

A CONCEPT VALIDATION OF A MAGNETOMETRY-BASED TECHNOLOGY FOR  
DETECTING CONCEALED WEAPONS IN VEHICLE DOOR PANELS

A Thesis  
presented to  
the Faculty of California Polytechnic State University,  
San Luis Obispo

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science in Mechanical Engineering

by  
Nar Vang  
August 2015

© 2015  
Nar Vang  
ALL RIGHTS RESERVED

## COMMITTEE MEMBERSHIP

TITLE: A Concept Validation of a Magnetometry-Based Technology for  
Detecting Concealed Weapons in Vehicle Door Panels

AUTHOR: Nar Vang

DATE SUBMITTED: August 2015

COMMITTEE CHAIR: Hemanth Porumamilla, Ph.D.  
Associate Professor of Mechanical Engineering

COMMITTEE MEMBER: Frank Owen, Ph.D.  
Professor of Mechanical Engineering

COMMITTEE MEMBER: Xi Wu, Ph.D.  
Associate Professor of Mechanical Engineering

## ABSTRACT

### A Concept Validation of a Magnetometry-Based Technology for Detecting Concealed Weapons in Vehicle Door Panels

Nar Vang

Acts of insurgency have become an increasing threat resulting in extensive measures being taken by the law enforcement authorities to mitigate their devastating effects on human life and infrastructure. This thesis introduces a magnetometry-based information, and signal processing methodology for detecting concealed ferrous objects in vehicle body panels. From extensive literature research, it was observed that while magnetic sensors have been used in a variety of related applications, but they have not been extensively applied to the on-road detection of firearms and explosives concealed in vehicles. This study utilized an extensive experimental protocol for preliminary concept validation. The main idea behind the approach was that almost all concealed weapons and explosives are made up of a considerable amount of ferrous material, and hence produce a local distortion in the Earth's magnetic field. This distortion can then be identified by utilizing sensitive magnetic sensors.

To detect concealed ferrous objects, magnetic signatures of a vehicle door panel were obtained by using a scanning assembly design in this thesis project, and compared to a base magnetic signature of the same vehicle door panel. The base magnetic signature is the magnetic field data of the same vehicle where no foreign ferrous objects were present. To analyze the data, a signal processing methodology was designed. To achieve the objective of accurately detecting concealed ferrous objects, simple measures such as magnetic field strength and its energy density were computed. These simple measures were then used in conjunction with more sophisticated statistical methods such as, normalized cross-correlation and Mahalanobis distance. Although all these methodologies were able to detect a magnetic footprint anomaly in the presence of a concealed object, the Mahalanobis distance approach, in particular provided the most conclusive results in all the test cases considered.

Keywords: signal analysis, national security, magnetic sensors, magnetic anomaly detection,  
Mahalanobi distance, cross correlation, classification

## ACKNOWLEDGMENTS

I would like to acknowledge the support provided by Cal Poly State University through the Extramural Funding Initiative (EFI) program, which helped in the set up of the ground work and the subsequent concept validation of the magnetometry-based technology for concealed weapon detection project.

I would also like to thank Dr. Hemanth Porumamilla for his constant support and mentoring.

In addition, I would like to thank my parents for their continuous support. Thank you for never giving up on me and constantly motivating me to do my best in everything that I pursue.

To my wife, Luc Her, thank you for your patience, constant support, and understanding. Without you, I would not be where I am today.

## TABLE OF CONTENTS

	Page
LIST OF TABLES .....	xi
LIST OF FIGURES .....	xii
CHAPTERS	
1. Background and Motivation .....	1
1.1 Literature Survey .....	1
1.1.1 Detection .....	1
1.1.2 Imaging .....	2
1.1.3 Locating .....	2
1.1.4 Available Weapon Detection Systems .....	3
1.1.4.1 X-Ray Inspection Systems .....	3
1.1.4.2 Millimeter Wave Scanners .....	6
1.1.4.3 Metal Detectors .....	7
1.2 Magnetic Sensors and Its Applications .....	9
1.2.1 Sensor Integration for Low Cost Crash Avoidance Project .....	9
1.3 Magnetic Materials and Its Induced Magnetic Fields .....	10
1.4 The Extramural Funding Initiative Proposal .....	11
1.4.1 Concept Design .....	12
1.5 Final Concept Design .....	14
1.6 Project Overview .....	16
2. Design and Build of Research Prototype System .....	18
2.1 System Design and Build .....	19
2.1.1 Mechanical Design .....	19
2.1.1.1 Door Platform .....	20
2.1.1.2 Scanning Assembly .....	21

2.1.1.2.1 Carriage.....	22
2.1.1.2.2 Scanning Platform.....	22
2.1.1.2.3 Sensor Assembly.....	23
2.1.2 Sensors.....	24
2.1.2.1 Magnetic Sensor.....	24
2.1.2.2 Infrared Sensor.....	24
2.1.3 Electrical System Design and Build .....	25
2.1.4 Computer Workstation.....	26
2.1.5 Data Acquisition System Programming.....	27
3. Testing .....	29
3.1 Test Door .....	29
3.2 Set-Up and Procedures.....	29
4. Analysis .....	31
4.1 Detection of Conceal Ferrous Objects Using Cross-Correlation .....	31
4.2 Detection of Conceal Ferrous Objects Using Mahalanobi Distance.....	32
4.2.1 Mahalanobis Distance and Classification .....	35
4.2.2 Mahalanobis Distance Between the Means of Two Classes.....	36
5. Results.....	40
5.1 Cross-Correlation.....	40
5.2 Magnetic Field Strength and Energy Density .....	41
5.3 Mahalanobis Distance.....	43
5.4 Challenges.....	49
5.4.1 Variation in Scanning Direction .....	49
5.4.2 Variation in Proximity .....	50
6. Other Considerations and Future Work .....	52
6.1 Discussion On Field Implementation.....	52



6.1.1 Vehicles and Magnetic Signatures.....	52
6.1.2 Non-Ferrous Weapons .....	53
6.2 Future Work .....	54
6.2.1 Geomagnetic Field Survey.....	54
6.2.2 Detection of Concealed Weapons Beyond the Door .....	54
6.2.3 Detection Using Fuzzy Logic .....	56
7. Conclusion .....	57
REFERENCES .....	59
APPENDICES	
Appendix A. Sensor Evaluation and Selection .....	62
Magnetic Sensor Evaluation and Selection.....	62
FLC-100 Magnetometer Overview .....	62
HMC-2003 Magnetometer Overview .....	63
Magnetic Sensor Comparison and Selection .....	63
Distance Sensor Evaluation and Selection.....	64
LV-MaxSonar-EZ4 Ultrasonic Sensor Overview.....	64
GP2D120 Infrared Sensor Overview .....	64
Distance Sensor Comparison and Selection .....	64
Appendix B. HMC2003 Set/Reset Circuit.....	65
References.....	67
Appendix C. Computer Build for Computer Workstation .....	68
Appendix D. LabView Program for PCI-NI 6259 .....	69
Channel Configuration.....	69
File Configuration.....	70
Data Read/Write and End Task.....	70
Appendix E. MatLab Program.....	71

MatLab Main Program.....	71
MatLab Functions .....	104

## LIST OF TABLES

Table	Page
Table 1 Results of Functions 1, 2, and 3 .....	45
Table 2 Threshold and Limit Index Results.....	48
Table 3 Magnetic Sensor Comparison .....	63
Table 4 Distance Sensor Comparison .....	64
Table 5 Part Lists for Set/Reset Circuit Assembly .....	66
Table 6 Software Installed .....	68
Table 7 Computer Parts List .....	68

## LIST OF FIGURES

Figure	Page
Figure 1 American Science and Engineering Inc.'s Backscatter X-ray Systems.....	4
Figure 2 ASE's various types of applications .....	5
Figure 3 Backscatter X-ray Images obtained from the Z Portal scanner .....	5
Figure 4 L3 Millimeter Wave Scanner .....	7
Figure 5 Rapiscan Systems' Metor 6M metal detectors with independent zone detectors .....	8
Figure 6 Dipole locations for a passenger car .....	10
Figure 7 Dipole locations for a class 8 truck .....	10
Figure 8 Initial Concept of the National Security Workstation .....	12
Figure 9 Initial Concept of the National Security Workstation Sensor Array .....	14
Figure 10 Concept Design of the Scanning System.....	15
Figure 11 Sensor and Scanning Assembly.....	15
Figure 12 Thesis Project Layout .....	18
Figure 13 Scanning System and Test Set Up.....	20
Figure 14 Door Fixture .....	20
Figure 15 Scanning Assembly .....	21
Figure 16 Motorized Carriage.....	21
Figure 17 Carriage Displaying the Self Aligning Ball Transfers.....	22
Figure 18 Scanning Platform with the Motorized Carriage .....	23
Figure 19 Sensor Assembly and Enclosure.....	23
Figure 20 Honeywell HMC2003 Magneto-Resistive Sensor.....	24
Figure 21 Sharp GP2D120XJ00F Infrared Sensor .....	25
Figure 22 Sensor Assembly with Sensors.....	25
Figure 23 Wiring Layout .....	26
Figure 24 Computer Workstation .....	26
Figure 25 National Instrument Multifunctional Data Acquisition PCI Card .....	27
Figure 26 Data Acquisition Set Up.....	27
Figure 27 Ferrous Objects in the Door Panel.....	29
Figure 28 Test Matrix Set Up .....	30
Figure 29 Euclidean and Mahalanobis Distances .....	33
Figure 30 Mahalanobis Distance for Classification .....	38
Figure 31 Augmented Mahalanobis distance for Classification .....	38
Figure 32 Threshold and Limit Index for Classification.....	39
Figure 33 Magnetic signature of base door panel .....	40
Figure 34 Normalized Cross-Correlation.....	41
Figure 35 Magnetic Field Strength and Energy Density.....	42
Figure 36 Mahalanobis Distribution (Function 1) .....	44
Figure 37 Mahalanobis Distribution (Function 2) .....	44
Figure 38 Mahalanobis Distribution (Function 3) .....	44
Figure 39 Mahalanobis Distribution (Test cases 1, 2 and 3).....	45
Figure 40 Mahalanobis Distribution (Test cases 4, 5 and 6).....	46
Figure 41 Mahalanobis Distribution (Offsets).....	47

Figure 42 Mahalanobis Distribution (Spikes).....	47
Figure 43 Threshold and Limit Index for Classification.....	48
Figure 44 Magnetic Signatures for E-W direction.....	49
Figure 45 Mahalanobis Distribution for E-W direction.....	50
Figure 46 Mahalanobis Distribution for varying proximity.....	50
Figure 47 Threshold and Limit Index for varying proximity.....	51
Figure 48 Orientation of AMR of the magnetic sensor .....	65
Figure 49 HMC2003 Set/Reset Strap Driver .....	66
Figure 50 HMC2003 Prototype Set/Reset Circuit .....	67
Figure 51 LabView Data Acquisition Program .....	69

## 1. Background and Motivation

Countless of terrorist attacks in the past decades have scarred the world. The September 11th attack in 2001 caused enormous destructions in New York and Washington D.C. and resulted in a death toll of over 5,000 people. In March 2012, a terrorist-related car explosion in Damascus and Aleppo, Syria, killed over 20 individuals and injured over 100 people. Due to past attacks, U.S. and other foreign governments have increased their efforts in counter terrorism strategies through anti-terrorism intelligences and technologies to prevent similar attacks from occurring. Though extensive efforts had been devoted to the research and development of weapon detection technologies, the public is still faced with terrorist attacks as seen in the Syria event. In this research project, a magnetometry-based weapon detection system was investigated in laboratory settings for on-road applications. The goal of this research is to design a simple mechanism with off-the-shelf electronic components that can be used to scan door panels to validate the concept of a concealed ferrous object detection system.

### 1.1 Literature Survey

Weapon detection has been a challenge for both academic researchers and defense companies for decades. As technology advances so does the development of many different methodologies to detecting and localizing concealed weapons. Concealed weapon detection systems can be categorized based on the following types of application defined by the National Institute of Justice: detection, imaging and locating [1]. In the next few sections, a basic overview of each type of application is further explained with examples of currently used technologies.

#### 1.1.1 Detection

As defined in [1], detection applications are the most basic form. Detection gives the operator information on the presence of objects in the detection space, where the space is defined as the volume in which the operator is searching. A detection system is a system that will provide the least information since it will only alert the operator if a concealed object is currently present within

the detection space. An example of a detection only system would be a hand-held metal detector or a stationary metal detector at airport security check point.

#### 1.1.2 Imaging

Imaging systems require acquiring images of the detection space and using image recognition to indicate whether a concealed weapon is present. The indications could be a visual or audible alarm. Imaging systems are very expensive due to large computing power requirement for data processing. An imaging system will include one or more imaging sensors, scanning structure, computing hardware, software, and display. Although an imaging system is more expensive, the information provided can be more useful than a detection only system [1].

#### 1.1.3 Locating

A locating system is used to show the operator where a concealed contraband item or weapon is hidden within the detection space. Although locating systems do not require an image, an image type system can be used to locate objects. The cost of a locating system and its complexity ranges closer to a detection system because a locating system contains a group of detectors where the acquisition of information is rapid [1].

Due to current advancements in technology and research, a hybrid type system that can detect, produce an image, and locate the concealed weapon can be achieved through the combination of two or more types of sensors and signal processing techniques. In this literature survey, different type of systems that may fall under one or more of the three types of applications defined by the National Institute of Justice will be briefly explored. The goal of this literature survey is to identify the following:

1. Identify the variety of weapon detection system readily available in the market and the potential costs for purchase and implementation.
2. Identify the challenges of each of the weapon detection system as they are readily available.
3. Identify the different applications of magnetic sensors and the possibility of applying them to weapon detection in vehicles.

#### 1.1.4 Available Weapon Detection Systems

There are a variety of weapon detection systems readily available in the market for use by airports, US Border Protection Agency, military, and public law enforcers. These systems range from the simplest forms, like hand held metal detectors used at the entrance of a concert to the most complex, like robot X-ray inspection systems roaming the streets of a highly populated downtown area inspecting suspicious vehicles. In this literature survey, X-ray inspection systems, millimeter wave scanners, and metal detectors will be reviewed since they are the most common in application.

##### 1.1.4.1 X-Ray Inspection Systems

X-ray was accidentally discovered in 1895 by a German physicist, Wilhelm Conrad Rontgen while experimenting with electrical currents through glass cathode ray tubes. Today, there are numerous of advancements in X-ray technologies that are applied to the field of medicine, raw material manufacturing, and national security. In national security applications, backscatter X-ray technology is commonly used to detect conceal weapons or contrabands.

In traditional X-ray systems, the imaging sensor and X-ray tube sandwiches the object of interest. As opposed to traditional X-ray scanners, the X-ray tube and the imaging sensor are on the same side in a backscatter X-ray system. Backscatter X-rays are much weaker compared to the traditional X-ray machines used at the doctor's or radiologist's office. The radiation of the backscatter X-ray system does not go through the flesh and bones like traditional X-ray machines, but instead reflects depending on the density of the material. The reflected X-rays are then captured by the imaging sensor to produce an image for visual analysis. Any concealed foreign objects carried by the individual can be detected in the image produced by the system.

The backscatter X-ray technology is currently used in many national security applications from body scanners at the airport to large scale cargo scanning systems. Common application includes the detection of explosives in parking structures by law enforcers and the detection of contrabands and drugs at borders by the US Customs and Border Protection as seen in Figure 1.





Figure 1 American Science and Engineering Inc.'s Backscatter X-ray Systems

Robot-towed AXISS Z Backscatter system scans a vehicle (top), vehicle going through the Z Portal Backscatter system (bottom).

Photo courtesy of ASE.

Currently, American Science and Engineering Inc. (ASE) have designs of their backscatter X-ray scanners available for various security applications i.e. car and semi-truck scanners. These security applications include, gantry or drive-through applications and mobile applications for on-the-go and on-the-road inspection as see in Figure 2. Similar to traditional X-ray images, the backscatter X-ray system produces X-ray images that assist operators at borders or toll ways in finding contrabands and drugs as seen in Figure 3.



Figure 2 ASE's various types of applications

Mobile (top left), gantry (top right), and drive-through (bottom).

Photo courtesy of ASE.

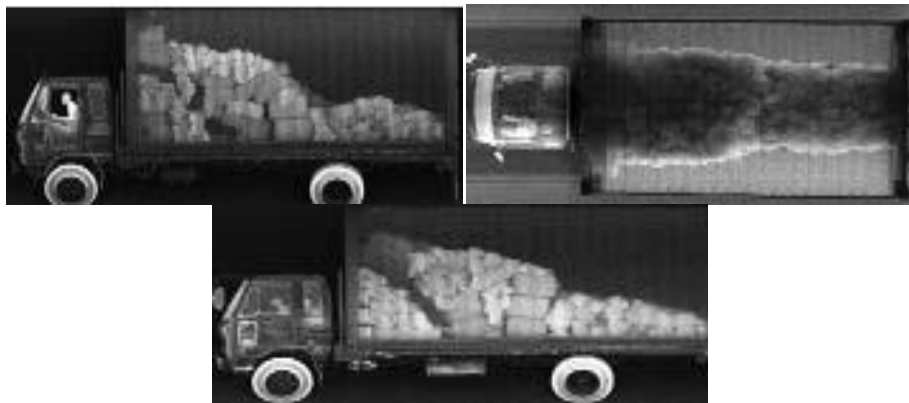


Figure 3 Backscatter X-ray Images obtained from the Z Portal scanner

Driver side (top left), top (top right) and passenger side (bottom).

Photo courtesy of ASE.

Detecting and locating these contrabands, drugs, or weapons in the backscatter X-ray scanner relies heavily on trained security operators. The information regarding the costs and accuracy of a backscatter X-ray system could not be found through this literature survey.

#### 1.1.4.2 Millimeter Wave Scanners

A competing technology of the backscatter X-ray scanner is the millimeter wave scanner, which is used for body scanning to detect contrabands, drugs or weapons concealed underneath clothing. Millimeter wave scanners are mainly used at security checks points like those used at the airports. Millimeter waves are ideal for scanning technologies because its energy passes through clothing and therefore capable of detecting concealed contrabands, weapons or drugs. Although millimeter wave scanners work similar to backscatter X-ray scanners because of its use of electromagnetic radiation to obtain images, millimeter wave scanners differ in other of manners.

The two differences of the millimeter wave scanner lies in the type of electromagnetic radiation the system emits, and the operation and layout of the scanner. Instead of emitting ionizing radiation like the backscatter X-ray scanner, the scanner emits millimeter wave, hence the name millimeter wave scanner. The electronics are set up in a vertical arrays, where each electronic component in the array can act as a transmitter and receiver during the scan. A millimeter wave scanner has two vertical arrays of transmitter/receiver that are 180 degrees from one another and would rotate around the individual being inspected.

There are only a handful of companies around the world that have produced and implemented the millimeter wave scanner like the one similar to the ProVision ATD Millimeter Wave Scanner by L3. As can be seen in the Figure 4, the scanner rotates around the individual being inspected as the individual become stationary in the scanner. Prior to the scan, the individual must raise both hands above the head. After the scan, the scanner produces an analysis image indicating possible areas where concealed items may be located, which security operators can further search. If the scanner does not detect any foreign objects after a scan, the system produces a notification and gives the security operator the “OK” to release the individual.

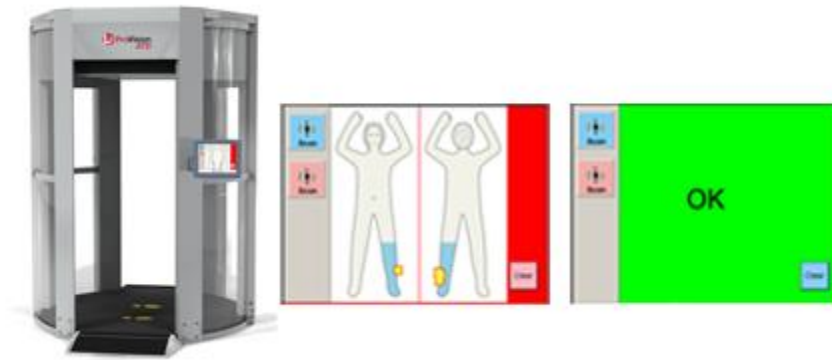


Figure 4 L3 Millimeter Wave Scanner

Scanner (left), image produced if possible concealed items are detected (middle), and image produced after passing a scan (right).

Photo courtesy of L-3 Communications.

The millimeter wave scanner have the ability to detect and locate on the image the system produces. Information regarding the pricing and accuracy of millimeter wave systems could not be found through the literature survey.

#### 1.1.4.3 Metal Detectors

Metal Detectors are used in a variety of applications ranging from airport security check points to metal finders used on a beach to find a lost property like jewelry. These metal detectors uses a technology called pulse induction. Pulse induction systems use a single coil of wire on one side of the metal detector. The single coil of wire acts as the transmitter and receiver for the system. During normal operations, the transmitter sends a powerful and short burst of current through the wire and generates a momentary magnetic field in the area of detection. When the current stops, the magnetic field reverse its polarity and suddenly collapse. This creates a spike in current known as the reflected pulse that lasts about 30 microseconds. Once the reflected pulse collapse, the process restarts. In a case where metal objects are in the detection space, the reflected pulse takes longer to collapse.

There are many walk through metal detector manufacturers in the world, and one popular manufacturer is Rapiscan Systems, who designed the Meteor 6M in Figure 5. The Meteor 6M is

capable of independent zone detection, allowing more accurate detection and automatic locating of the concealed metal objects. The Metor 6M features an accurate 20 segment zone, allowing security personnel to perform quick searches from the system's visual alarms.



Figure 5 Rapiscan Systems' Metor 6M metal detectors with independent zone detectors

Photo courtesy of Rapiscan Systems.

With today's advancements in security technology, metal detectors have become more than just detectors but also locating systems. The official costs of the Rapiscan Systems Metor 6M has not been publicly reported, but other similar walk through metal detectors found on the web costs up to \$5,500.

From the simplest concealed weapon detection system such as a metal detector to the most complex like the backscatter X-ray system, there are many options to select from depending on the application and cost. There are been numerous of advancements in technology today that can be beneficial to other types of industry. For example, magnetic sensor technology used in crash avoidance or traffic control applications can be utilized for security type applications. For the remaining of the literature survey, magnetic sensors and its application will be briefly reviewed. A project developed at California Polytechnic State University, San Luis Obispo that uses magnetic sensors will also be reviewed.

## 1.2 Magnetic Sensors and Its Applications

Magnetic sensors have been used in a variety of applications such as steel manufacturing, traffic control, and military type applications. In steel manufacturing industry, magnetic sensors are used to detect flaws such as cracks in parts as they leave their steel casting process. In the military, magnetic sensors are used to detect buried unexploded ordnance in missile testing areas and detect the presence of other submarines in the ocean. On city roads, magnetic sensors are used to detect the presence of vehicles at stop lights for traffic control. Magnetic sensor technology has been used in many applications and has proven its ability in each field. Other applications such as vehicle detection for crash avoidance also uses magnetic sensors to increase its detection rate in a sensor fusion application, as proven in the Sensor Integration for Low Cost Crash Avoidance project in California Polytechnic State University, San Luis Obispo. The project is briefly reviewed in the next section.

