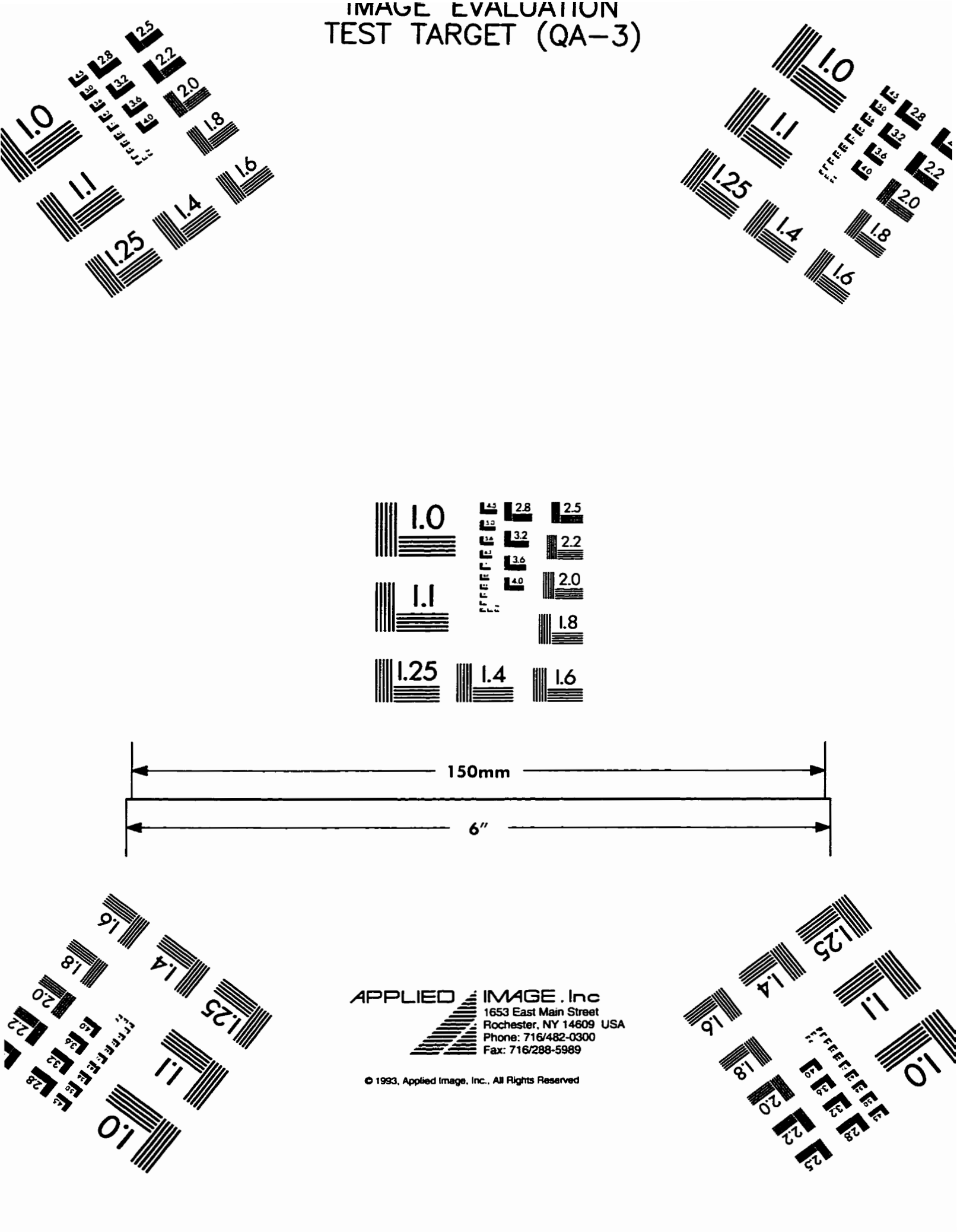


IMAGE EVALUATION
TEST TARGET (QA-3)
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
150mm
6"
APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989
© 1993, Applied Image, Inc., All Rights Reserved