### 1.2.1 Sensor Integration for Low Cost Crash Avoidance Project

In 2009, a crash avoidance system was developed by master student, Stephane Roussel and research professors, utilizing the information obtained from ultrasonic and magnetic sensors to assist semi-truck drivers in maneuvering their trucks and reducing accidents. The overall goal is to utilize off-the-shelf sensors to reduce the cost and integrate the information to alarm the truck drivers. Through rigorous research and development, Bayes algorithm was applied to the information provided by the sensors to create a probabilistic belief when the system had detected a vehicle lateral to the vehicle carrying the sensor system. After full scale system testing and tuning, a sensor system that was fully operational in assisting semi-truck drivers was developed.

One of the key components that made the crash avoidance system highly successful was magnetic sensors, which motivated this thesis project. The magnetic sensor is significant in the development of the crash avoidance system due to the fact that all vehicles consist of concentrated ferrous metal and will always produce a change in the geomagnetic field. Through mathematic modeling and field testing for passenger and truck class vehicles as seen in Figure 6 and Figure 7, dipole locations

are closely related to the metal concentration areas such as, axles, engine manifold, etc. The result showed the capability of magnetic sensors detecting concentrated metals. The ability of the magnetic sensor could be utilized to detect the presence of concealed firearms and other ferrous metals. The idea of investigating a magnetometry-based system for detecting conceal ferrous objects in vehicles came to mind using the key component discovered in Stephane's research. In the next section, induced magnetic fields will be introduced and will explain how capturing the magnetic signatures will help in detecting conceal ferrous objects in door panels.

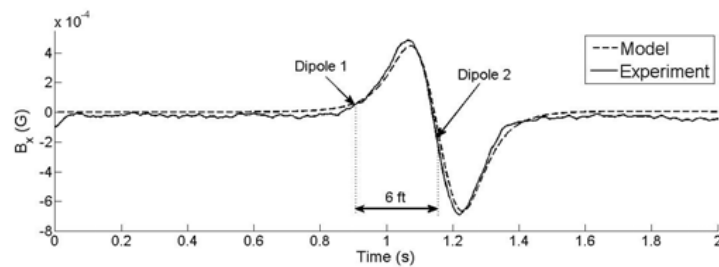


Figure 6 Dipole locations for a passenger car

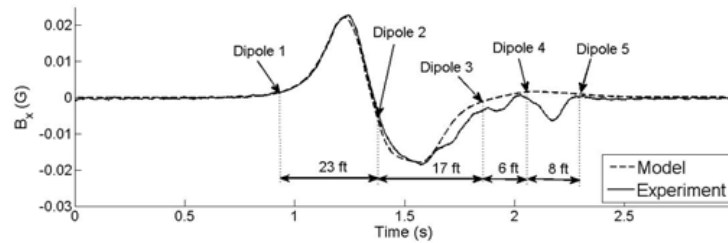


Figure 7 Dipole locations for a class 8 truck

### 1.3 Magnetic Materials and Its Induced Magnetic Fields

All metal creates an induced magnetic field as the material is exposed to a magnetic field, i.e. geomagnetic field or earth's magnetic field. The induced magnetic field can be significantly noticeable like those created from ferrous material. The induced magnetic field is the result of the external magnetic field changing the direction of the electrons' spin at the atomic level of the atom. Though not all materials' magnetic field results from induction, ferromagnetic material exhibits a net magnetization even without the external field. This is called remnant magnetization. The

magnetometer sensor readings are the result of the superposition of the external and induced magnetic field.

To detect conceal ferrous objects, for example in door panels (also made from ferrous material), a base magnetic signature of the door panel or vehicle must be obtained and stored in a database as a reference. It is important to have a reference because all ferrous objects will produced an induced magnetic field, hence a magnetic signature. If a ferrous object is introduced to the door panel, the magnetic signature changes. When comparing the reference magnetic signature with the magnetic signature of the door panel with a concealed ferrous object, differences in the signals will exist. In the opposite case, where there is no change to the door panel, the magnetic signature will be the same as the reference. With this motivation, the Extramural Funding Initiative proposal was put together to fund the development of a system that detects suspicious ferrous items in vehicles.

#### 1.4 The Extramural Funding Initiative Proposal

The Cal Poly Extramural Funding Initiative provides funding opportunities for newly hired faculty members who are interested in pursuing research and scholarly activities, and for established faculty members who are seeking to open new scholarly directions. The concept of using magnetic sensors for detecting weapons in door panels is a new research direction for weapon detection, therefore a proposal with a detailed system design was submitted. The detailed proposal was approved by Cal Poly's Office of Research and Economic Development. The funding allowed for the purchase of all items to support and design the research system, along with the labor required for building and testing.

The initial proposed system consisted of a portable national security workstation that has the ability to investigate the use of magnetic sensors to detect concealed ferrous objects in vehicles, and investigate other sensors that will be beneficial to national security type applications. After providing an initial concept design and detailed implementation plan in the proposal, the project was approved for funding and allowed the detailed design and design iterations to begin.



#### 1.4.1 Concept Design

The initial proposed system consisted of a portable national security workstation that has the ability to investigate the concept of magnetic sensors used to detect concealed ferrous objects in vehicle, and investigate other sensors that will be beneficial to national security type applications. The portable national security workstation would be designed to be deploy at tolls and border crossings. The initial design of the workstation consisted of the three main components: main power unit, sensor frame assembly, and primary frame assembly. The main power unit will supply the power to the sensors and actuators, and store the data acquisition system that processes the information obtained from the sensors. The sensor frame assembly will house the sensor arrays, i.e. magnetic sensors, ultrasonic sensors, digital cameras, and biosensors. The primary frame assembly will be the support structure for the sensor frame assembly as seen in Figure 8.

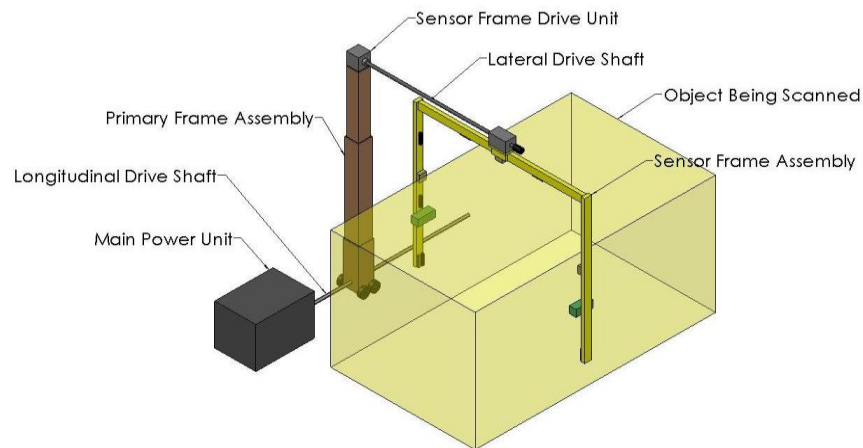


Figure 8 Initial Concept of the National Security Workstation

During normal operations, the sensor assembly in Figure 9 would scan along the length of the object that is powered by the longitudinal drive shaft. The sensor frame drive unit would adjust the sensor frame assembly in the appropriate position along the width of the object being scanned. The sensor array consist of magnetic sensors, ultrasonic sensors, biosensors, and digital cameras. During the scan, data from all the sensors will be acquired and analyzed by the onboard computer system

within the power unit to detect the anomaly in the magnetic profiles. The purpose of each of the sensors are listed below.

#### Magnetic Sensors

The magnetic sensors will be used for the main purpose of this project. The magnetic sensors will capture the magnetic profile of the vehicle or object being scanned and stored in the computer system. The magnetic profile will then be compared with a reference magnetic profile to check for any anomaly.

#### Ultrasonic Sensors

The ultrasonic sensors will be used to track the distance from the sensor to the vehicle. The distance will be used to adjust the reference magnetic profile readings for comparison with the actual field magnetic profile readings.

#### Biosensors

The biosensors will be used to detect contrabands and drugs around the vehicle.

#### Digital Cameras

The digital cameras will be used for three purposes. The cameras will be used to capture photos and videos, and track drivers and passengers who passes through tollways and borders. The camera will enable biometric analysis to alert law enforcements when a fugitive attempts to pass through. Lastly, the digital cameras will be used to monitor drivers and passengers for any strange behaviors during vehicle scanning that would signal suspicion and the need for further search.

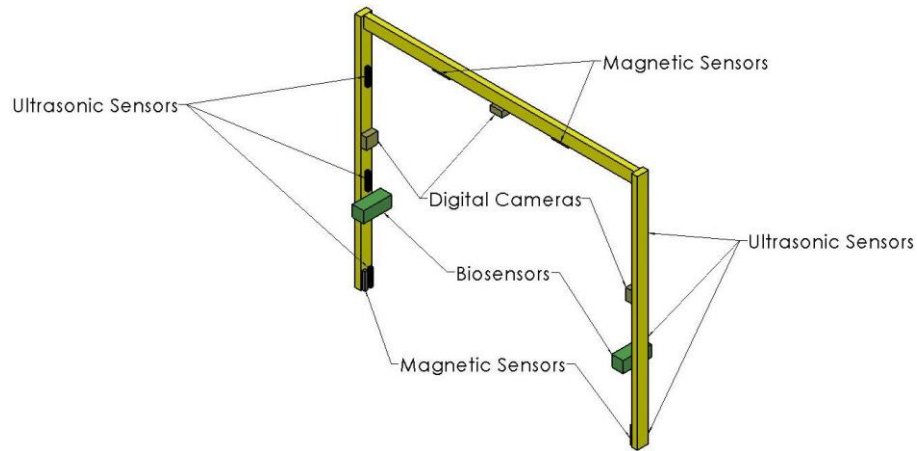


Figure 9 Initial Concept of the National Security Workstation Sensor Array

### 1.5 Final Concept Design

As the goal of the thesis project was brought to focus, the concept design changed accordingly. The goal of this thesis project is to develop a portable workstation to further investigate the use of magnetic sensors to detect concealed ferrous objects in vehicles for national security applications. With this goal, biosensors and digital cameras were removed from the sensor array though they are necessary for a well-rounded intelligent system. The final system would be a simple system that is portable for field testing purposes that satisfies the concept design criteria listed below.

#### Concept Design Criteria

1. Minimum moving parts to reduce maintenance purposes.
2. Minimum power usage so it can be easily used in the field for testing.
3. Sensitive magnetic sensors.
4. Portable systems that can easily be set up in the field by one person.
5. System can be used for lab testing as well as field testing.
6. Consist of computer system that can acquire data consistently based on the distance travelled by the scanning system (not time based data sample but spatial).
7. Data to be stored in files that can be easily accessible with software such as MatLab and Microsoft Excel.
8. Sensors will be enclosed for protection but shall not affect their readings.

With the design criteria in mind, a scanning platform that guides a sensor system was designed as seen in Figure 10 and Figure 11. A ground based platform was proposed due to its simplicity. To obtain data, the scanning assembly and sensor system starts from the front end of the vehicle and

stops at the back end of the vehicle. During the scan, the computer system stores data from the magnetic and distance sensors into a text file for post processing and investigation.

A total of three magnetic sensor and three distance sensor are used. The magnetic and distance sensor are paired together in one enclosure for protection from falling or hitting the object during the scan. As seen in Figure 11, the sensor and scanning assembly relies on a motor, a timing belt and pulley system, and wheels for motion. With the guiding rail system on the scanning platform, the sensor and scanning assembly stays parallel with the object being scanned.

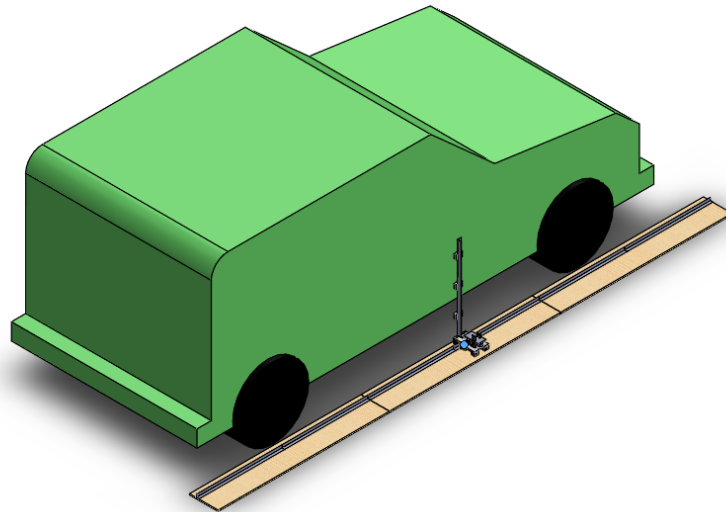


Figure 10 Concept Design of the Scanning System

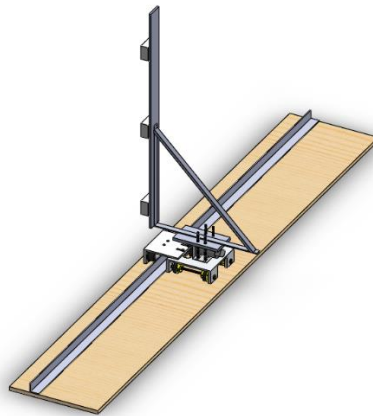


Figure 11 Sensor and Scanning Assembly

In addition to satisfying the design criteria, a door platform was designed and built to mount the scanning system and a test door panel for testing in lab environment. The purpose of the door platform is to enable a controlled environment test of detecting concealed ferrous objects in car doors. The detailed descriptions of each major part of the system will be explained in the next chapter.

## 1.6 Project Overview

The goal of this project thesis is to investigate and validate the use of magnetometry-based technology to detect and localize anomalous ferrous objects in vehicles, specifically in door panels and fenders. Detection occurs after the comparison of a reference magnetic signature (the magnetic signature with no object concealed) and the scanned sample magnetic signature of the vehicle's door panel. In this research paper, magnetic signature, magnetic footprint and magnetic profile will be used interchangeably. When there are large discrepancies between the two magnetic signatures, the computer system pinpoints the location of the discrepancy which signals the location of the concealed weapon or object.

The project was designed into two phases. The first phase included the design and build of the research prototype system used as a platform to investigate the use of magnetometers for detecting concealed ferrous objects. The second phase incorporated the experimental validation of using magnetometry-based technology for concealed ferrous object detection. Basic statistical algorithms, Mahalanobi distance and cross-correlation were implemented to detect the changes in the magnetic signatures.

In the first phase, a research prototype system was designed to investigate and validate the use of magnetometry-based technology for concealed weapon detection in vehicles. For validation, a test door was used instead of a vehicle to prove that the technology is capable for this application. The prototype system designed can be categorized into four main sub-systems: scanning assembly, door platform, sensor assembly, and computer and data acquisition system. The scanning assembly consisted of a motorized four wheel carriage that travels along the length of the test door. The

scanning assembly provides a platform to mount the magnetic sensor array and record the spatial magnetic signature of the test door mounted onto the door platform. The door platform provides a fixture to hold the door panel at an upright position similar to a mounted door on a vehicle, and provides a stage to mount the scanning assembly. Casters were mounted under the door platform for mobility and enabled different scanning orientation in earth's magnetic field. The computer and data acquisition is the last sub-system and allows data logging of the magnetic sensors.

## 2. Design and Build of Research Prototype System

The thesis project consist of an integrated design to deliver a research system that enables the magnetic signature evaluation for conceal ferrous object detection purposes. The thesis project design process consist of the mechanical design and build of the research system, sensor evaluation and selection, computer build, electrical wiring, programming of data acquisition, and testing as can be seen in Figure 12. The search of magnetic sensors proved to be difficult due to the system requirements of its low cost and high sensitivity. When sensors that met the requirements were found, the sensors were then evaluated. After the purchase of the sensors, the design and build of electrical system began. When the electrical and mechanical system design and build were complete, the task of programming and testing the data acquisition system began. This is to ensure no mechanical, electrical, and software glitches were present in the system before the testing phase.

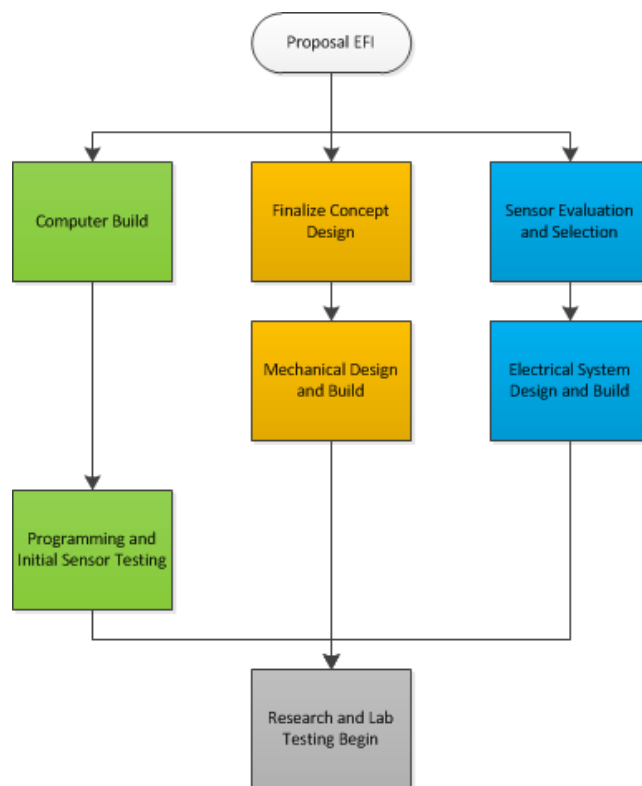


Figure 12 Thesis Project Layout

## 2.1 System Design and Build

The research prototype system is composed of four main sub-systems: scanning assembly, door platform, sensors assembly, and the computer and data acquisition system. The scanning assembly is a motorized carriage with the main purpose of carrying the sensors for scanning. The door platform provides a fixture to hold the door in an upright position similar to being mounted on a car, and a stage to mount the scanning assembly. To acquire the magnetic field data, three HMC2003 Honeywell magnetometers and three Sharp infrared sensors were utilized to capture the magnetic signature of the door and track the door-to-sensor distance. Each of the two sensors were enclosed in a clear plastic box for protection. To read the information from the sensors, a computer with a National Instruments PCI Multifunctional data acquisition card was used. The data acquisition card was programmed with LabView to acquire the necessary data for this project.

### 2.1.1 Mechanical Design

The mechanical system is composed of three main sub-systems: scanning assembly, sensor assembly, and door platform. The scanning assembly is a motorized carriage with the main purpose of carrying the sensors during the scan. The scanning assembly also contains the scanning platform, which is a wooden surface with aluminum rails to guide the scanning assembly in a linear motion during scanning. The sensor assembly provides a platform to mount the sensors. The door platform provides a fixture to hold the door in an upright position similar to being mounted on a car, and a stage to mount the scanning assembly for lab testing. The different sub-systems can be seen in Figure 13.



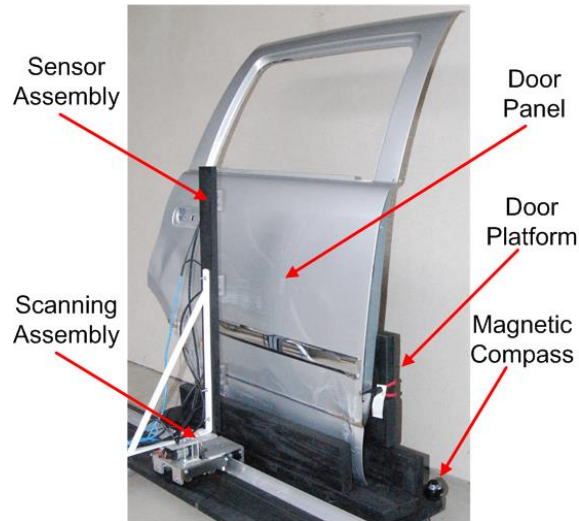


Figure 13 Scanning System and Test Set Up

#### 2.1.1.1 Door Platform

The door platform is a support platform to hold the test door panel upright similar to when door panels are mounted on a vehicle. The door platform also provides a stage to hold the scanning platform during testing. The door platform contains vertical support made from two by fours so the door panel can be secured onto the platform using expandable tie downs. Casters are used to easily transport the door platform and scanning assembly from and to its storage unit and testing room, and maneuver the test set-up during testing. A magnetic compass is also mounted on the door platform to track the scanning direction respective to the geomagnetic field during testing.

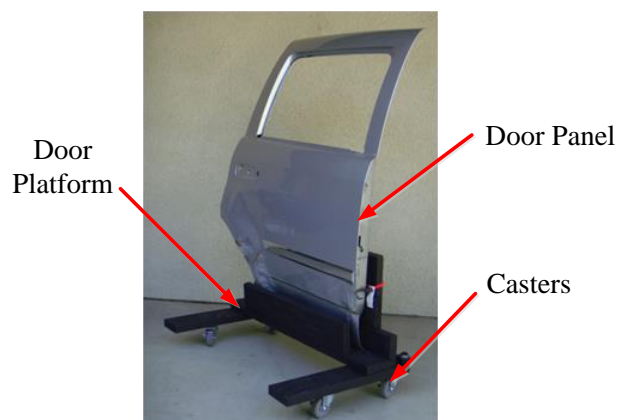


Figure 14 Door Fixture

### 2.1.1.2 Scanning Assembly

The scanning assembly consist of a motorized carriage, a sensor assembly, and an eight feet scanning platform which guides the carriage parallel to the test door during scanning as seen in Figure 15. The carriage runs along a straight line, and is guided by an aluminum rail secured onto the scanning platform. To move the carriage, a single motor with a rotary encoder drives the carriage by a timing belt and pulley system.

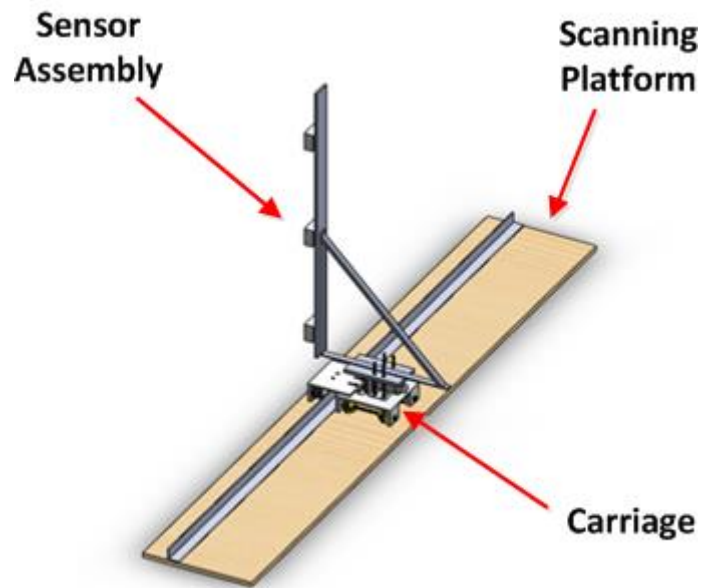


Figure 15 Scanning Assembly

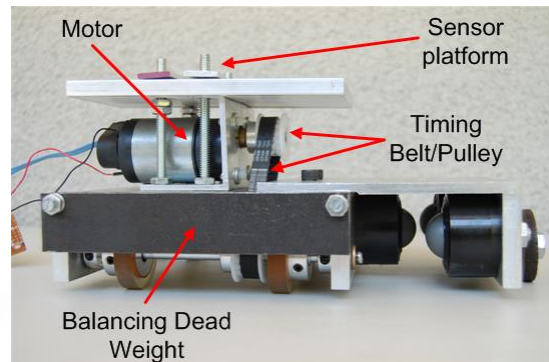
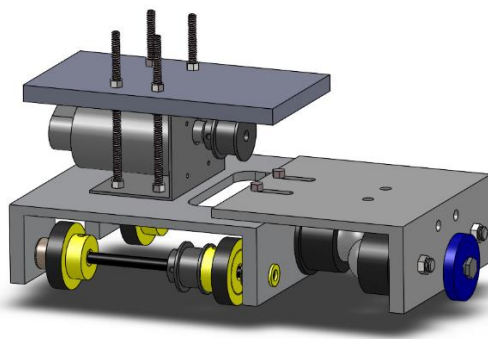


Figure 16 Motorized Carriage

#### 2.1.1.2.1 Carriage

The carriage is designed to have a compact foot print for easy transport and set up by a single operator. The carriage is a two shaft mechanical system with four wheels and shaft locking collars. Brass bearings were used for smooth rotary motion during movement. Collars were used to lock the shafts from shifting laterally during scanning. A 30-volt Pittman motor with an 11.5:1 gear box was carefully selected because of its compact size and torque output to drive the carriage. Due to the space constraint, two extra light miniature fourteen teeth acetal timing pulleys and neoprene timing belt were used to drive the carriage as seen in Figure 16. Balancing weights were mounted on the carriage to increase the weight of the carriage to reduce vibration of the sensor assembly during scanning. The carriage also contained a self-aligning system made from a precision machined L-bracket with nylon ball transfers. The self-aligning L-bracket can be adjusted to reduce friction during scanning as seen in Figure 17.

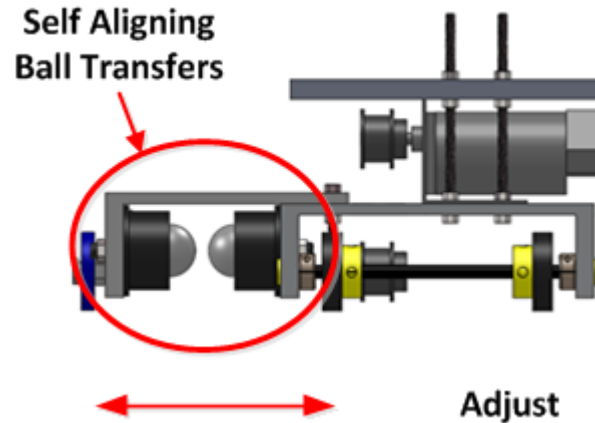


Figure 17 Carriage Displaying the Self Aligning Ball Transfers

#### 2.1.1.2.2 Scanning Platform

A sub-system of the scanning assembly is the scanning platform. The scanning platform is made from an eight feet standard wooden board with a one and a half inch L-brackets carefully mounted along the length of the board. The L-bracket's purpose is to make contact with the self-aligning ball transfer on the carriage. This is to keep the carriage parallel to the object being scanned when

the carriage is in motion. The size of the aluminum L-brackets were properly sized in order to clear the self-aligning system on the carriage. Plastic door stoppers are mounted on both ends of the scanning platform to prevent the carriage from over running the length of the scanning platform as seen in Figure 18.

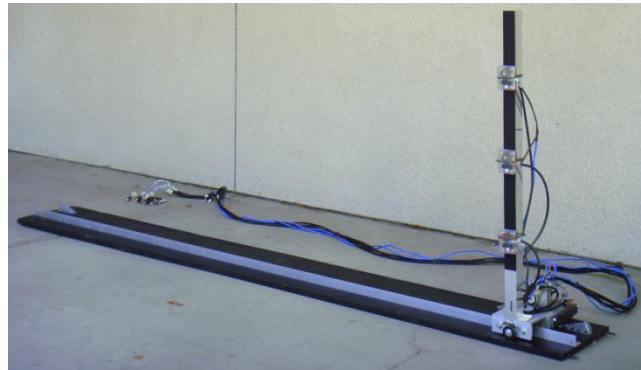


Figure 18 Scanning Platform with the Motorized Carriage

#### 2.1.1.2.3 Sensor Assembly

The sensor assembly consist of the enclosed sensors and the sensor mounting bracket. The enclosure is compact and light weight. Velcro is used to mount the enclosure on the sensor mounting bracket. The sensor mounting bracket is lightweight, and is three and a half feet in height. In order to create a sensor array, Velcro is used along the vertical length of the sensor assembly so the sensor enclosures can easily be moved for different sensor-to-sensor distance arrangements.

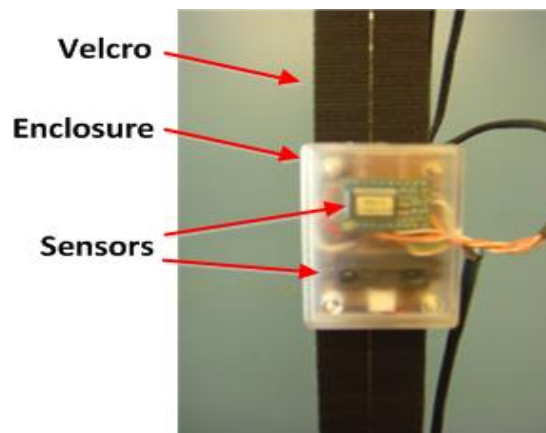


Figure 19 Sensor Assembly and Enclosure

### 2.1.2 Sensors

Two types of sensors were utilized in the research system to acquire the magnetic field strength and distance information, which include Honeywell's HMC2003 magneto-resistive magnetic sensors and Sharp's short range GP2D120XJ00F infrared sensors. The HMC2003 magnetometers were used to detect the magnetic field strengths around the test door and the infrared sensors were used to measure the distance from the sensors to the test door.

#### 2.1.2.1 Magnetic Sensor

The Honeywell HMC2003 magnetic sensor is a precision three axis magnetometer with a detection range from -2 to +2 gauss and sensitivity of 1V/gauss, and has the capability to detect small magnetic field strengths down to 4 micro-gauss. The sensor was selected due to its small footprint, magnetic field strength resolutions, and analog outputs for each of the axes.



Figure 20 Honeywell HMC2003 Magneto-Resistive Sensor

Photo Courtesy of Honeywell.

#### 2.1.2.2 Infrared Sensor

The Sharp GP2D120XJ00F infrared sensor is an inexpensive optical distance finder with detection range of 1.5 inches to 12 inches, and has a 1-inch resolution. [2] This inexpensive sensor was selected due to its ability to detect objects at short range, its output readings in analog voltages, and its small form factor.



Figure 21 Sharp GP2D120XJ00F Infrared Sensor

Photo Courtesy of Sharp.

Each sensor are paired to detect magnetic field strengths of nearby ferrous objects and the distance of the test door. The Sharp infrared sensor detects object distance as well as the presence of an object making magnetic signature comparison simpler. The sensor assembly with a close up of the sensors can be viewed in Figure 22.

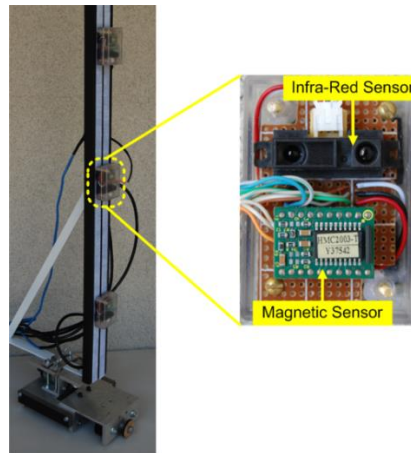


Figure 22 Sensor Assembly with Sensors

### 2.1.3 Electrical System Design and Build

After all the sensors were purchased, the electrical system design and build process began. CAT-7 wiring were used for the wiring system due to its compact size and 4 pair strand copper with shield. The CAT-7 provided the necessary electrical connections required for each sensor enclosure, and the cables were light weight. Three twenty feet CAT-7 cables were wired to the sensor enclosures and another to provide power to all the sensors, motor, and encoder which can be seen in the wiring layout in Figure 23.

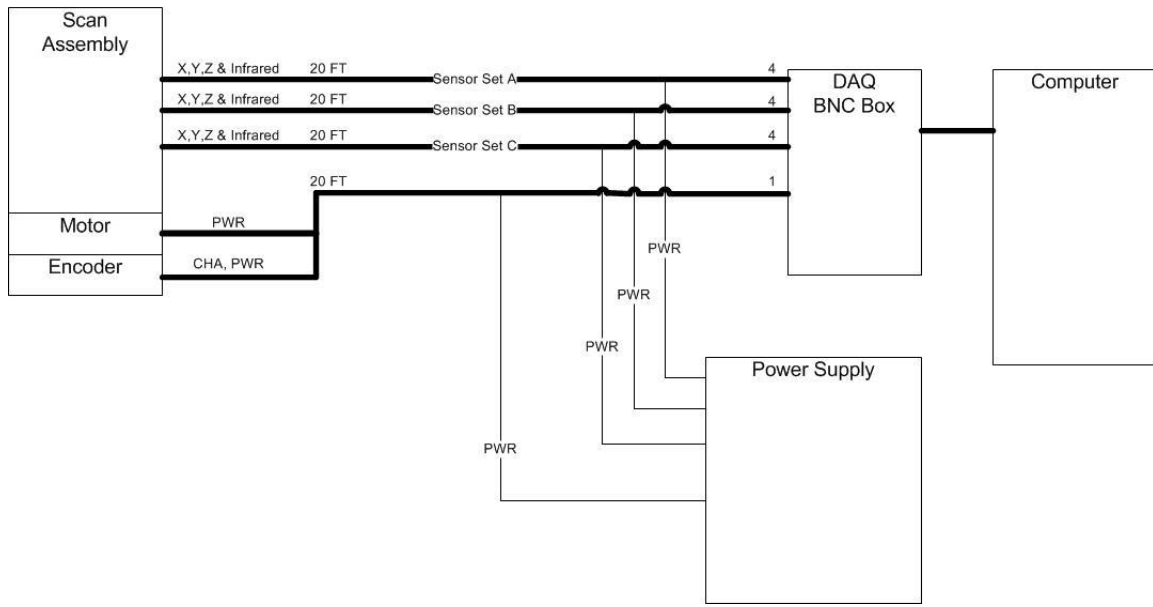


Figure 23 Wiring Layout

#### 2.1.4 Computer Workstation

A workstation was put together to supply the power to the scanning system, and acquire and process the data from the sensors as seen in Figure 24. Multiple regulated power supplies were mounted on the workstation to power all the sensors, motor, and encoder on the carriage. To acquire and process the information from the magnetic and infrared sensor, a computer was built with a National Instruments PCI-6259 Multifunctional data acquisition card. The details regarding the hardware and software installed can be viewed in Appendix III.

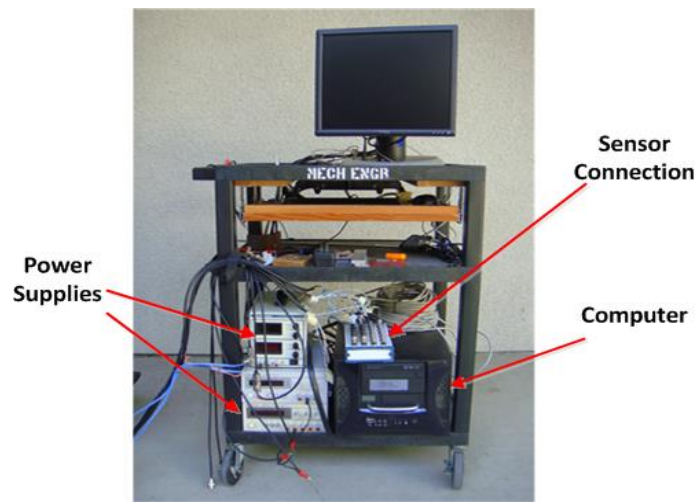


Figure 24 Computer Workstation



### 2.1.5 Data Acquisition System Programming

Three magneto-resistive sensors and three infrared sensors were utilized in this research. To automatically read the analog voltages from each of the sensors, a National Instrument PCI-6259 Multifunctional data acquisition card in Figure 25 was used and programmed with LabView. The data acquisition card has I/O capability of 32 analog inputs with a 16 bit analog-to-digital converter, and 48 programmable digital I/O. This data acquisition board supports external digital inputs for the sampling clock, which is implemented in this project.

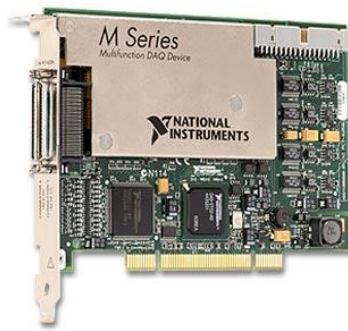


Figure 25 National Instrument Multifunctional Data Acquisition PCI Card

Photo Courtesy of National Instruments.

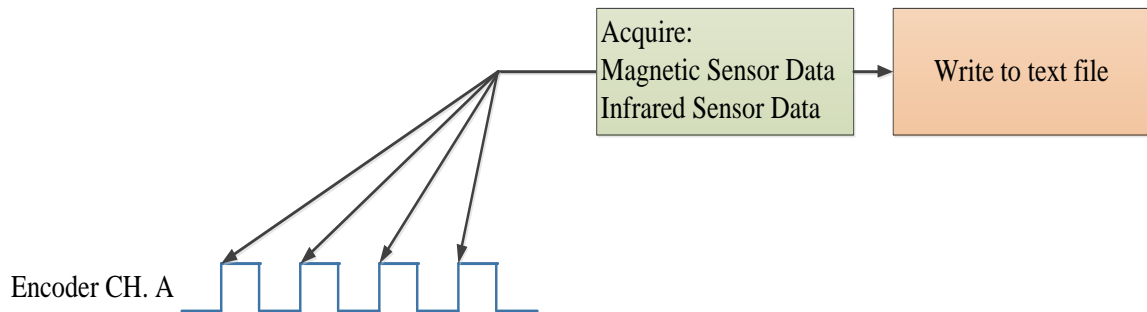


Figure 26 Data Acquisition Set Up

Since magnetic fields of test objects are a function of space (assuming that time variation of earth's geomagnetic field is constant over the test period, and tests are performed in the absences of power lines and transformers that produces large time varying magnetic fields), sampling the analog inputs based on position is essential for capturing magnetic signatures of the test door. To sample in this manner, channel A of the encoder was used to drive the sample clock on the data acquisition board.



At every rising edge of the encoder output or every  $7.8\text{e}^{-3}$  inch the carriage travels. All 12 analog inputs are acquired and logged to a file before the next encoder pulse as shown in Figure 26. This gives uniform data as a function of space and independent from the speed of the scanning system.

### 3. Testing

The testing phase of this project incorporates an intensive procedure in obtaining information from the magnetic and infrared sensors. The next two sections will describe the test set-up and the test procedures of the thesis project.

#### 3.1 Test Door

The test door used in this research was from a sport utility vehicle. This door panel was used because of the large door panel access holes as can be seen in Figure 27. These access holes allowed easy placement of the ferrous objects that simulated concealed weapons. If a door panel with smaller access holes was used, modifications will be required, hence affecting the magnetic signature of the door. The goal is to not make modifications to the door panel so it is representative of a door used on an actual vehicle.

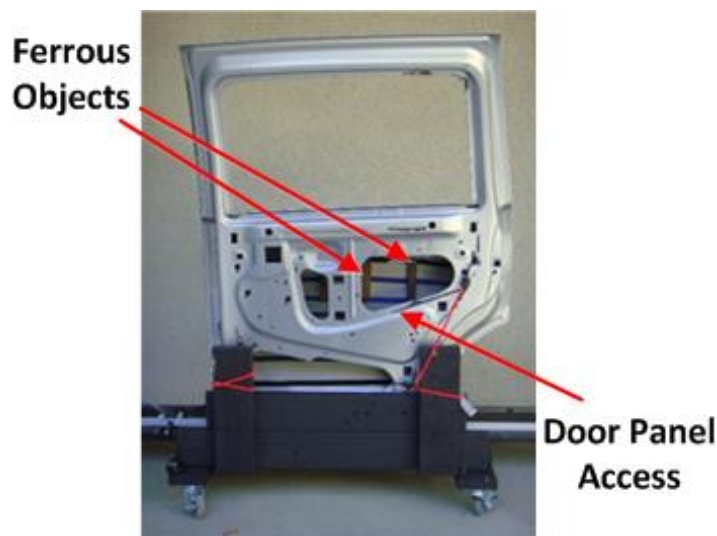


Figure 27 Ferrous Objects in the Door Panel

#### 3.2 Set-Up and Procedures

To obtain the magnetic signature reading of the test door, the scanning assembly is mounted onto the door platform where the distance of the scanning assembly to the door panel can be varied for different proximity tests. See Figure 13 for the test set up. Before the scanning assembly runs, the data acquisition software in LabView is activated to save the data into a file based on the unique

date and time. Once the scanning assembly begins the scan, the data acquisition system acquires all the sensor data at every pulse of the motor encoder as seen in Figure 26. When the scanning assembly reaches the end of the scanning platform, the operator terminates the data acquisition session as the file is saved on the computer hard-drive. The data acquisition system is then ready for the next scan.

In a controlled environment, the scanning platform was mounted onto the door platform. Prior to introducing test ferrous objects into the door panel, reference magnetic signatures of the door were acquired. Since magnetic signatures are spatial signals, the scans are only limited to one direction. When introducing ferrous objects to the door panel, the objects would be placed in different sections of the door using Velcro as seen in Figure 27.

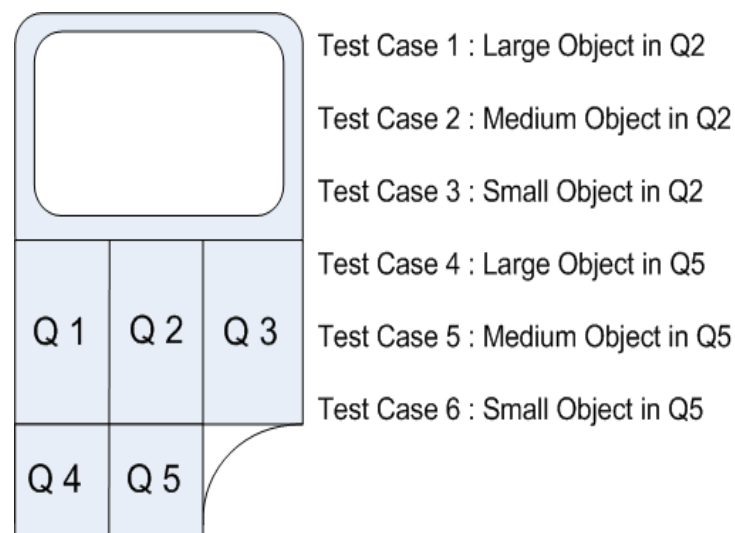


Figure 28 Test Matrix Set Up

As can be seen in Figure 28, the door panel was split into five sections and three different sized ferrous objects with volumes  $18 \text{ in}^3$ ,  $9 \text{ in}^3$ , and  $6 \text{ in}^3$  were chosen to represent the concealed object. The six test cases as shown in Figure 28 were chosen to represent a sample of the wide spectrum of possibilities.

#### 4. Analysis

In order to detect magnetic anomalies, the magnetic profile obtained from a scan is compared to a reference magnetic profile using signal processing and statistical methods. In this investigation, two signal processing and statistical methods, cross-correlation and Mahalanobi distance were identified to detect these changes in the magnetic signatures of the test door. Cross-correlation was considered since it's the most natural approach to evaluating the similarity between two signals. Similarly, Mahalanobis distance was utilized to help classify if a magnetic signature obtained from a scan contains a concealed ferrous object. The two indexes, Threshold Index and Limit Index were developed to evaluate the Mahalanobis distance for concealed ferrous objects and false positives, respectively. The following sub-sections will discuss the various signal and statistical processing techniques that were identified and developed through this research.

##### 4.1 Detection of Conceal Ferrous Objects Using Cross-Correlation

Cross-correlation is a measure of similarity of two waveforms as a function of a time-lag applied to one of them. It is commonly used for searching a long-duration signal for a shorter known feature. Hence, cross-correlation is useful when a certain signal  $x$  has to be compared with a set of prototype signals  $p$  in order to obtain a measure of mismatch between the unknown signal and a known pattern. Suppose  $y$  is a signal from the known pattern set  $p$ , then the Normalized Cross-Correlation can be quantify from the  $l^2$  norm, which is a measure of the mismatch between the signals  $x$  and  $y$  given by Equation 1.

$$d\langle x, y \rangle = \|x - y\|_2 = \|x\|^2 + \|y\|^2 - 2\langle x, y \rangle$$

Equation 1

where,

$y$  - prototype signal have unit norm  $\|y\| = 1$

When signal  $x$  matches prototype signal  $y$ ,  $d\langle x, y \rangle$  is minimized while the inner product,  $\langle x, y \rangle$  maximizes, hence inspiring the important method of matching signals, Normalized Cross-Correlation. For discrete signals, the Cross-Correlation between a test signal,  $x(n)$  and prototype signal,  $p(n)$  is given by Equation 2.

$$y(k) = \sum_{n=-\infty}^{\infty} p(n-k)x(n) = \sum_{n=k}^N p(n-k)x(n)$$

Equation 2

When  $y(k)$  is at its maximum value, the offset  $k$  corresponds to the location where the prototype pattern  $p(n)$  best matches the source signal  $x(n)$ . By using this information, the measure of match between the pattern  $p(n)$  and the signal  $x(n)$  on signal segment,  $[k, k+N]$  is then given by the Normalized Cross-Correlation function,

$$C_{p(n-k), x(n)} = \frac{y(k)}{\left[ \sum_{n=k}^N x^2(n) \right]^{1/2}} = \frac{\sum_{n=k}^N p(n-k)x(n)}{\left[ \sum_{n=k}^N x^2(n) \right]^{1/2}}$$

Equation 3

If the signal  $x$  matches exactly with the pattern  $p$ , the normalized cross-correlation becomes the maximum value of 1 and becomes -1 if  $x$  and  $p$  are mirror images of each other.

Hence, due to this pattern matching ability of normalized cross-correlation, it appears to be an attractive technique for detecting concealed ferrous objects, since magnetic signatures of vehicles show a well-defined pattern. When introducing ferrous objects into these vehicles, it alters this base pattern, therefore results in a non-correlation between the reference magnetic signature and test magnetic signature. The results of this technique are presented in the Results Chapter.

#### 4.2 Detection of Conceal Ferrous Objects Using Mahalanobi Distance

The Mahalanobis distance is a rule for calculating the distance between two points involving non-spherically symmetric data distributions, which is better adapted than the usual "Euclidian

distance". It is particularly useful when multi-normal distributions are involved, although its definition does not require the distributions to be normal.

The Mahalanobis distance metric is a very useful measure for determining the following:

1. Distance between the means of two distributions, which in this case helps with the determination of whether or not an object is concealed in the first place.
2. Distance of a point to the mean of a distribution, which can help with determining the size of the concealed object based on the proximity of the measurement to the mean.

Hence, Mahalanobis distance is an excellent measure for detecting and quantifying a concealed object. The remaining of the section provides a brief mathematical and graphical description of how it is utilized for the specific case of detecting and quantifying a concealed object.

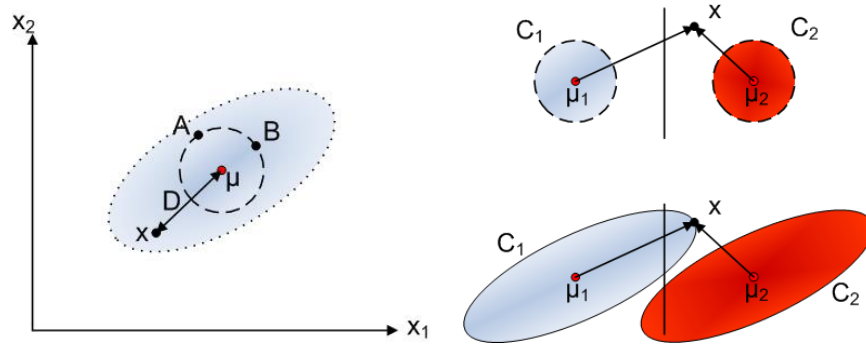


Figure 29 Euclidean and Mahalanobis Distances

Consider a distribution as shown in Figure 29. The quantities  $x_1$  and  $x_2$  are any measures that individually quantify a certain property of the system under consideration. In this case,  $x_1$  corresponds to the quantity  $\int (\Delta B)^2$  and  $x_2$  corresponds to  $\max \{(\Delta B)^2\}$ , where  $\Delta B$  is the change in magnetic field strength. Consider two points A and B that are equally distant from the center  $\mu$  of a distribution as shown in Figure 29. The Euclidean distance of point A from the mean is given by Equation 4.

$$d^2(A, \mu) = \sum_i (a_i - \mu_i)^2$$

Equation 4

where,

$a_i$  – observation  $A$  with  $i^{th}$  dimension

$\mu_i$  – mean of a class distribution with  $i^{th}$  dimension

$d$  – Euclidean distance from an observation,  $A$ , to the mean of a class,  $\mu$

$S_i$  – summation to the  $i^{th}$  dimensions

The Euclidean distance of point  $A$  from the mean is the same as the Euclidean distance of point  $B$  from the mean. However, it would be inappropriate to say that they occupy "equivalent" positions with respect to the mean since point  $B$  occurs in a higher density region than point  $A$ . Hence, the Euclidean distance is not the correct measure of the “distance” of any point to the mean value of the distribution.

Mahalanobis distance method considers two points as "equally distant from the mean" if both these points have the same probability density, making them equally probable when drawing observations from the distribution. Hence the modified distance formula is given by Equation 5.

$$D^2(x, \mu) = (x - \mu)^T \sum^{-1} (x - \mu)$$

Equation 5

where,

$x$  – an observation with  $i^{th}$  dimension

$\mu$  – mean of the class distribution with  $i^{th}$  dimension

$D$  – Mahalanobis distance from an observation to the mean of a class distribution

$S$  – covariance matrix for the sample set of a class

With  $S$ , the covariance matrix of the distribution is given by Equation 6.

$$\Sigma = \begin{pmatrix} \sigma_{11} & \bullet & \bullet & \bullet & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & & & \bullet \\ \bullet & & \bullet & & \bullet \\ \bullet & & & \bullet & \bullet \\ \sigma_{n1} & \bullet & \bullet & \bullet & \sigma_{nn} \end{pmatrix}$$

Equation 6

where,

$\sigma_{11}$  – variance for measure  $x_1$

$\sigma_{22}$  – variance for measure  $x_2$

$\sigma_{nn}$  – variance for measure  $x_n$

$\sigma_{12}$  – covariance for measures  $x_1$  and  $x_2$

$\sigma_{mn}$  – covariance for measures  $x_m$  and  $x_n$

#### 4.2.1 Mahalanobis Distance and Classification

To distinguish between two equally extended spherical classes with equal a priori probabilities, the best classification rule is simply to assign an observation  $x$  to the class whose center (mean) is closer to  $x$  in the sense of the ordinary Euclidian distance as shown in Figure 29.

But it is not so if the classes no longer spherical (although still multi-normal with identical covariance matrices). The observation,  $x$  should be assigned to the class to which it has the larger probability of belonging to, that is the class with the largest probability density in  $x$  (because of the equal a *priori* probabilities), and therefore to the class with the lower value of the Mahalanobis distance of  $x$  to the class mean. For example, in Figure 29, from a Euclidian point of view  $x$  should be assigned to class  $C_1$  although it is in " $C_2$  territory". Hence, in order to ensure that the observation  $x$  is assigned to the class whose mean is closer, the more appropriate measure is Mahalanobis distance. For the problem at hand, class  $C_1$  corresponds to the magnetic signatures of the door panel alone where class  $C_2$  now corresponds to the signatures of the same door panel now with a concealed ferrous object. Hence, Mahalanobis distance is an excellent approach to utilize in



determining if a magnetic scan belongs to class  $C_1$  or  $C_2$  thereby detecting the presence of concealed objects, if present.

#### 4.2.2 Mahalanobis Distance Between the Means of Two Classes

In the preceding section, the Mahalanobis distance between an observation and the mean of a distribution was defined. This distance provided information on the proximity of the observation to the mean value of the distribution.

The same definition can also be used for defining the Mahalanobis distance between the means of two distributions that we assume to be multinormal with identical covariance matrices  $\Sigma$  as shown in Figure 29. The mathematical representation of this distance,  $D_{I2}$  represents the distance between the two classes  $C_1$  and  $C_2$  given by Equation 7.

$$D_{I2}^2(\mu_1, \mu_2) = (\mu_2 - \mu_1)^T \Sigma^{-1} (\mu_2 - \mu_1)$$

Equation 7

where,

$\mu_1$  – mean of class  $C_1$  with  $i^{th}$  dimension

$\mu_2$  – mean of class  $C_2$  with  $i^{th}$  dimension

$D_{I2}$  – Mahalanobis distance between the means of two distributions  $C_1$  and  $C_2$ .

The common covariance matrix  $\Sigma$  can be computed as follows. Suppose  $n_1$  observations are drawn from  $C_1$ , and  $n_2$  observations are drawn from  $C_2$ . The common covariance matrix  $\Sigma$  is then estimated by  $\Sigma^*$ , the ponderated mean of the empirical covariance matrices of  $C_1$  and of  $C_2$ , denoted by  $\Sigma_1^*$  and  $\Sigma_2^*$  and is given by Equation 8.

$$\Sigma^* = \frac{n_1 \Sigma_1^* + n_2 \Sigma_2^*}{n_1 + n_2 - 2}$$

Equation 8

where,

$n_1$  – number of samples for class  $C_1$

$n_2$  – number of samples for class  $C_2$

$S_1^*$  - covariance matrix for class  $C_1$  based on the sample  $n_1$

$S_2^*$  - covariance matrix for class  $C_2$  based on the sample  $n_2$

This Mahalanobis distance  $D_{12}$  in conjunction with the previously obtained distance  $D_1$  and  $D_2$  can be utilized to help classify an observation as either belonging to class  $C_1$  or class  $C_2$ . To achieve this objective, a measure known as the Threshold Index ( $\psi$ ) defined in Equation 9 can be computed.

$$\psi = \frac{D_2}{(D_{12} - D_{1(\max)})}$$

Equation 9

where,

$D_2$  – Mahalanobis distance of an observation from the mean value of the distribution  $C_2$ .

$D_{12}$  – Mahalanobis distance between the means of two distributions  $C_1$  and  $C_2$ .

$D_{1\max}$  – Maximum value of the Mahalanobis distance of an observation from the mean value of the distribution  $C_1$ .

$0 \leq \psi < 1 \Rightarrow$  The observation lies closer to  $C_2$  than  $C_1$ .

$\psi > 1 \Rightarrow$  The observation lies closer to  $C_1$  than  $C_2$ .

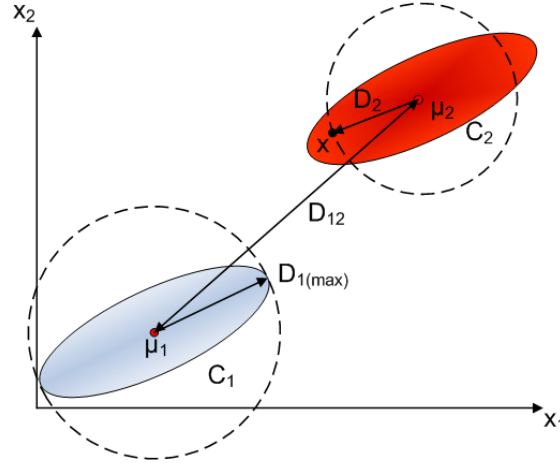


Figure 30 Mahalanobis Distance for Classification

In Figure 30 the concept of Mahalanobis distance was utilized to classify an observation as belonging to either class  $C_1$  or  $C_2$ . However, it is equally important to identify the occurrences of false positives (observations that don't belong to either  $C_1$  or  $C_2$ ). Mahalanobis distance could also be utilized to obtain a conservative measure of the occurrence of false positives as shown in Figure 31. The region of false positives can be determined by computing the Limit Index ( $\phi$ ) given by Equation 9.

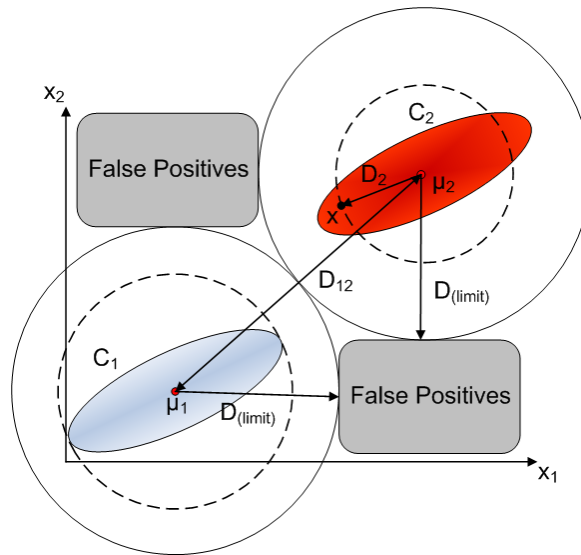


Figure 31 Augmented Mahalanobis distance for Classification

$$\varphi = \frac{D_{\text{limit}} - D_1}{D_2 - D_{\text{limit}}} \quad , \quad D_{\text{limit}} = \frac{D_{12}}{2}$$

Equation 10

where,

$D_1$  – Mahalanobis distance of an observation from the mean value of class  $C_1$

$D_2$  – Mahalanobis distance of an observation from the mean value of class  $C_2$

$D_{12}$  – Mahalanobis distance between the mean value of class  $C_1$  and mean value of class  $C_2$

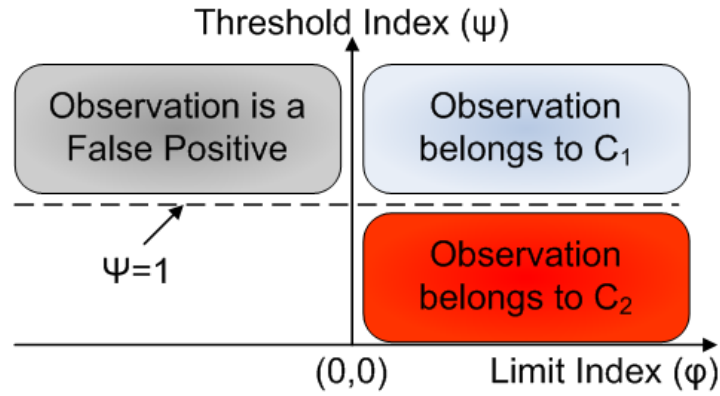


Figure 32 Threshold and Limit Index for Classification

If the Limit Index  $\varphi$  is less than zero, then it implies that the observation lies in one of the regions of the false positives and does not belong to either of the two classes  $C_1$  and  $C_2$ . Since the Threshold Index provides information on whether the observation belongs to class  $C_1$  or  $C_2$  and the Limit Index is able to identify the occurrence of false positives, they can be used in conjunction to fully determine both these results together as shown in Figure 32.

## 5. Results

In this study, an extensive experimental test protocol was employed for the proof-of-concept. All the three techniques; Normalized Cross-Correlation, Magnetic Field Strength/Energy Density, and Mahalanobis Distance were employed to post process the data obtained through Magnetometry. Figure 28 shows the test matrix designed to encompass several scenarios involving varying sized ferrous objects being concealed at different locations in the door panel. Test Case 1 was utilized to also determine the techniques that provided the best results and only these techniques were employed for the remainder of the cases.

Before proceeding with the test matrix, it was important to ascertain that the magnetic sensor is able to provide repeatable results. Figure 33 shows the magnetic signatures as recorded by the three magnetic sensors onboard the scanning assembly with the scan direction being south to north. As can be seen, all the sensors provide repeatable results showing that the reference magnetic footprints are consistently similar.

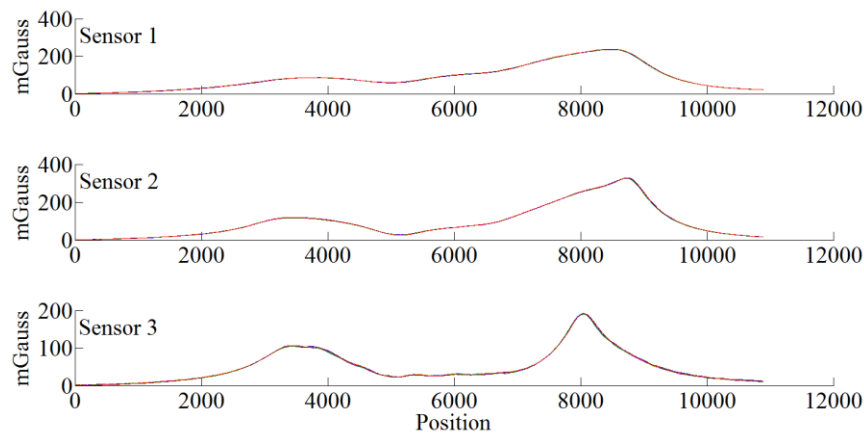


Figure 33 Magnetic signature of base door panel

### 5.1 Cross-Correlation

Test case 1 was selected to first ascertain the performance of each of the techniques used in the post processing stage. The Normalized Cross-Correlation was the first technique tested. In the upper left

corner of Figure 34, the histogram displays the distribution of the cross-correlation between case 1 in blue and case 2 in red. Case 1 is where the base magnetic signatures of the door panel were correlated with itself, and case 2 is where the base magnetic signatures were correlated with the magnetic signatures of the door panel with conceal objects. Though the distribution for the two cases has visually distinctive groups, the difference in their cross-correlation values is of the order  $1e^{-4}$ . But, as seen in Figure 34, the magnetic profiles actually show a clear mismatch at the location of the concealed object which is not as evident in the cross correlation plot. Hence the method of Normalized cross-correlation was not considered as a useful measure in positively detecting the presence of concealed objects.

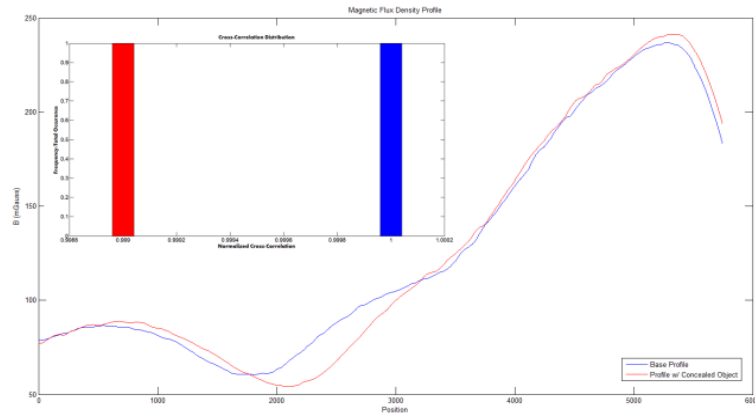


Figure 34 Normalized Cross-Correlation

## 5.2 Magnetic Field Strength and Energy Density

The idea for the second technique, Magnetic Field Strength and Energy Density was actually obtained from the previous study [6], where it provided excellent results in detecting the size of the vehicles.

Figure 35 shows the graphs of the three functions in Equation 11 that were used to capture the differential magnetic signatures resulting from the presence of concealed objects. While all three functions picked up the anomaly, function 2 in particular was able to produce the enhanced effect necessary for making a clear judgment on the presence of a concealed object.

$$|\Delta B^2| = \sum_{i=x}^z |B_c^2 - B_r^2| \rightarrow \text{Function 1}$$

$$|\Delta B|^2 = \sum_{x=i}^z |B_r - B_c|^2 \rightarrow \text{Function 2}$$

$$|\Delta B| = \sum_{x=i}^z |B_c - B_r| \rightarrow \text{Function 3}$$

Equation 11

where,

$B_r$  – reference magnetic signature

$B_c$  – test magnetic signature

$\Delta B$  – change in magnetic field strength

$\sum_{x=i}^z |f|$  - summation of function,  $f$  applied to magnetic field

components, x, y and z

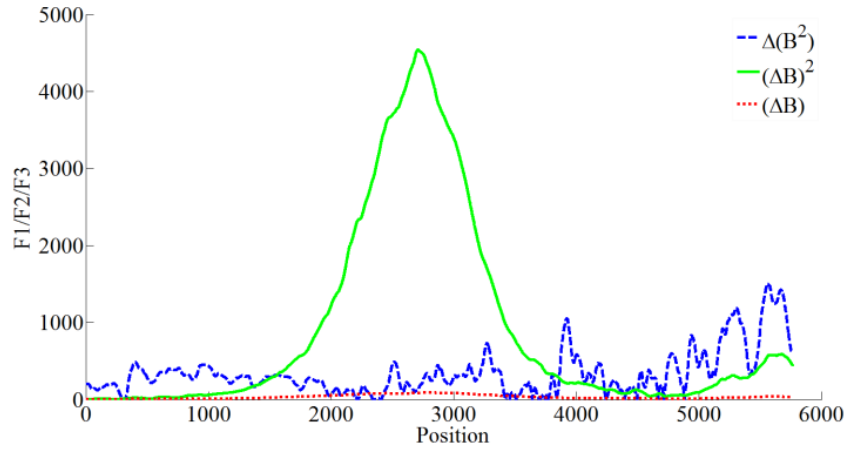


Figure 35 Magnetic Field Strength and Energy Density

Although, function 2 by itself is able to provide good results, its peculiar property of being able to significantly magnify the effect of the anomaly made it an excellent choice for usage in the Mahalanobis distance method. However, it was important to verify that this was indeed the case

and hence, all three functions were utilized as measures in the computation of the Mahalanobis distance. The results were compared to ascertain the best performing function.

### 5.3 Mahalanobis Distance

As mentioned in the Analysis chapter, Mahalanobis distance computation requires the use of two measures ( $x_1$  and  $x_2$ ), each of which being able to provide some information that helps with the detection of a concealed object. For the case at hand,  $x_1$  was chosen to be the area under the curve for the functions  $F_1$ ,  $F_2$  and  $F_3$ , and  $x_2$  was chosen to be the corresponding maximum value of these functions. The objective was to obtain two distributions, one representing the case of the door panel alone (class  $C_1$ ) and the other representing the door panel with concealed objects (class  $C_2$ ) that were well separated from each other for clear distinction. The results obtained for the three functions are as shown in Figure 36, Figure 37, and Figure 38. It clearly shows that function 1 is unable to provide the clear picture that functions 2 and 3 provides. Moreover, between functions 2 and 3, function 2 is able to produce an even greater separation of the two distributions, which corroborates the results obtained in Figure 35.

Figure 38 shows that the top and middle magnetic sensor signatures are clustered in class  $C_2$ , whereas the bottom sensor signatures is clustered along with class  $C_1$  corresponding to the base door panel configuration. This clearly shows the ability of the Mahalanobis method in localizing the placement of the concealed object in addition to detecting its presence.

The ability of function 2 to provided pronounced and unambiguous results as concluded in Table 1. Function 2 was used in the computation of Mahalanobis distance for the remainder of the test cases.



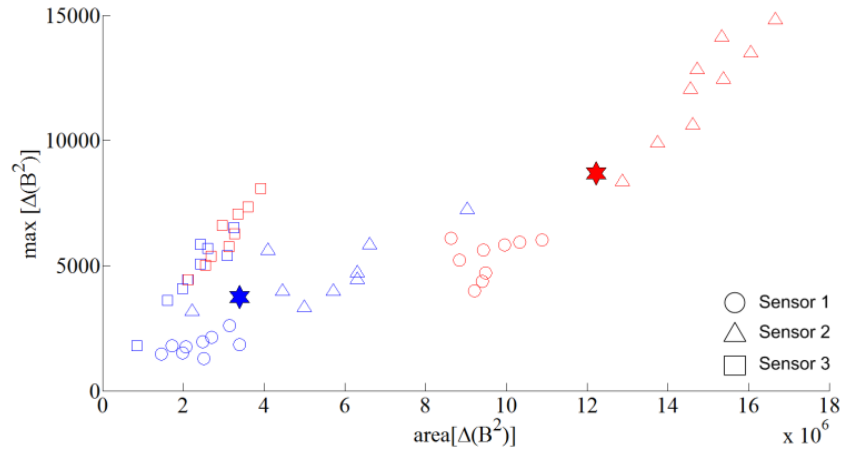


Figure 36 Mahalanobis Distribution (Function 1)

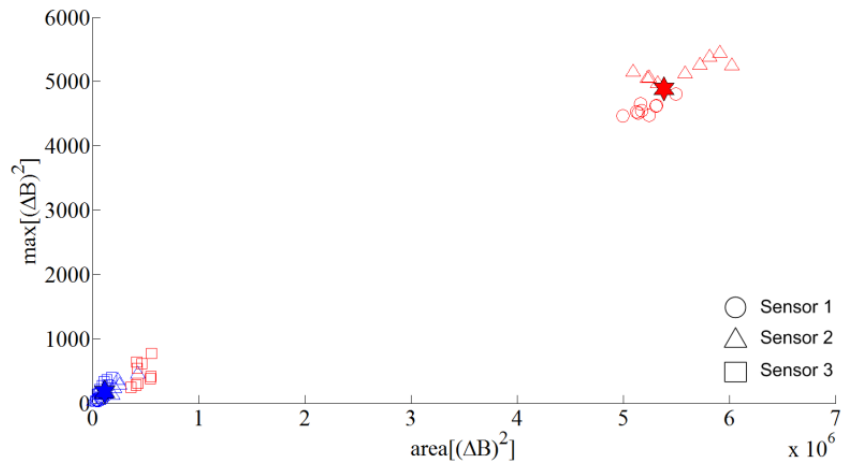


Figure 37 Mahalanobis Distribution (Function 2)

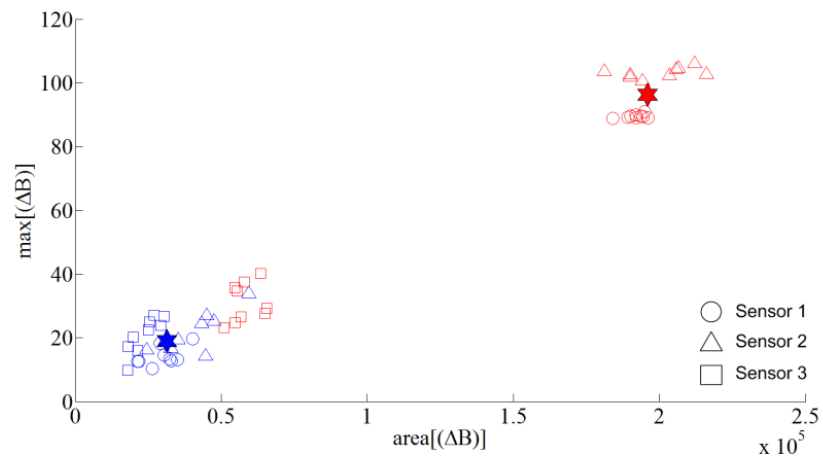


Figure 38 Mahalanobis Distribution (Function 3)

Table 1 Results of Functions 1, 2, and 3

	Conclusive in Determining Difference in Magnetic Signatures	Magnify differences in Magnetic Signatures
Function 1	No	No
Function 2	Yes	Yes
Function 3	Yes	No

In the test matrix, objects of different sizes were chosen to determine the ability of the magnetic sensors to detect small as well as large concealed objects. Figure 39 shows the results of the Mahalanobis distance applied in test cases 1, 2 and 3. Again, the Mahalanobis method is able to not only identify the different sizes of concealed objects that were used in the test cases, but is also able to localize its position within the door panel. Figure 40 shows the corresponding results for the test cases 4, 5 and 6, which are very similar.

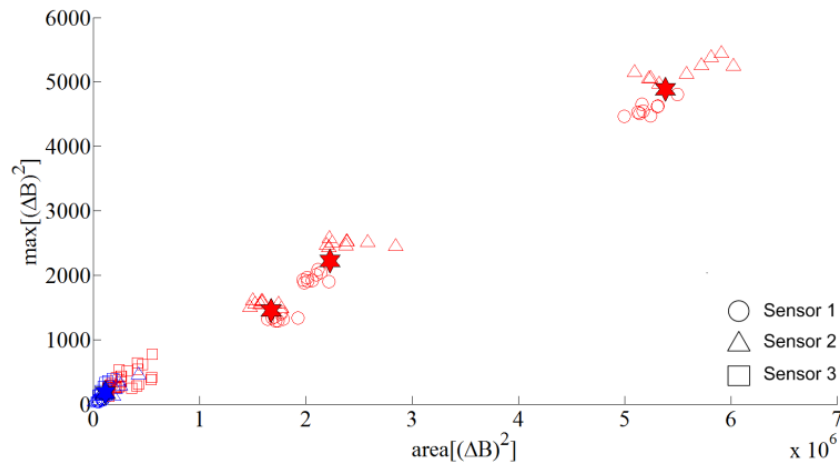


Figure 39 Mahalanobis Distribution (Test cases 1, 2 and 3)

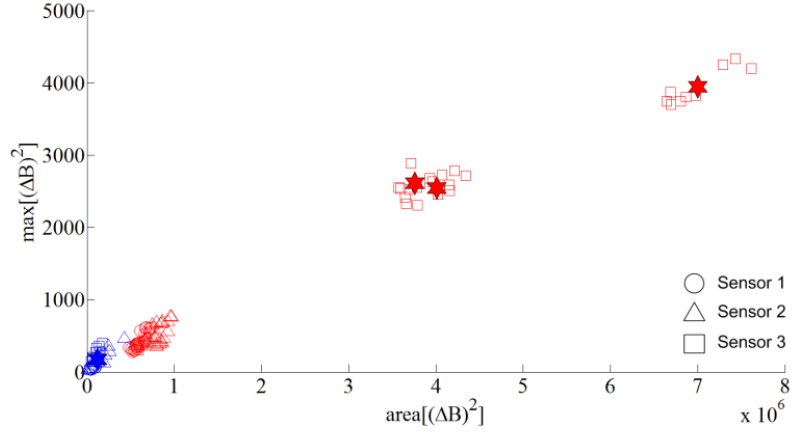


Figure 40 Mahalanobis Distribution (Test cases 4, 5 and 6)

In order to positively detect the presence of concealed objects, it's extremely important to identify the occurrence of false positives resulting from anomalies other than the concealed object. In this study, two typical anomalies, spikes and offsets were investigated. Random spikes may appear in a measurement due to electrical noise, and offsets in the measurements could be due to a bias in the calibration of the sensor that has accumulated over time. Figure 41 and Figure 42 show the results obtained for these two types of anomalies. In the case of the spikes, the sudden spike in magnitude could result in a large maximum value. The area under the spike is generally very small and hence, the observations cluster is in the top left segment of the graph, away from both classes  $C_1$  and  $C_2$ . Similarly, for the case of the offsets, since the scenario is exactly opposite to that of the spikes, the observations cluster is in the bottom right segment of the graph, away from both classes  $C_1$  and  $C_2$ . Comparing these results with the schematic of Figure 31, it can be clearly noticed that the observations for both the spikes as well as the offsets lie within the regions of the false positives. This enhances the credibility of the Mahalanobis method in accurately identifying the scenario.

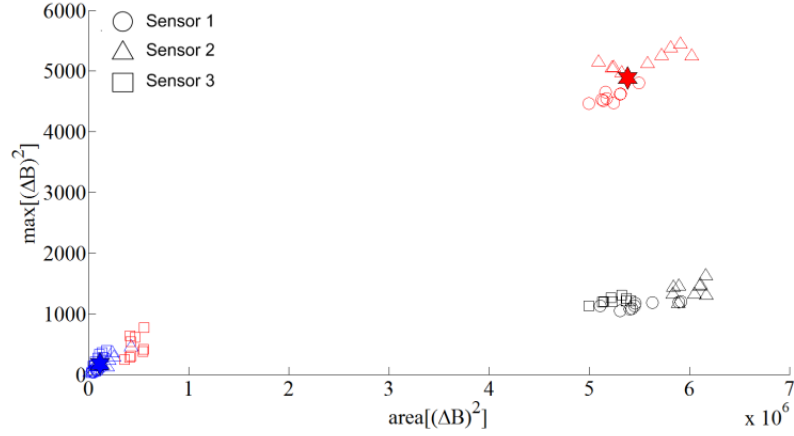


Figure 41 Mahalanobis Distribution (Offsets)

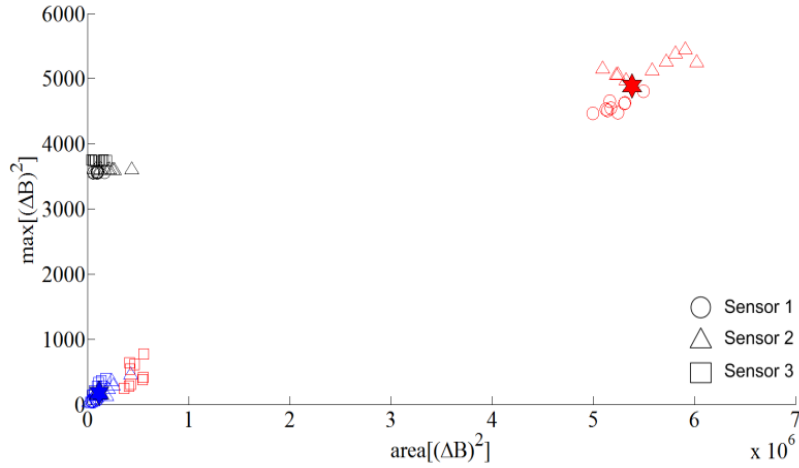


Figure 42 Mahalanobis Distribution (Spikes)

It was shown in the Analysis chapter that the Threshold Index ( $\Psi$ ) and the Limit Index ( $\phi$ ), which are based on the Mahalanobis distance have the ability to combine the results of detection and classification. Figure 43 shows the results obtained when applied to the test case 1 scenario. The observations that belong to the two classes,  $C_1$  and  $C_2$  show up in the top and bottom portions of the positive quadrant, while all the false positives end up in the negative quadrant of the Limit Index. It can be noticed that this combined approach can single handedly provide all the necessary information for positively identifying the presence of a concealed object, and at the same time eliminating the occurrence of false positives. See Table 2 for a summary of the Threshold Index

and Limit Index results for the various test cases. The Threshold and Limit Index does not only enable the automatic identification of the presence of conceal weapons, but also benefits field implementation where the security system can be operated by less advanced operator. Highly trained operators will no longer be required to interpret data produced by the security system, and less technologically inclined security operators can easily use the automatic weapon detection system.

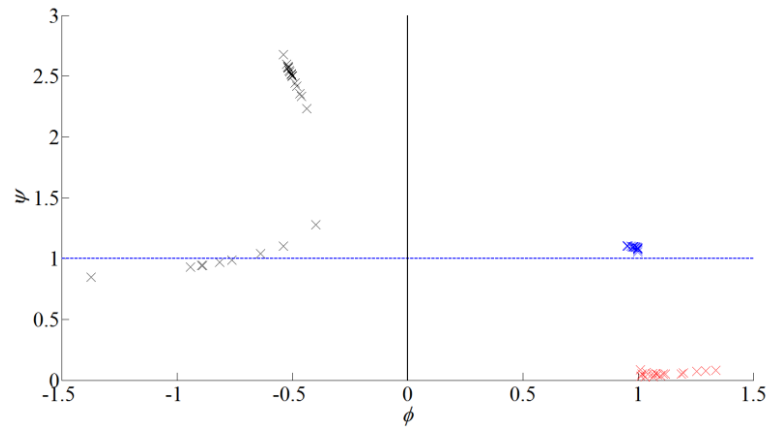


Figure 43 Threshold and Limit Index for Classification

Table 2 Threshold and Limit Index Results

	Able to Classify Correctly Using Theshold Index	Ability To Detect False Positive Correctly Using Limit Index
No Concealed Ferrous Objects	Yes	Yes
Small Ferrous Object Detected	Yes	Yes
Medium Ferrous Object Detected	Yes	Yes
Large Ferrous Object Detected	Yes	Yes
Induced Noise	No	Yes
Induced Drift in Magnetic Sensor Reading	No	Yes

## 5.4 Challenges

Though using Mahalanobis distance in conjunction with the Threshold and Limit Index proved a solid methodology in detecting concealed ferrous objects in the door panel, there were some challenges faced during the research that were related to magnetic fields. These challenges include the effects of geomagnetic field on magnetic signatures for different scanning orientation, and the detectability of the change in magnetic signature when sensor-to-door distance increases. These challenges are discussed in the next two sub-sections.

### 5.4.1 Variation in Scanning Direction

The Earth's magnetic field direction has an impact on all the previously mentioned test cases. The orientation of the door platform was aligned and maintained in the north-south and east-west direction for magnetic field data collection. By observing the raw signals in Figure 44, the classification of the base E-W profiles result in a magnetic anomaly detection.

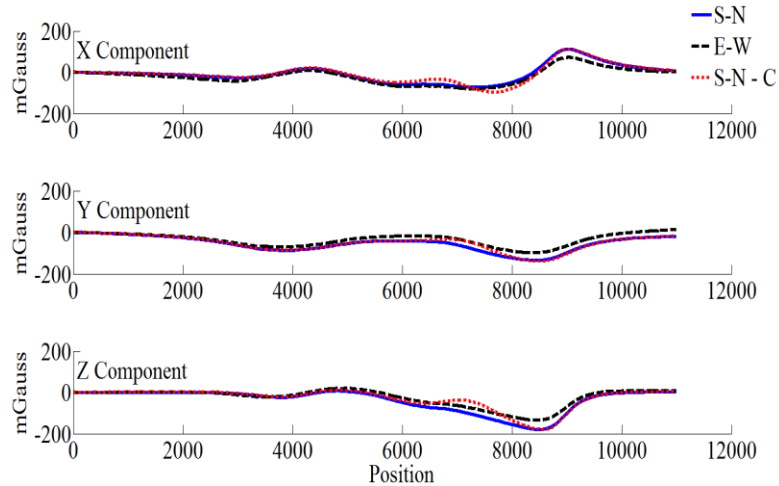


Figure 44 Magnetic Signatures for E-W direction

Also, by observing the data distribution in Figure 45, the data for sensor 1 through 3 for an E-W scanning orientation lies closer to the data that classifies a concealed object exists, hence a false

positive. From these observations, magnetic strength readings can vary based on the orientation of the scan relative to the geomagnetic field.

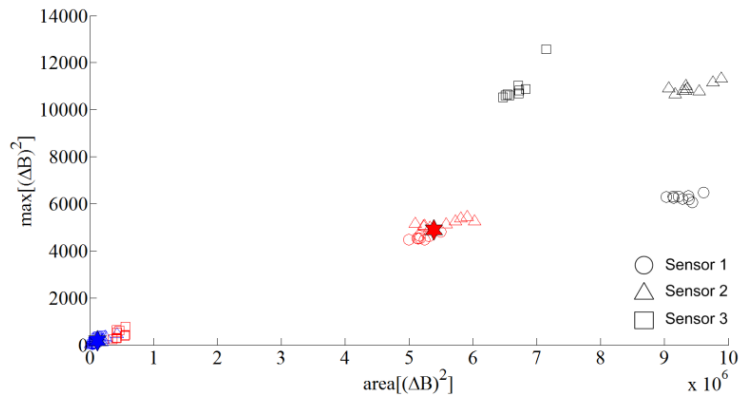


Figure 45 Mahalanobis Distribution for E-W direction

#### 5.4.2 Variation in Proximity

The strength of an induced magnetic field is known to vary as an inverse-cubic of the distance from the ferrous object to the magnetic sensor. To test this nature of the magnetic field strength and to determine the maximum allowed distance for scanning to obtain positive detection, tests were carried out where the scanning assembly was held at varying distances (1", 3" and 5") from the door panel. The resulting magnetic signatures were recorded in Figure 46 and Figure 47.

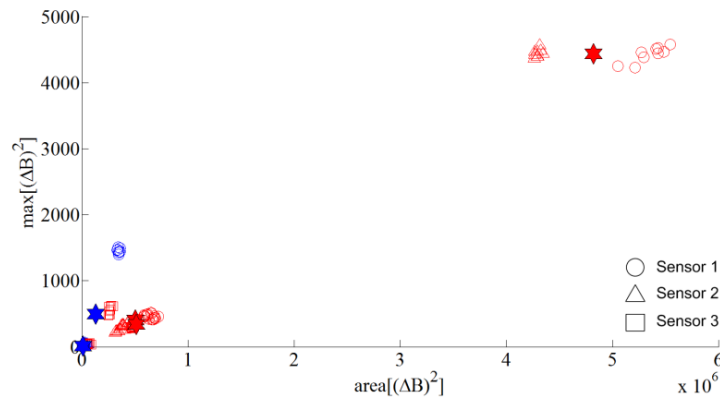


Figure 46 Mahalanobis Distribution for varying proximity

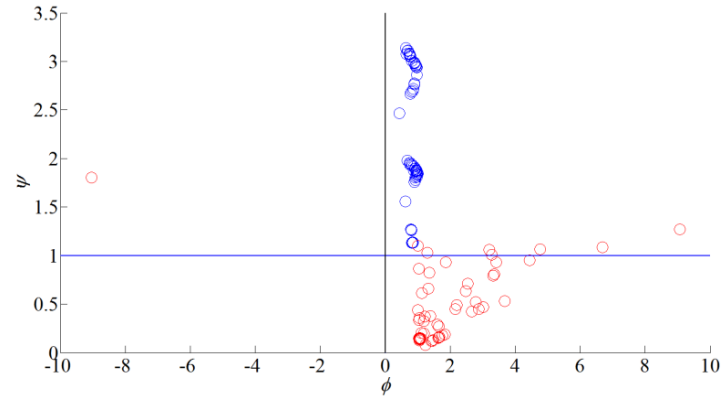


Figure 47 Threshold and Limit Index for varying proximity

While Mahalanobis distance by itself is only able to provide conclusive results for the case when the sensor is closest to the door panel, Figure 46 is able to provide a much clearer distribution between the classes. Although, some of the readings fall in the region very close to the boundary between  $C_1$  and  $C_2$ , the ability of the Threshold Index and Limit Index to enhance the results of the Mahalanobis distance is further evident.



## 6. Other Considerations and Future Work

Though results of the mechanical system and the statistical techniques proved outstanding from the test results, the following sections provide a discussion on magnetic sensors and field implementation, and possible future work using Fuzzy Logic Inference to improve concealed weapon detection.

### 6.1 Discussion On Field Implementation

Although this research project have provided the ground work on using magnetic sensors for security type applications, there are some requirements in order to implement the magnetic sensor technology in the field. Two considerations for field implementation are discussed in this section. The first consideration is that all vehicles have different magnetic signatures requiring a computer database to store the reference magnetic signatures. The second consideration is that not all weapons are made from ferrous material. Therefore, the magnetic sensor alone will not be able to detect the non-ferrous weapons since non-ferrous weapons will not distort the local magnetic field. The following two sub-sections will provide discussions on the topics.

#### 6.1.1 Vehicles and Magnetic Signatures

In this research project, one sport utility vehicle door panel was used to validate the concept of using magnetic sensors to detect concealed weapons. The questions that arises are: what happens when a different door panel is used for the testing, and most importantly, how will the system detect concealed weapon or ferrous objects for various makes and models of vehicles?

When using a door panel from another vehicle or most importantly when the system scans different vehicles, the magnetic signatures will vary from door to door or vehicle to vehicle. In order for correct magnetic signature comparison, the magnetic signatures for all years, makes and models of vehicles must be obtained for reference. The reference magnetic signatures should be stored into a database and organized based on the year, make and model of the vehicles for quick accessing of these data. To reduce operator intervention in determining the year, make, and model of the vehicle under inspection, the system should incorporate an auto-recognition license plate system that

identifies the vehicle information using the license plate number. When the auto-recognition license plate system makes out the license plate of the vehicle under inspection, the computer system will automatically access the year, make and model information through the Department of Motor Vehicle computer system. With the correct year, make, and model of the vehicle, the correct reference magnetic signature will be obtained from the database and compared to the magnetic signature under inspection.

#### 6.1.2 Non-Ferrous Weapons

One concern of using a magnetometry only detection system is that it is only able to detect ferrous materials. Although considerable amount of weapons are made with ferrous materials, there are other weapons that are made from aluminum, plastics or other non-ferrous materials. Therefore, the system will not be able to detect non-ferrous concealed weapons. By using a magnetometry-based technology, it will be able to detect the majority of weapons. But if the system developed in this research is fused with other security technology, the capability of detecting and locating conceal weapons, made from both ferrous and non-ferrous material increases.

For example, in an integrated backscatter X-ray system and a magnetometry-based weapon detection system, the information can be fused together to evaluate the vehicle under inspection. The magnetometry-based system will be used to detect ferrous weapons, while the image developed from the backscatter X-ray system will be used to verify the results of the magnetometry-based system, and located possible non-ferrous weapons. The magnetometry-based system can display on the image where the discrepancy in the magnetic signatures are located. The image produced from the backscatter X-ray system can verify the discrepancies of the magnetic signatures via the different shades that correlates with material density. Hence, the image and magnetometry information will locate the possible concealed ferrous weapon location. With the image, the automatic identification will reduce the operator's inspection time and increase vehicle throughput reducing traffic jams. Therefore the two different system serves as an overall self-check to

reduce false positive readings and increase detection rates for all weapons, both ferrous and non-ferrous material.

## 6.2 Future Work

To further investigate the use of magnetic sensor and improve classifications, the following two sub-sections will discuss the possible steps for future work. Even though the proposed next steps are not limited to only the three following sections, the author feels that they are the most logical subsequent steps. The next three section will discuss the investigation of geomagnetic field and its effects on the magnetic signatures, the investigation of using the magnetic sensors to detect concealed weapons in areas other than door panels, and the investigation of a Fuzzy Logic inferential system to help improve classification.

### 6.2.1 Geomagnetic Field Survey

From the Results chapter, the test results revealed the effects of scanning orientation on the magnetic signature readings, hence possible in creating a false positive reading. This section discusses the possible next step to study the effects of scanning orientation on magnetic signature readings.

As can be seen in Figure 44 and Figure 45, the magnetic signatures of a door panel without conceal ferrous objects changes with scanning orientation, and will produce false positive readings. As proposed for the next step, investigation of the geomagnetic field as to how it affects the magnetic signature should be pursued. The investigation is important since the reference magnetic signature may be obtained from a different scanning direction (for example, east to west) compared to an inspection scan (for example, southwest to northeast). Finding a correlation and a solution to removing the effects from different scanning orientation will help remove false positive classification.

### 6.2.2 Detection of Concealed Weapons Beyond the Door

In this research project, magnetic sensors were used to detect weapons in door panels for concept validation. Now the question is, will the magnetometry-based system developed in this research

project be able to detect concentrated ferrous objects or weapons in areas other than door panels, i.e. in the trunk or underneath the back seats. The answer to this questions is the detection is dependent upon three factors: size of the weapon or weapons, the distance of the concealed weapons or weapons to the magnetic sensors, and the number of magnetic sensors in the array use for detection.

Based on the findings of this research, as the size of the weapon or weapons increases, the easier the system will be able to find the discrepancies in the magnetic signatures, hence the detection of the concealed weapon. Similarly, if the weapon or weapons are closer to the magnetic sensors, it is easier for the system to recognize the change in the magnetic signatures. To increase detection, sensor-to-sensor distance can be reduced to increase the number of sensors. Instead of scanning along the just side of the vehicle, the sensor system can scan underneath and above the vehicle to cover all possible areas where concealed weapons can exist. With a dense magnetic sensor array, detection rate will increase even for scenarios when the weapon or weapons are concealed in areas other than the door panels. The greatest challenge is detecting a small weapon in the middle of the vehicle where the weapon is far from the sensor.

Though magnetic sensors have their limitations in scenarios when there is a large distance between the sensor and the small weapon, magnetic sensors should still be capable of detecting the weapons when there is a large number of weapons present. Magnetic sensors may not be able to detect a single hand gun under the driver seat, but they should be able to detect multiple crates of assault rifles in the back seat which is most important from a national security aspect. Even though these assumptions are based on the findings from the testing, the scenario of detecting large concealed ferrous objects in the middle of a vehicle has not been tested in the field. The next logical step is to investigate if the magnetic sensors are capable to detect various sizes of conceal ferrous objects inside different parts of the vehicle, i.e. trunk, underneath passenger seat, inside the engine compartment, and etc.

### 6.2.3 Detection Using Fuzzy Logic

In this research project, Mahalanobis distance, Threshold Index, and Limit Index were used in conjunction with the change in magnetic energy density and magnetic field strength, to classify if an observation contains a concealed ferrous object. This section will briefly discuss how the findings of this research can help build a fuzzy logic inference model to improve detection.

Fuzzy logic inference is of interest for the analysis of magnetic signatures for concealed weapon detection due to one important property of magnetic signature of ferrous objects. Ferrous objects create a unique magnetic signature as a function of the size, the shape, the ambient magnetic field, and the direction of scanning of the object. With all these varying conditions, an inferential system that makes a decision off of complex conditions is required to reduce false positive results. Through the use of fuzzy logic inference, the magnetic signatures can be evaluated based on special features, i.e. Magnetic Field Strength or Energy Density, Mahalanobis distance, correlation between segments of the signature, concavity for segments of the signature and so on. These evaluations will translate into scalar output/s, where the output/s represents a value of fuzzy logic belief if a ferrous object or weapon is concealed. The architecture of the fuzzy logic system allows the development of a subjective and complex model based on the researcher's experience without requiring a mathematical model, hence a decision making system that thinks like a human.

## 7. Conclusion

A thorough literature survey of existing scanning systems that can be used by security personnel was conducted along with the different categories of security system as defined by the National Institute of Justice. From previous studies, concentrated ferrous metals create drastic changes in the earth's magnetic footprint, hence the motivation to writing a proposal to Cal Poly's Extramural Funding Initiative to fund this research project. Through this project, a scanning system with the capability of field testing and lab testing was developed. The scanning system included a scanning mechanism, the electronics and computer system that recorded the data from the electronics.

This research introduced a magnetometry-based information and signal processing methodology for detecting concealed ferrous objects in vehicle body panels. The study utilized an extensive experimental protocol for preliminary concept validation. To detect concealed ferrous objects in vehicles, magnetic signatures of a vehicle door panel were obtained and compared to a base magnetic signature of the same vehicle door panel (the magnetic field data of the same vehicle where no foreign ferrous objects were present). To achieve the objective of accurately detecting concealed ferrous objects, a test matrix comprised of varying sized objects being concealed at different locations was identified. Simple measures such as Magnetic Field Strength and its Energy Density were computed and used in conjunction with more sophisticated statistical methods such as Normalized Cross-Correlation and Mahalanobis distance. Although all these methodologies were able to detect a magnetic footprint anomaly in the presence of a concealed object, the Mahalanobis distance approach in particular provided the most conclusive results in all the cases considered because it not only detects the presence of a concealed object, but also localize its position within the door panel. Two additional measures in the form of the Threshold Index ( $\psi$ ) and Limit Index ( $\phi$ ) were developed to automatically and accurately detect the presence of a concealed object and eliminate the occurrence of false positives.

In addition to the analysis and test results that supported the concept validation of this research project, a discussion on field implementation and possible future work were discussed in detail.

The discussion covered a possible approach to implementing magnetic signature evaluation for conceal weapon detection in the field, as well as a possible sensor fusion system to detect both ferrous and non-ferrous weapons. For possible future work, investigation of the geomagnetic field effects on magnetic signatures, investigation in detecting concealed weapons in other parts of the vehicle, and investigation of fuzzy logic inference system were also discussed as the logical next step for future work.

## REFERENCES

- [1] National Institute of Justice, "Guide to Technologies of Concealed Weapon and Contraband Imaging and Detection." 2001.
- [2] Honeywell, "HMC2003: Three Axis Magnetic Sensor Hybrid," [www.magneticsensors.com/literature.php](http://www.magneticsensors.com/literature.php).
- [3] Sharp, "GP2D120: Optoelectronic Device," [www.pololu.com/file/download/GP2D120-DATA-SHEET.pdf?file\\_id=0J157](http://www.pololu.com/file/download/GP2D120-DATA-SHEET.pdf?file_id=0J157).
- [4] National Instruments, "NI PCI-6259," <http://sine.ni.com/nips/cds/view/p/lang/en/nid/14128>.
- [5] J.E. Lenz: "A review of magnetic sensors," *Proceedings of the IEEE*, vol.78, no.6, pp.973-989, Jun 1990.
- [6] Roussel S., Porumamilla H., Birdsong C., Schuster P., "Modeling of Vehicle Magnetic Footprint in 3-D Space for Type Detection", *Proceedings of 2012 ASME International Mechanical Engineering Congress and Exposition*.
- [7] American Science and Engineering Inc., "AS&E 2011." 2011.
- [8] ProVision ATD | Advanced Imaging Technology | L3 Communications – Security and Detection Systems. (n.d.). Retrieved August 3, 2015. (<http://www.sds.l-3com.com/advancedimaging/provision-at.htm>)
- [9] Billings S., Pasion C., Walker S., Beran L., "Magnetic Models of Unexploded Ordnance," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 8, Aug 2006.
- [10] Billings S., "Discrimination and Classification of Buried Unexploded Ordnance Using Magnetometry," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 6, Jun 2004.
- [11] Holmes J., "Theoretical Development of Laboratory Techniques for Magnetic Measurement of Large Objects", *IEEE Transactions on Magnetics*, vol 37, no.5, Sept 2001.
- [12] Fischer G., Fine J., Edelstein A., "Magnetic Signatures: Small Arms Testing of Multiple Examples of Same Model Weapons," Army Research Laboratory, April 2009.
- [13] Schmidt P., Clark D., "The magnetic gradient tensor: Its properties and uses in source characterization" CSIRO Industrial Physics, Jan 2006.
- [14] Blum R., Xue Z., Lui Z. Forsyth D., "Multisensor Concealed Weapon Detection by Using A Multiresolution Mosaic Approach" IEEE, 2004.
- [15] Dimitropoulos P., Avaritsiotis J., "A 2-D ferrous object imaging technique based on magnetic field sensor arrays" *Elsevier Sensors and Actuators*, pp. 336-339, 2003.



- [16] Varshney P., Slamani M., Alford M., Ferris D., "On the Modeling of the Sensor Fusion Process for Concealed Weapon Detection," *IEEE*, 1998.
- [17] Sheinker A., Frumkis L., Ginzburg B., Salmonski N., Kaplan B., "Magnetic Anomaly Detection using a Three-Axis Magnetometer," *IEEE Transaction on Magnetics*, vol. 45, no. 1, Jan 2009.
- [18] Mori K., "Detection of Magnetic Anomaly Signal by Applying Adjustable Weight Functions," *IEEE Transaction on Magnetics*, vol. 26, no. 2, Mar 1990.
- [19] Wynn W., Frahm C., Carroll P., Clark R., Wellhoner J., Wynn M., "Advanced Superconducting Gradiometer/Magnetometer Arrays and a Novel Signal Processing Technique," *IEEE Transaction on Magnetics*, vol. 11, no. 2, Mar 1975.
- [20] McMillan R., Currie N., Ferris D., Wicks M., "Concealed Weapon Detection Using Microwave and Millimeter Wave Sensors," *IEEE*, 1988.
- [21] Kasatkin S., Polyakov O., Rusakova N., Rusakov A., "On uniqueness of solution of a reverse problem of magnetic location," *Journal of Magnetism and Magnetic Materials*, 361-364. February 2006.
- [22] Olsen R., Lyon C., "Modeling of Extremely Low Frequency Magnetic Field Sources Using Multiple Techniques," *IEEE Transactions on Power Delivery*, Vol. 11, No. 3, July 1996.
- [23] Collins L., Zhang Y., Li J., Wang H., Carin L., Hart S., Rose-Pehrsson S., Nelson H., McDonald J., "A Comparison of the Performance of Statistical and Fuzzy Algorithms for Unexpected Ordnance Detection," *IEEE Transactions on Fuzzy Systems*, Vol. 9, No. 1, February 2001.
- [24] Sheinker A., Salomonski N., Ginzburg B., Frumkis L., Kaplan B., "Magnetic anomaly detection using entropy filter," *Measurement Science and Technology*, 19, February 2008.
- [25] McFee J., Das Y., Ellingson R., "Locating and Identifying Compact Ferrous Objects," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 28, No. 2, March 1990.
- [26] Agurto A., Li Y., Tian G., Bowring N., Lockwood S., "A Review of Conealed Weapon Detection and Research in Perspective," *Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control*, 2007.
- [27] Costianes P., "An Overview of Concealed Weapon Detection for Homeland Security," *Proceedings of the 34<sup>th</sup> Applied Imagery and Pattern Recognition Workshop*, April 2005.
- [28] Robinson S., Smith E., Jarman K., Runkle R., Ashbaker E., Jordan D., Kaye William., Warren G., "An Overview of Concealed Weapon Detection for Homeland Security," *IEEE Nuclear Science Symposium Conference Record*, 2006.
- [29] Sutton V., Bromley A., "Understanding technologies of terror," *Elsevier Technology in Society* 27, pp. 263-285, 2005.

- [30] Allen R., Mills D., "Signal Analysis: Time, Frequency, Scale and Structure," *Wiley-Interscience*, 2004.
- [31] Golubtsov P., Moskaliuk S., "Fuzzy Logic, Informativeness and Bayesian Decision-Making Problems," 2006.
- [32] Mendel J., "Fuzzy Logic Systems for Engineer: A Tutorial," *Proceedings of the IEEE*, vol.83, no.3, 1995.

## APPENDICES

### Appendix A. Sensor Evaluation and Selection

The purpose of sensor evaluation is to determine a magnetic sensor and distance sensing sensor that best fits the design criteria of the research system. The selection criteria for the sensors are listed as follows:

#### Sensor Criteria:

1. Small foot print in order to be mounted in an enclosure
2. Light weight in order to reduce impact on motion system
3. Powered by 5-24 VDC
4. Magnetic sensors needs to have highest possible sensitivity as it is available as an off the shelf product
5. Distance sensor must be able to detect as low as one inch and up to one foot
6. Magnetic sensor must be able to detect the small changes in earth's geomagnetic field (range 0.25 – 0.6 gauss)

Through vigorous research, two magnetic sensors and two distance sensing sensors were found to be possible candidates for the sensor system. The two magnetic sensors were sampled for field testing were the FLC-100 from Stefan Mayer Instruments and HMC-2003 from Honeywell. The two distance sensing sensors that were sampled for field testing were the ultrasonic sensor LV-MaxSonar-EZ4 from MaxBotix and the infrared distance sensor GP2D120 from Sharp. The details regarding the specifications and selection process will be found in the next section.

#### Magnetic Sensor Evaluation and Selection

##### FLC-100 Magnetometer Overview

The FLC-100 is a single axis fluxgate magnetic sensor from Stefan Mayer Instruments with measuring range up of +/- 100 uT or +/- 1 gauss and resolution of 3000 ugauss. The sensor comes in a small foot print of 1.75" by 0.4" circuit board layout with a thickness of roughly 0.25". The sensor comes with soldering pads that are positioned in a standard grid of 0.1", ready to be integrated onto circuit boards or just direct wiring.

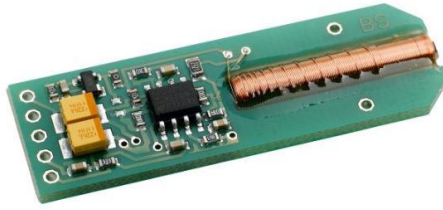


Figure 12 FLC-100 Single Axis Fluxgate Magnetic Sensor

(Courtesy of Stefan Mayer Instruments)

#### HMC-2003 Magnetometer Overview

The HMC-2003 is a three axis magneto-resistive sensor from Honeywell with measuring range up of  $\pm 2$  gauss and resolution of 40 ugauss. The sensor comes in a small 20 pin DIP footprint of 1" by 0.75" with thickness of 0.5" and ready for field application.

#### Magnetic Sensor Comparison and Selection

Table 3 Magnetic Sensor Comparison

	FLC-100	HMC-2003
Range	$\pm 100$ uT ( $\pm 1$ gauss)	$\pm 2$ gauss
Resolution	3000 ugauss	40 ugauss
Sensitivity	1 V/50 uT (2 V/gauss)	1 V/gauss
Supply Voltage	5V	6-15 VDC
Foot Print	1.75" x 0.4" x 0.25"	1" x 0.75" x 0.5"
Sensing Axis	1	3

Through comparison, the Honeywell HMC-2003 was selected because the sensor will provide more benefits to the project i.e. larger detectability range, three axis measurement and small foot print.

Though the FLC-100 has a higher sensitivity range (voltage output per magnetic flux density), the sensor itself has a broader resolution, meaning the sensor is not as sensitivity as the HMC-2003.

The detailed reasons why the HMC-2003 was selected can be seen below.

#### HMC-2003 Selection

1. HMC-2003 has twice the detectability range of the Stefan Mayer Instrument FLC-100.

2. HMC-2003 will require the less foot print as opposed to the FLC-100. The FLC-100 will require three times its foot print to provide the same information as the HMC-2003 since the FLC-100 is only a single axis magnetic sensor.
3. HMC-2003 is a three axis magnetic sensor that will provide a complete magnetic field reading
4. HMC-2003 sensing element is more sensitive (more resolution)

#### Distance Sensor Evaluation and Selection

##### LV-MaxSonar-EZ4 Ultrasonic Sensor Overview

The LV-MaxSonar-EZ4 from MaxBotix is an ultrasonic distance measuring sensor that uses a narrow beam for short distance detection. The ultrasonic sensor comes on a 0.785" by 0.870" circuit board foot print with a thickness of 0.645". Distance information can be output to the end user via analog output or RS-232. The EZ4 requires very low power with voltage supply from 2.5 to 5.5 volt and typically draws 2mA.

##### GP2D120 Infrared Sensor Overview

The GP2D120 from Sharp is an infrared sensor for short distance measuring. The infrared sensor comes in a 1.75" by 0.75" foot print with a thickness of 0.54". Distance information are output via analog terminal. The GP2D120 requires a 4.5 to 5.0 Volt power supply and typically draws 33mA.

#### Distance Sensor Comparison and Selection

Table 4 Distance Sensor Comparison

	LV-MaxSonar-EZ4	GP2D120
Range	6 to 254 inches	1.5 to 12 inches
Resolution	1 inch	unknown
Supply Voltage	2.5 to 5.5 V	4.5 to 5.5 V
Current Draw	2 mA	33 mA
Foot Print	0.785" x 0.870" x 0.645"	1.75" x 0.75" x 0.54"

Through lab tests, the infrared GP2D120 Sharp distance measuring sensor was selected due to the lower distance range that can be detected. Being able to detect the lower range is very important since the scanning system is designed for close range.

## Appendix B. HMC2003 Set/Reset Circuit

The HMC2003 sensors are made up of three anisotropic magneto-resistive sensor that was fabricated with permalloy (NiFe) thin films which the material changes in electrical resistance when an external field is applied. Strong magnetic fields can disrupt the magnetic domains of the film particle from the factory smooth to arbitrary orientation. See Figure 48. The accuracy and resolution will suffer if the Set/Reset of the magnetic domains are not performed.

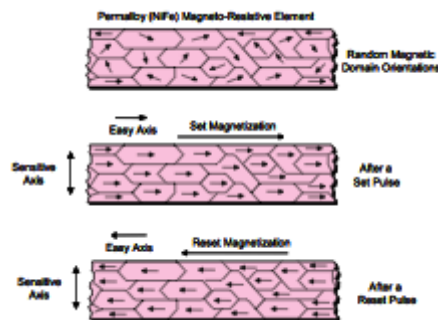


Figure 48 Orientation of AMR of the magnetic sensor

Photo courtesy of Honeywell

The reason to perform a Set/Reset on an AMR sensors are to recover from strong external magnetic field that have likely to magnetized the magnetic films of the sensor and to optimized the magnetic domains for most sensitive performance [1]. See [1] for more information on Honeywell magnetic sensors and the details of the Set/Reset circuit.

The Set/Reset circuit used to re-orient the magnetic film of the HMC2003 was obtained from [2] and can be seen in Figure 49. The parts list for the actual build of the circuit can be reviewed in Table 5. Item 3, 9 and 10 were not used in the circuit assembly in the first phase of the circuit build.

Table 5 Part Lists for Set/Reset Circuit Assembly

Item	DigiKey Part Number	Description
1	1014-1044-ND	MOSFET N/P-CH 13.2V 8PDIP
2	MC14504BCPGOS-ND	IC TRANSLATOR HEX CMOS 16DIP
3	1N5347BGOS-ND	DIODE ZENER 10V 5W AXIAL
4	497-1367-5-ND	IC HEX INVERTER 14 DIP
5	399-3530-ND	CAP TANT 0.15UF 35V 10% RADIAL
6	445-2881-ND	CAP CER .10UF 16V RADIAL
7	TR50JBC100R-ND	RES 100 OHM 50W 5% TO-220
8	LT1084CT#PBF-ND	LT Linear Voltage Regulator
9	309-1098-ND	8-SOIC to 8-DIP Adapter
10	IRF7105PBFCT-ND	N and P MOSFETS
11	N/A	47 Ohm Resistor
12	N/A	330 Ohm Resistor

Item 9 and 10 were purchased for the contingency plan if the MOSFET (Item 1) was not able to handle the amount of current drawn.

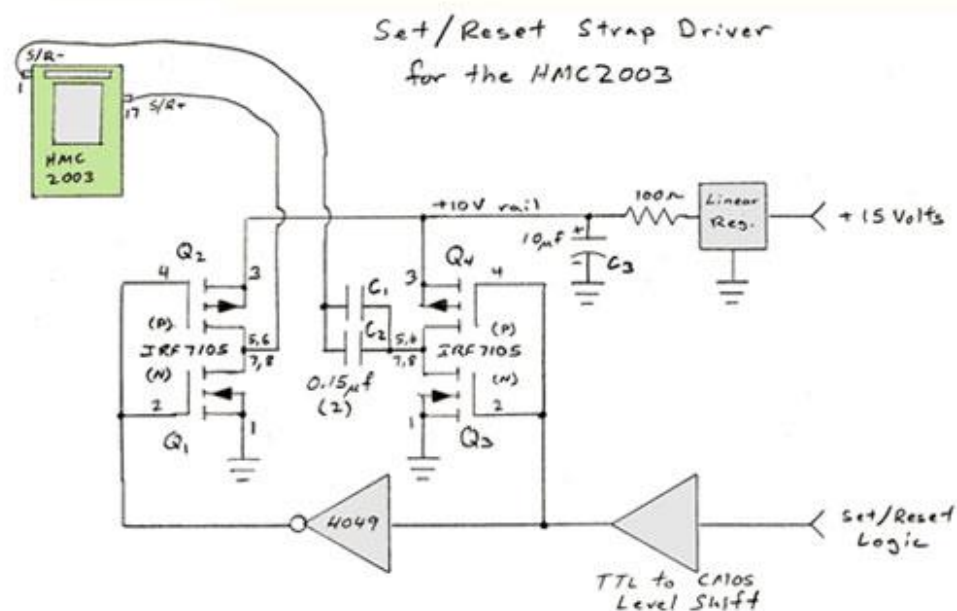


Figure 49 HMC2003 Set/Reset Strap Driver

Photo courtesy of Honeywell

The circuit from Figure 49 was used as a template in building the Set/Reset circuit. The actual circuit was built on a prototype bread board prototype before incorporating into the circuit design for the HMC-2003. See Figure 50 to see the prototype of the Set/Reset circuit.

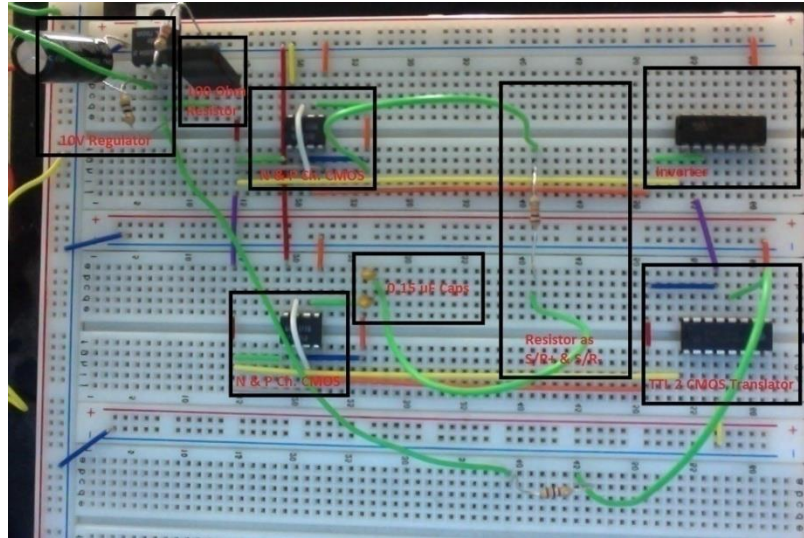


Figure 50 HMC2003 Prototype Set/Reset Circuit

#### References

- [1] Honeywell, "Set/Reset Function for Magnetic Sensors," [www.magneticsensors.com](http://www.magneticsensors.com).
- [2] Honeywell, "Set/Reset Strap Driver - HMC2003," <http://www.magneticsensors.com/magnetics-application-schematics.php>.



## Appendix C. Computer Build for Computer Workstation

Table 6 Software Installed

	Newegg Item #
Operating System	Windows XP SP2
Programming Software	LabView
	MatLab and Simulink

Table 7 Computer Parts List

Computer Part	Manufacturer	Model	Newegg Item #
Processor	Intel	Intel Core i5-2400 Sandy Bridge	N82E16819115074
Motherboard	ASUS	Intel P8H67-M PRO/CSM	N82E16813131711
Primary SSD	Plextor	PX-M2 Series PX-128M2S	N82E16820249010
2.5" to 3.5" Adapter	ICY DOCK	BM990SP-B Dual 2.5 to 3.5 SSD Bracket	N82E16817994087
Memory	G.SKILL	Ripjaws Series 4 GB (2x2GB) 240-Pin DDR3 1333	N82E16820231276
Tower	APEVIA	X-QPACK2-BK/500 Black Aluminum Body	N82E16811144140
Power Supply	ENERMAX	NAXN ENP550AWT 550W ATX	N82E16817194089

## Appendix D. LabView Program for PCI-NI 6259

A program was developed in LabView to acquire data from the magnetic sensors and infrared sensors. The program can be seen in the below Figure 51. There are four main sections of the data acquisition program: channel configuration (section A), file configuration (section B), data read and write (section C), and end task (section D).

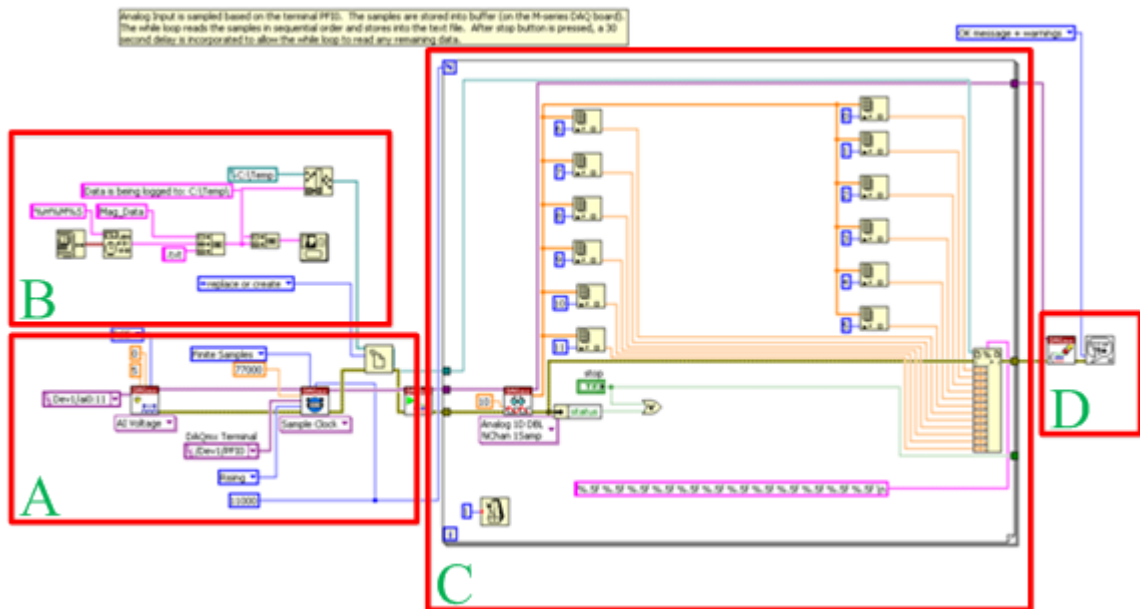


Figure 51 LabView Data Acquisition Program

### Channel Configuration

In this section A, the 12 analog channels were configured for a voltage range of 0-5 volts, which is appropriate since the magnetic sensors and infrared sensors outputs within that range. The data acquisition board is set up to acquire 11,000 samples spanning the length of the scanning assembly using the channel A of optical encoder for the sample clock.

### File Configuration

Section B configures the text file for the data to be written to. The file is automatically created with the prefix, "Mag\_Data" followed by the hour, minute, and second when the data acquisition session was activated. The file is saved under the directory "C:\Temp\".

### Data Read/Write and End Task

Section B and C writes the data collected at each rising edge of the encoder into the text file. The file is formatted such that each row correlates to each tick of channel A, therefore having a spatially uniform data acquisition. When the 11,000 samples have been collected, the data acquisition session is terminated and cleared by the end task section of the program.

## Appendix E. MatLab Program

### MatLab Main Program

```
% This script was written to compare 10 base profiles of the test door.
addpath('C:\Users\Public\Documents\Magnetic Sensing Research\Magnetic Sensing Test\Filtered
Test Data 5-17 and 5-19\Functions');
%
clear all
clc
data = load('Filtered Base 1-inch Scans.mat');
datac = load('Filtered 2-2 Orientation 4x 1-inch Scans.mat');
segment = 'top'; %top for top segment, bottom for bottom segment
DO = 0; %1 To graph base profile for different scanning orientation
addpath('C:\Users\Public\Documents\Magnetic Sensing Research\Magnetic Sensing Test\Filtered
Test Data 5-17 and 5-19\Test 3 - Variation in Scanning Directions - Base Profiles')
do = load('Filtered E-W Base 1-inch.mat');
% %
%loading x1 base profiles segments based on door detection
ir_s1_1 = bfilter(data.i1_filter_mv,0.009);
[i1_1] = edge_detector(ir_s1_1);
clear ir_s1_1
x1_base1 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),1) -
remove_offset(data.x1_filter(:,1))),0.009);
y1_base1 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),1) -
remove_offset(data.y1_filter(:,1))),0.009);
z1_base1 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),1) -
remove_offset(data.z1_filter(:,1))),0.009);

x1_base2 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),2) -
remove_offset(data.x1_filter(:,2))),0.009);
y1_base2 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),2) -
remove_offset(data.y1_filter(:,2))),0.009);
z1_base2 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),2) -
remove_offset(data.z1_filter(:,2))),0.009);

x1_base3 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),3) -
remove_offset(data.x1_filter(:,3))),0.009);
y1_base3 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),3) -
remove_offset(data.y1_filter(:,3))),0.009);
z1_base3 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),3) -
remove_offset(data.z1_filter(:,3))),0.009);

x1_base4 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),4) -
remove_offset(data.x1_filter(:,4))),0.009);
y1_base4 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),4) -
remove_offset(data.y1_filter(:,4))),0.009);
z1_base4 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),4) -
remove_offset(data.z1_filter(:,4))),0.009);
```

```

x1_base5 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),5) -
remove_offset(data.x1_filter(:,5))),0.009);
y1_base5 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),5) -
remove_offset(data.y1_filter(:,5))),0.009);
z1_base5 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),5) -
remove_offset(data.z1_filter(:,5))),0.009);

x1_base6 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),6) -
remove_offset(data.x1_filter(:,6))),0.009);
y1_base6 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),6) -
remove_offset(data.y1_filter(:,6))),0.009);
z1_base6 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),6) -
remove_offset(data.z1_filter(:,6))),0.009);

x1_base7 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),7) -
remove_offset(data.x1_filter(:,7))),0.009);
y1_base7 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),7) -
remove_offset(data.y1_filter(:,7))),0.009);
z1_base7 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),7) -
remove_offset(data.z1_filter(:,7))),0.009);

x1_base8 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),8) -
remove_offset(data.x1_filter(:,8))),0.009);
y1_base8 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),8) -
remove_offset(data.y1_filter(:,8))),0.009);
z1_base8 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),8) -
remove_offset(data.z1_filter(:,8))),0.009);

x1_base9 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),9) -
remove_offset(data.x1_filter(:,9))),0.009);
y1_base9 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),9) -
remove_offset(data.y1_filter(:,9))),0.009);
z1_base9 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),9) -
remove_offset(data.z1_filter(:,9))),0.009);

x1_base10 = bfilter(1000*(data.x1_filter(i1_1(1):i1_1(2),10) -
remove_offset(data.x1_filter(:,10))),0.009);
y1_base10 = bfilter(1000*(data.y1_filter(i1_1(1):i1_1(2),10) -
remove_offset(data.y1_filter(:,10))),0.009);
z1_base10 = bfilter(1000*(data.z1_filter(i1_1(1):i1_1(2),10) -
remove_offset(data.z1_filter(:,10))),0.009);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor two
%loading x1 base pr2ofiles segments based on door2 detection
ir_s2_1 = bfilter(data.i2_filter_mv,0.009);
[i2_1] = edge_detector(ir_s2_1);
clear ir_s2_1
x2_base1 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),1) -
remove_offset(data.x2_filter(:,1))),0.009);

```

```

y2_base1 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),1) -
remove_offset(data.y2_filter(:,1))),0.009);
z2_base1 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),1) -
remove_offset(data.z2_filter(:,1))),0.009);

x2_base2 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),2) -
remove_offset(data.x2_filter(:,2))),0.009);
y2_base2 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),2) -
remove_offset(data.y2_filter(:,2))),0.009);
z2_base2 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),2) -
remove_offset(data.z2_filter(:,2))),0.009);

x2_base3 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),3) -
remove_offset(data.x2_filter(:,3))),0.009);
y2_base3 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),3) -
remove_offset(data.y2_filter(:,3))),0.009);
z2_base3 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),3) -
remove_offset(data.z2_filter(:,3))),0.009);

x2_base4 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),4) -
remove_offset(data.x2_filter(:,4))),0.009);
y2_base4 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),4) -
remove_offset(data.y2_filter(:,4))),0.009);
z2_base4 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),4) -
remove_offset(data.z2_filter(:,4))),0.009);

x2_base5 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),5) -
remove_offset(data.x2_filter(:,5))),0.009);
y2_base5 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),5) -
remove_offset(data.y2_filter(:,5))),0.009);
z2_base5 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),5) -
remove_offset(data.z2_filter(:,5))),0.009);

x2_base6 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),6) -
remove_offset(data.x2_filter(:,6))),0.009);
y2_base6 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),6) -
remove_offset(data.y2_filter(:,6))),0.009);
z2_base6 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),6) -
remove_offset(data.z2_filter(:,6))),0.009);

x2_base7 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),7) -
remove_offset(data.x2_filter(:,7))),0.009);
y2_base7 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),7) -
remove_offset(data.y2_filter(:,7))),0.009);
z2_base7 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),7) -
remove_offset(data.z2_filter(:,7))),0.009);

x2_base8 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),8) -
remove_offset(data.x2_filter(:,8))),0.009);
y2_base8 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),8) -
remove_offset(data.y2_filter(:,8))),0.009);

```

```

z2_base8 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),8) -
remove_offset(data.z2_filter(:,8))),0.009);

x2_base9 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),9) -
remove_offset(data.x2_filter(:,9))),0.009);
y2_base9 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),9) -
remove_offset(data.y2_filter(:,9))),0.009);
z2_base9 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),9) -
remove_offset(data.z2_filter(:,9))),0.009);

x2_base10 = bfilter(1000*(data.x2_filter(i2_1(1):i2_1(2),10) -
remove_offset(data.x2_filter(:,10))),0.009);
y2_base10 = bfilter(1000*(data.y2_filter(i2_1(1):i2_1(2),10) -
remove_offset(data.y2_filter(:,10))),0.009);
z2_base10 = bfilter(1000*(data.z2_filter(i2_1(1):i2_1(2),10) -
remove_offset(data.z2_filter(:,10))),0.009);
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%Sensor Three
%loading x1 base pr2ofiles segments based on door2 detection
ir_s3_1 = bfilter(data.i3_filter_mv(:,1),0.009);
[i3_1] = edge_detector(ir_s3_1);
%clear ir_s3_1
x3_base1 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),1) -
remove_offset(data.x3_filter(:,1))),0.009);
y3_base1 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),1) -
remove_offset(data.y3_filter(:,1))),0.009);
z3_base1 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),1) -
remove_offset(data.z3_filter(:,1))),0.009);

x3_base2 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),2) -
remove_offset(data.x3_filter(:,2))),0.009);
y3_base2 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),2) -
remove_offset(data.y3_filter(:,2))),0.009);
z3_base2 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),2) -
remove_offset(data.z3_filter(:,2))),0.009);

x3_base3 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),3) -
remove_offset(data.x3_filter(:,3))),0.009);
y3_base3 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),3) -
remove_offset(data.y3_filter(:,3))),0.009);
z3_base3 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),3) -
remove_offset(data.z3_filter(:,3))),0.009);

x3_base4 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),4) -
remove_offset(data.x3_filter(:,4))),0.009);
y3_base4 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),4) -
remove_offset(data.y3_filter(:,4))),0.009);
z3_base4 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),4) -
remove_offset(data.z3_filter(:,4))),0.009);

```

```

x3_base5 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),5) -
remove_offset(data.x3_filter(:,5))),0.009);
y3_base5 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),5) -
remove_offset(data.y3_filter(:,5))),0.009);
z3_base5 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),5) -
remove_offset(data.z3_filter(:,5))),0.009);

x3_base6 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),6) -
remove_offset(data.x3_filter(:,6))),0.009);
y3_base6 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),6) -
remove_offset(data.y3_filter(:,6))),0.009);
z3_base6 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),6) -
remove_offset(data.z3_filter(:,6))),0.009);

x3_base7 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),7) -
remove_offset(data.x3_filter(:,7))),0.009);
y3_base7 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),7) -
remove_offset(data.y3_filter(:,7))),0.009);
z3_base7 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),7) -
remove_offset(data.z3_filter(:,7))),0.009);

x3_base8 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),8) -
remove_offset(data.x3_filter(:,8))),0.009);
y3_base8 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),8) -
remove_offset(data.y3_filter(:,8))),0.009);
z3_base8 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),8) -
remove_offset(data.z3_filter(:,8))),0.009);

x3_base9 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),9) -
remove_offset(data.x3_filter(:,9))),0.009);
y3_base9 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),9) -
remove_offset(data.y3_filter(:,9))),0.009);
z3_base9 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),9) -
remove_offset(data.z3_filter(:,9))),0.009);

x3_base10 = bfilter(1000*(data.x3_filter(i3_1(1):i3_1(2),10) -
remove_offset(data.x3_filter(:,10))),0.009);
y3_base10 = bfilter(1000*(data.y3_filter(i3_1(1):i3_1(2),10) -
remove_offset(data.y3_filter(:,10))),0.009);
z3_base10 = bfilter(1000*(data.z3_filter(i3_1(1):i3_1(2),10) -
remove_offset(data.z3_filter(:,10))),0.009);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sensor 1
%loading profiles with conceal objects
ir_s1_1c = bfilter(datac.i1_filter_mv,0.009);
[i1_1c] = edge_detector(ir_s1_1c);
clear ir_s1_1c
x1_c1 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),1) -
remove_offset(datac.x1_filter(:,1))),0.009);

```



```

y1_c1 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),1) -
remove_offset(datac.y1_filter(:,1))),0.009);
z1_c1 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),1) -
remove_offset(datac.z1_filter(:,1))),0.009);

x1_c2 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),2) -
remove_offset(datac.x1_filter(:,2))),0.009);
y1_c2 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),2) -
remove_offset(datac.y1_filter(:,2))),0.009);
z1_c2 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),2) -
remove_offset(datac.z1_filter(:,2))),0.009);

x1_c3 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),3) -
remove_offset(datac.x1_filter(:,3))),0.009);
y1_c3 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),3) -
remove_offset(datac.y1_filter(:,3))),0.009);
z1_c3 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),3) -
remove_offset(datac.z1_filter(:,3))),0.009);

x1_c4 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),4) -
remove_offset(datac.x1_filter(:,4))),0.009);
y1_c4 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),4) -
remove_offset(datac.y1_filter(:,4))),0.009);
z1_c4 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),4) -
remove_offset(datac.z1_filter(:,4))),0.009);

x1_c5 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),5) -
remove_offset(datac.x1_filter(:,5))),0.009);
y1_c5 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),5) -
remove_offset(datac.y1_filter(:,5))),0.009);
z1_c5 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),5) -
remove_offset(datac.z1_filter(:,5))),0.009);

x1_c6 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),6) -
remove_offset(datac.x1_filter(:,6))),0.009);
y1_c6 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),6) -
remove_offset(datac.y1_filter(:,6))),0.009);
z1_c6 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),6) -
remove_offset(datac.z1_filter(:,6))),0.009);

x1_c7 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),7) -
remove_offset(datac.x1_filter(:,7))),0.009);
y1_c7 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),7) -
remove_offset(datac.y1_filter(:,7))),0.009);
z1_c7 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),7) -
remove_offset(datac.z1_filter(:,7))),0.009);

x1_c8 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),8) -
remove_offset(datac.x1_filter(:,8))),0.009);
y1_c8 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),8) -
remove_offset(datac.y1_filter(:,8))),0.009);

```

```

z1_c8 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),8) -
remove_offset(datac.z1_filter(:,8))),0.009);

x1_c9 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),9) -
remove_offset(datac.x1_filter(:,9))),0.009);
y1_c9 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),9) -
remove_offset(datac.y1_filter(:,9))),0.009);
z1_c9 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),9) -
remove_offset(datac.z1_filter(:,9))),0.009);

x1_c10 = bfilter(1000*(datac.x1_filter(i1_1c(1):i1_1c(2),10) -
remove_offset(datac.x1_filter(:,10))),0.009);
y1_c10 = bfilter(1000*(datac.y1_filter(i1_1c(1):i1_1c(2),10) -
remove_offset(datac.y1_filter(:,10))),0.009);
z1_c10 = bfilter(1000*(datac.z1_filter(i1_1c(1):i1_1c(2),10) -
remove_offset(datac.z1_filter(:,10))),0.009);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 2
ir_s2_1c = bfilter(datac.i2_filter_mv,0.009);
[i2_1c] = edge_detector(ir_s2_1c);
clear ir_s2_1c
x2_c1 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),1) -
remove_offset(datac.x2_filter(:,1))),0.009);
y2_c1 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),1) -
remove_offset(datac.y2_filter(:,1))),0.009);
z2_c1 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),1) -
remove_offset(datac.z2_filter(:,1))),0.009);

x2_c2 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),2) -
remove_offset(datac.x2_filter(:,2))),0.009);
y2_c2 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),2) -
remove_offset(datac.y2_filter(:,2))),0.009);
z2_c2 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),2) -
remove_offset(datac.z2_filter(:,2))),0.009);

x2_c3 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),3) -
remove_offset(datac.x2_filter(:,3))),0.009);
y2_c3 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),3) -
remove_offset(datac.y2_filter(:,3))),0.009);
z2_c3 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),3) -
remove_offset(datac.z2_filter(:,3))),0.009);

x2_c4 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),4) -
remove_offset(datac.x2_filter(:,4))),0.009);
y2_c4 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),4) -
remove_offset(datac.y2_filter(:,4))),0.009);
z2_c4 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),4) -
remove_offset(datac.z2_filter(:,4))),0.009);

```

```

x2_c5 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),5) -
remove_offset(datac.x2_filter(:,5))),0.009);
y2_c5 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),5) -
remove_offset(datac.y2_filter(:,5))),0.009);
z2_c5 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),5) -
remove_offset(datac.z2_filter(:,5))),0.009);

x2_c6 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),6) -
remove_offset(datac.x2_filter(:,6))),0.009);
y2_c6 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),6) -
remove_offset(datac.y2_filter(:,6))),0.009);
z2_c6 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),6) -
remove_offset(datac.z2_filter(:,6))),0.009);

x2_c7 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),7) -
remove_offset(datac.x2_filter(:,7))),0.009);
y2_c7 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),7) -
remove_offset(datac.y2_filter(:,7))),0.009);
z2_c7 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),7) -
remove_offset(datac.z2_filter(:,7))),0.009);

x2_c8 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),8) -
remove_offset(datac.x2_filter(:,8))),0.009);
y2_c8 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),8) -
remove_offset(datac.y2_filter(:,8))),0.009);
z2_c8 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),8) -
remove_offset(datac.z2_filter(:,8))),0.009);

x2_c9 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),9) -
remove_offset(datac.x2_filter(:,9))),0.009);
y2_c9 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),9) -
remove_offset(datac.y2_filter(:,9))),0.009);
z2_c9 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),9) -
remove_offset(datac.z2_filter(:,9))),0.009);

x2_c10 = bfilter(1000*(datac.x2_filter(i2_1c(1):i1_1c(2),10) -
remove_offset(datac.x2_filter(:,10))),0.009);
y2_c10 = bfilter(1000*(datac.y2_filter(i2_1c(1):i1_1c(2),10) -
remove_offset(datac.y2_filter(:,10))),0.009);
z2_c10 = bfilter(1000*(datac.z2_filter(i2_1c(1):i1_1c(2),10) -
remove_offset(datac.z2_filter(:,10))),0.009);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%Sensor 3

```

```

ir_s3_1c = bfilter(datac.i3_filter_mv,0.009);
[i3_1c] = edge_detector(ir_s3_1c);
clear ir_s3_1c
x3_c1 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),1) -
remove_offset(datac.x3_filter(:,1))),0.009);

```

```

y3_c1 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),1) -
remove_offset(datac.y3_filter(:,1))),0.009);
z3_c1 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),1) -
remove_offset(datac.z3_filter(:,1))),0.009);

x3_c2 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),2) -
remove_offset(datac.x3_filter(:,2))),0.009);
y3_c2 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),2) -
remove_offset(datac.y3_filter(:,2))),0.009);
z3_c2 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),2) -
remove_offset(datac.z3_filter(:,2))),0.009);

x3_c3 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),3) -
remove_offset(datac.x3_filter(:,3))),0.009);
y3_c3 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),3) -
remove_offset(datac.y3_filter(:,3))),0.009);
z3_c3 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),3) -
remove_offset(datac.z3_filter(:,3))),0.009);

x3_c4 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),4) -
remove_offset(datac.x3_filter(:,4))),0.009);
y3_c4 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),4) -
remove_offset(datac.y3_filter(:,4))),0.009);
z3_c4 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),4) -
remove_offset(datac.z3_filter(:,4))),0.009);

x3_c5 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),5) -
remove_offset(datac.x3_filter(:,5))),0.009);
y3_c5 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),5) -
remove_offset(datac.y3_filter(:,5))),0.009);
z3_c5 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),5) -
remove_offset(datac.z3_filter(:,5))),0.009);

x3_c6 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),6) -
remove_offset(datac.x3_filter(:,6))),0.009);
y3_c6 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),6) -
remove_offset(datac.y3_filter(:,6))),0.009);
z3_c6 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),6) -
remove_offset(datac.z3_filter(:,6))),0.009);

x3_c7 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),7) -
remove_offset(datac.x3_filter(:,7))),0.009);
y3_c7 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),7) -
remove_offset(datac.y3_filter(:,7))),0.009);
z3_c7 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),7) -
remove_offset(datac.z3_filter(:,7))),0.009);

x3_c8 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),8) -
remove_offset(datac.x3_filter(:,8))),0.009);
y3_c8 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),8) -
remove_offset(datac.y3_filter(:,8))),0.009);

```

```

z3_c8 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),8) -
remove_offset(datac.z3_filter(:,8))),0.009);

x3_c9 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),9) -
remove_offset(datac.x3_filter(:,9))),0.009);
y3_c9 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),9) -
remove_offset(datac.y3_filter(:,9))),0.009);
z3_c9 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),9) -
remove_offset(datac.z3_filter(:,9))),0.009);

x3_c10 = bfilter(1000*(datac.x3_filter(i3_1c(1):i1_1c(2),10) -
remove_offset(datac.x3_filter(:,10))),0.009);
y3_c10 = bfilter(1000*(datac.y3_filter(i3_1c(1):i1_1c(2),10) -
remove_offset(datac.y3_filter(:,10))),0.009);
z3_c10 = bfilter(1000*(datac.z3_filter(i3_1c(1):i1_1c(2),10) -
remove_offset(datac.z3_filter(:,10))),0.009);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Difference Orientation Scans
%Sensor 1
ir_s1_1 = bfilter(do.i1_filter_mv,0.009);
[i1_1do] = edge_detector(ir_s1_1);
clear ir_s1_1
x1_base1do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),1) -
remove_offset(do.x1_filter(:,1))),0.009);
y1_base1do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),1) -
remove_offset(do.y1_filter(:,1))),0.009);
z1_base1do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),1) -
remove_offset(do.z1_filter(:,1))),0.009);

x1_base2do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),2) -
remove_offset(do.x1_filter(:,2))),0.009);
y1_base2do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),2) -
remove_offset(do.y1_filter(:,2))),0.009);
z1_base2do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),2) -
remove_offset(do.z1_filter(:,2))),0.009);

x1_base3do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),3) -
remove_offset(do.x1_filter(:,3))),0.009);
y1_base3do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),3) -
remove_offset(do.y1_filter(:,3))),0.009);
z1_base3do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),3) -
remove_offset(do.z1_filter(:,3))),0.009);

x1_base4do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),4) -
remove_offset(do.x1_filter(:,4))),0.009);
y1_base4do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),4) -
remove_offset(do.y1_filter(:,4))),0.009);
z1_base4do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),4) -
remove_offset(do.z1_filter(:,4))),0.009);

```

```

x1_base5do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),5) -
remove_offset(do.x1_filter(:,5))),0.009);
y1_base5do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),5) -
remove_offset(do.y1_filter(:,5))),0.009);
z1_base5do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),5) -
remove_offset(do.z1_filter(:,5))),0.009);

x1_base6do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),6) -
remove_offset(do.x1_filter(:,6))),0.009);
y1_base6do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),6) -
remove_offset(do.y1_filter(:,6))),0.009);
z1_base6do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),6) -
remove_offset(do.z1_filter(:,6))),0.009);

x1_base7do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),7) -
remove_offset(do.x1_filter(:,7))),0.009);
y1_base7do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),7) -
remove_offset(do.y1_filter(:,7))),0.009);
z1_base7do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),7) -
remove_offset(do.z1_filter(:,7))),0.009);

x1_base8do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),8) -
remove_offset(do.x1_filter(:,8))),0.009);
y1_base8do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),8) -
remove_offset(do.y1_filter(:,8))),0.009);
z1_base8do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),8) -
remove_offset(do.z1_filter(:,8))),0.009);

x1_base9do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),9) -
remove_offset(do.x1_filter(:,9))),0.009);
y1_base9do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),9) -
remove_offset(do.y1_filter(:,9))),0.009);
z1_base9do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),9) -
remove_offset(do.z1_filter(:,9))),0.009);

x1_base10do = bfilter(1000*(do.x1_filter(i1_1do(1):i1_1do(2),10) -
remove_offset(do.x1_filter(:,10))),0.009);
y1_base10do = bfilter(1000*(do.y1_filter(i1_1do(1):i1_1do(2),10) -
remove_offset(do.y1_filter(:,10))),0.009);
z1_base10do = bfilter(1000*(do.z1_filter(i1_1do(1):i1_1do(2),10) -
remove_offset(do.z1_filter(:,10))),0.009);
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%Sensor two
%loading x1 base pr2ofiles segments based on door2 detection
ir_s2_1 = bfilter(do.i2_filter_mv,0.009);
[i2_1do] = edge_detector(ir_s2_1);
clear ir_s2_1
x2_base1do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),1) -
remove_offset(do.x2_filter(:,1))),0.009);

```

```

y2_base1do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),1) -
remove_offset(do.y2_filter(:,1))),0.009);
z2_base1do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),1) -
remove_offset(do.z2_filter(:,1))),0.009);

x2_base2do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),2) -
remove_offset(do.x2_filter(:,2))),0.009);
y2_base2do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),2) -
remove_offset(do.y2_filter(:,2))),0.009);
z2_base2do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),2) -
remove_offset(do.z2_filter(:,2))),0.009);

x2_base3do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),3) -
remove_offset(do.x2_filter(:,3))),0.009);
y2_base3do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),3) -
remove_offset(do.y2_filter(:,3))),0.009);
z2_base3do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),3) -
remove_offset(do.z2_filter(:,3))),0.009);

x2_base4do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),4) -
remove_offset(do.x2_filter(:,4))),0.009);
y2_base4do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),4) -
remove_offset(do.y2_filter(:,4))),0.009);
z2_base4do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),4) -
remove_offset(do.z2_filter(:,4))),0.009);

x2_base5do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),5) -
remove_offset(do.x2_filter(:,5))),0.009);
y2_base5do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),5) -
remove_offset(do.y2_filter(:,5))),0.009);
z2_base5do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),5) -
remove_offset(do.z2_filter(:,5))),0.009);

x2_base6do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),6) -
remove_offset(do.x2_filter(:,6))),0.009);
y2_base6do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),6) -
remove_offset(do.y2_filter(:,6))),0.009);
z2_base6do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),6) -
remove_offset(do.z2_filter(:,6))),0.009);

x2_base7do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),7) -
remove_offset(do.x2_filter(:,7))),0.009);
y2_base7do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),7) -
remove_offset(do.y2_filter(:,7))),0.009);
z2_base7do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),7) -
remove_offset(do.z2_filter(:,7))),0.009);

x2_base8do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),8) -
remove_offset(do.x2_filter(:,8))),0.009);
y2_base8do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),8) -
remove_offset(do.y2_filter(:,8))),0.009);

```

```

z2_base8do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),8) -
remove_offset(do.z2_filter(:,8))),0.009);

x2_base9do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),9) -
remove_offset(do.x2_filter(:,9))),0.009);
y2_base9do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),9) -
remove_offset(do.y2_filter(:,9))),0.009);
z2_base9do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),9) -
remove_offset(do.z2_filter(:,9))),0.009);

x2_base10do = bfilter(1000*(do.x2_filter(i2_1do(1):i2_1do(2),10) -
remove_offset(do.x2_filter(:,10))),0.009);
y2_base10do = bfilter(1000*(do.y2_filter(i2_1do(1):i2_1do(2),10) -
remove_offset(do.y2_filter(:,10))),0.009);
z2_base10do = bfilter(1000*(do.z2_filter(i2_1do(1):i2_1do(2),10) -
remove_offset(do.z2_filter(:,10))),0.009);
%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%Sensor Three
%loading x1 base pr2ofiles segments based on door2 detection
ir_s3_1 = bfilter(do.i3_filter_mv(:,1),0.009);
[i3_1do] = edge_detector(ir_s3_1);
clear ir_s3_1
x3_base1do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),1) -
remove_offset(do.x3_filter(:,1))),0.009);
y3_base1do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),1) -
remove_offset(do.y3_filter(:,1))),0.009);
z3_base1do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),1) -
remove_offset(do.z3_filter(:,1))),0.009);

x3_base2do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),2) -
remove_offset(do.x3_filter(:,2))),0.009);
y3_base2do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),2) -
remove_offset(do.y3_filter(:,2))),0.009);
z3_base2do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),2) -
remove_offset(do.z3_filter(:,2))),0.009);

x3_base3do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),3) -
remove_offset(do.x3_filter(:,3))),0.009);
y3_base3do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),3) -
remove_offset(do.y3_filter(:,3))),0.009);
z3_base3do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),3) -
remove_offset(do.z3_filter(:,3))),0.009);

x3_base4do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),4) -
remove_offset(do.x3_filter(:,4))),0.009);
y3_base4do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),4) -
remove_offset(do.y3_filter(:,4))),0.009);
z3_base4do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),4) -
remove_offset(do.z3_filter(:,4))),0.009);

```



```

x3_base5do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),5) -
remove_offset(do.x3_filter(:,5))),0.009);
y3_base5do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),5) -
remove_offset(do.y3_filter(:,5))),0.009);
z3_base5do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),5) -
remove_offset(do.z3_filter(:,5))),0.009);

```

```

x3_base6do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),6) -
remove_offset(do.x3_filter(:,6))),0.009);
y3_base6do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),6) -
remove_offset(do.y3_filter(:,6))),0.009);
z3_base6do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),6) -
remove_offset(do.z3_filter(:,6))),0.009);

```

```

x3_base7do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),7) -
remove_offset(do.x3_filter(:,7))),0.009);
y3_base7do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),7) -
remove_offset(do.y3_filter(:,7))),0.009);
z3_base7do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),7) -
remove_offset(do.z3_filter(:,7))),0.009);

```

```

x3_base8do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),8) -
remove_offset(do.x3_filter(:,8))),0.009);
y3_base8do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),8) -
remove_offset(do.y3_filter(:,8))),0.009);
z3_base8do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),8) -
remove_offset(do.z3_filter(:,8))),0.009);

```

```

x3_base9do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),9) -
remove_offset(do.x3_filter(:,9))),0.009);
y3_base9do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),9) -
remove_offset(do.y3_filter(:,9))),0.009);
z3_base9do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),9) -
remove_offset(do.z3_filter(:,9))),0.009);

```

```

x3_base10do = bfilter(1000*(do.x3_filter(i3_1do(1):i3_1do(2),10) -
remove_offset(do.x3_filter(:,10))),0.009);
y3_base10do = bfilter(1000*(do.y3_filter(i3_1do(1):i3_1do(2),10) -
remove_offset(do.y3_filter(:,10))),0.009);
z3_base10do = bfilter(1000*(do.z3_filter(i3_1do(1):i3_1do(2),10) -
remove_offset(do.z3_filter(:,10))),0.009);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function 2
% (Delta B)^2 = (Bbx - Bcx)^2 + (Bby - Bcy)^2 + (Bbz - Bcz)^2
%Computing the difference of each component before squaring

```

```

[db,mb,lb] =
dimcheck(x1_base1,x1_base2,x1_base3,x1_base4,x1_base5,x1_base6,x1_base7,x1_base8,x1_base9,x1_base10);
[dc,mc,lc] = dimcheck(x1_c1,x1_c2,x1_c3,x1_c4,x1_c5,x1_c6,x1_c7,x1_c8,x1_c9,x1_c10);
ls = min([lc;lb]);
%Sensor 1
% Finding the difference between x components for Base profile
diffb1x1 = zeros(length(x1_base1),9);
diffb1x1(:,1) = Sig_Diff(x1_base1,x1_base2);
diffb1x1(:,2) = Sig_Diff(x1_base1,x1_base3);
diffb1x1(:,3) = Sig_Diff(x1_base1,x1_base4);
diffb1x1(:,4) = Sig_Diff(x1_base1,x1_base5);
diffb1x1(:,5) = Sig_Diff(x1_base1,x1_base6);
diffb1x1(:,6) = Sig_Diff(x1_base1,x1_base7);
diffb1x1(:,7) = Sig_Diff(x1_base1,x1_base8);
diffb1x1(:,8) = Sig_Diff(x1_base1,x1_base9);
diffb1x1(:,9) = Sig_Diff(x1_base1,x1_base10);

% Finding the difference between y components for Base profile
diffb1y1 = zeros(length(y1_base1),9);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base2);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base3);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base4);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base5);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base6);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base7);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base8);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base9);
diffb1y1(:,1) = Sig_Diff(y1_base1,y1_base10);

% Finding the difference between z components for Base profile
diffb1z1 = zeros(length(z1_base1),9);
diffb1z1(:,1) = Sig_Diff(z1_base1,z1_base2);
diffb1z1(:,2) = Sig_Diff(z1_base1,z1_base3);
diffb1z1(:,3) = Sig_Diff(z1_base1,z1_base4);
diffb1z1(:,4) = Sig_Diff(z1_base1,z1_base5);
diffb1z1(:,5) = Sig_Diff(z1_base1,z1_base6);
diffb1z1(:,6) = Sig_Diff(z1_base1,z1_base7);
diffb1z1(:,7) = Sig_Diff(z1_base1,z1_base8);
diffb1z1(:,8) = Sig_Diff(z1_base1,z1_base9);
diffb1z1(:,9) = Sig_Diff(z1_base1,z1_base10);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 2
diffb1x2 = zeros(length(x2_base1),9);
diffb1x2(:,1) = Sig_Diff(x2_base1,x2_base2);
diffb1x2(:,2) = Sig_Diff(x2_base1,x2_base3);
diffb1x2(:,3) = Sig_Diff(x2_base1,x2_base4);
diffb1x2(:,4) = Sig_Diff(x2_base1,x2_base5);
diffb1x2(:,5) = Sig_Diff(x2_base1,x2_base6);
diffb1x2(:,6) = Sig_Diff(x2_base1,x2_base7);

```

```

diffb1x2(:,7) = Sig_Diff(x2_base1,x2_base8);
diffb1x2(:,8) = Sig_Diff(x2_base1,x2_base9);
diffb1x2(:,9) = Sig_Diff(x2_base1,x2_base10);

```

% Finding the difference between y components for Base profile

```

diffb1y2 = zeros(length(y2_base1),9);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base2);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base3);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base4);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base5);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base6);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base7);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base8);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base9);
diffb1y2(:,1) = Sig_Diff(y2_base1,y2_base10);

```

% Finding the difference between z components for Base profile

```

diffb1z2 = zeros(length(z2_base1),9);
diffb1z2(:,1) = Sig_Diff(z2_base1,z2_base2);
diffb1z2(:,2) = Sig_Diff(z2_base1,z2_base3);
diffb1z2(:,3) = Sig_Diff(z2_base1,z2_base4);
diffb1z2(:,4) = Sig_Diff(z2_base1,z2_base5);
diffb1z2(:,5) = Sig_Diff(z2_base1,z2_base6);
diffb1z2(:,6) = Sig_Diff(z2_base1,z2_base7);
diffb1z2(:,7) = Sig_Diff(z2_base1,z2_base8);
diffb1z2(:,8) = Sig_Diff(z2_base1,z2_base9);
diffb1z2(:,9) = Sig_Diff(z2_base1,z2_base10);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

%Sensor 3

```

[db,mb,lb] =
dimcheck(x3_base1,x3_base2,x3_base3,x3_base4,x3_base5,x3_base6,x3_base7,x3_base8,x3_base9,x3_base10);
[dc,mc,lc] = dimcheck(x3_c1,x3_c2,x3_c3,x3_c4,x3_c5,x3_c6,x3_c7,x3_c8,x3_c9,x3_c10);
ls = min([lc;lb]);
clear db mb dc mc
diffb1x3 = zeros(ls,9);
diffb1x3(:,1) = Sig_Diff(x3_base1(1:ls,:),x3_base2(1:ls,:));
diffb1x3(:,2) = Sig_Diff(x3_base1(1:ls,:),x3_base3(1:ls,:));
diffb1x3(:,3) = Sig_Diff(x3_base1(1:ls,:),x3_base4(1:ls,:));
diffb1x3(:,4) = Sig_Diff(x3_base1(1:ls,:),x3_base5(1:ls,:));
diffb1x3(:,5) = Sig_Diff(x3_base1(1:ls,:),x3_base6(1:ls,:));
diffb1x3(:,6) = Sig_Diff(x3_base1(1:ls,:),x3_base7(1:ls,:));
diffb1x3(:,7) = Sig_Diff(x3_base1(1:ls,:),x3_base8(1:ls,:));
diffb1x3(:,8) = Sig_Diff(x3_base1(1:ls,:),x3_base9(1:ls,:));
diffb1x3(:,9) = Sig_Diff(x3_base1(1:ls,:),x3_base10(1:ls,:));

```

% Finding the difference between y components for Base profile

```

diffb1y3 = zeros(ls,9);
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base2(1:ls,:));
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base3(1:ls,:));

```

```

diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base4(1:ls,:));
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base5(1:ls,:));
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base6(1:ls,:));
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base7(1:ls,:));
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base8(1:ls,:));
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base9(1:ls,:));
diffb1y3(:,1) = Sig_Diff(y3_base1(1:ls,:),y3_base10(1:ls,:));

```

**% Finding the difference between z components for Base profile**

```

diffb1z3 = zeros(ls,9);
diffb1z3(:,1) = Sig_Diff(z3_base1(1:ls,:),z3_base2(1:ls,:));
diffb1z3(:,2) = Sig_Diff(z3_base1(1:ls,:),z3_base3(1:ls,:));
diffb1z3(:,3) = Sig_Diff(z3_base1(1:ls,:),z3_base4(1:ls,:));
diffb1z3(:,4) = Sig_Diff(z3_base1(1:ls,:),z3_base5(1:ls,:));
diffb1z3(:,5) = Sig_Diff(z3_base1(1:ls,:),z3_base6(1:ls,:));
diffb1z3(:,6) = Sig_Diff(z3_base1(1:ls,:),z3_base7(1:ls,:));
diffb1z3(:,7) = Sig_Diff(z3_base1(1:ls,:),z3_base8(1:ls,:));
diffb1z3(:,8) = Sig_Diff(z3_base1(1:ls,:),z3_base9(1:ls,:));
diffb1z3(:,9) = Sig_Diff(z3_base1(1:ls,:),z3_base10(1:ls,:));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

[db,mb,lb] =
dimcheck(x1_base1,x1_base2,x1_base3,x1_base4,x1_base5,x1_base6,x1_base7,x1_base8,x1_base9,x1_base10);
[dc,mc,lc] = dimcheck(x1_c1,x1_c2,x1_c3,x1_c4,x1_c5,x1_c6,x1_c7,x1_c8,x1_c9,x1_c10);
ls = min([lc;lb]);
clear db mb dc mc

```

**% Sensor 1**

**% Finding the difference between x components for Base profile vs conceal**

**% profile**

```

diffc1x1 = zeros(ls,9);
diffc1x1(:,1) = Sig_Diff(x1_base1(1:ls,:),x1_c2(1:ls,:));
diffc1x1(:,2) = Sig_Diff(x1_base1(1:ls,:),x1_c3(1:ls,:));
diffc1x1(:,3) = Sig_Diff(x1_base1(1:ls,:),x1_c4(1:ls,:));
diffc1x1(:,4) = Sig_Diff(x1_base1(1:ls,:),x1_c5(1:ls,:));
diffc1x1(:,5) = Sig_Diff(x1_base1(1:ls,:),x1_c6(1:ls,:));
diffc1x1(:,6) = Sig_Diff(x1_base1(1:ls,:),x1_c7(1:ls,:));
diffc1x1(:,7) = Sig_Diff(x1_base1(1:ls,:),x1_c8(1:ls,:));
diffc1x1(:,8) = Sig_Diff(x1_base1(1:ls,:),x1_c9(1:ls,:));
diffc1x1(:,9) = Sig_Diff(x1_base1(1:ls,:),x1_c10(1:ls,:));

```

**% Finding the difference between y components for Base profile vs conceal**

**% profile**

```

diffc1y1 = zeros(ls,9);
diffc1y1(:,1) = Sig_Diff(y1_base1(1:ls,:),y1_c2(1:ls,:));
diffc1y1(:,2) = Sig_Diff(y1_base1(1:ls,:),y1_c3(1:ls,:));
diffc1y1(:,3) = Sig_Diff(y1_base1(1:ls,:),y1_c4(1:ls,:));
diffc1y1(:,4) = Sig_Diff(y1_base1(1:ls,:),y1_c5(1:ls,:));
diffc1y1(:,5) = Sig_Diff(y1_base1(1:ls,:),y1_c6(1:ls,:));
diffc1y1(:,6) = Sig_Diff(y1_base1(1:ls,:),y1_c7(1:ls,:));

```

```

diffc1y1(:,7) = Sig_Diff(y1_base1(1:ls,:),y1_c8(1:ls,:));
diffc1y1(:,8) = Sig_Diff(y1_base1(1:ls,:),y1_c9(1:ls,:));
diffc1y1(:,9) = Sig_Diff(y1_base1(1:ls,:),y1_c10(1:ls,:));

```

**% Finding the difference between z components for Base profile vs conceal**

**% profile**

```

diffc1z1 = zeros(ls,9);
diffc1z1(:,1) = Sig_Diff(z1_base1(1:ls,:),z1_c2(1:ls,:));
diffc1z1(:,2) = Sig_Diff(z1_base1(1:ls,:),z1_c3(1:ls,:));
diffc1z1(:,3) = Sig_Diff(z1_base1(1:ls,:),z1_c4(1:ls,:));
diffc1z1(:,4) = Sig_Diff(z1_base1(1:ls,:),z1_c5(1:ls,:));
diffc1z1(:,5) = Sig_Diff(z1_base1(1:ls,:),z1_c6(1:ls,:));
diffc1z1(:,6) = Sig_Diff(z1_base1(1:ls,:),z1_c7(1:ls,:));
diffc1z1(:,7) = Sig_Diff(z1_base1(1:ls,:),z1_c8(1:ls,:));
diffc1z1(:,8) = Sig_Diff(z1_base1(1:ls,:),z1_c9(1:ls,:));
diffc1z1(:,9) = Sig_Diff(z1_base1(1:ls,:),z1_c10(1:ls,:));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

[db,mb,lb] =
dimcheck(x2_base1,x2_base2,x2_base3,x2_base4,x2_base5,x2_base6,x2_base7,x2_base8,x2_base9,x2_base10);

```

```

[dc,mc,lc] = dimcheck(x2_c1,x2_c2,x2_c3,x2_c4,x2_c5,x2_c6,x2_c7,x2_c8,x2_c9,x2_c10);

```

```

ls = min([lc;lb]);

```

```

clear db mb dc mc

```

**% Sensor 2**

**% Finding the difference between x components for Base profile vs conceal**

**% profile**

```

diffc1x2 = zeros(ls,9);
diffc1x2(:,1) = Sig_Diff(x2_base1(1:ls,:),x2_c2(1:ls,:));
diffc1x2(:,2) = Sig_Diff(x2_base1(1:ls,:),x2_c3(1:ls,:));
diffc1x2(:,3) = Sig_Diff(x2_base1(1:ls,:),x2_c4(1:ls,:));
diffc1x2(:,4) = Sig_Diff(x2_base1(1:ls,:),x2_c5(1:ls,:));
diffc1x2(:,5) = Sig_Diff(x2_base1(1:ls,:),x2_c6(1:ls,:));
diffc1x2(:,6) = Sig_Diff(x2_base1(1:ls,:),x2_c7(1:ls,:));
diffc1x2(:,7) = Sig_Diff(x2_base1(1:ls,:),x2_c8(1:ls,:));
diffc1x2(:,8) = Sig_Diff(x2_base1(1:ls,:),x2_c9(1:ls,:));
diffc1x2(:,9) = Sig_Diff(x2_base1(1:ls,:),x2_c10(1:ls,:));

```

**% Finding the difference between y components for Base profile vs conceal**

**% profile**

```

diffc1y2 = zeros(ls,9);
diffc1y2(:,1) = Sig_Diff(y2_base1(1:ls,:),y2_c2(1:ls,:));
diffc1y2(:,2) = Sig_Diff(y2_base1(1:ls,:),y2_c3(1:ls,:));
diffc1y2(:,3) = Sig_Diff(y2_base1(1:ls,:),y2_c4(1:ls,:));
diffc1y2(:,4) = Sig_Diff(y2_base1(1:ls,:),y2_c5(1:ls,:));
diffc1y2(:,5) = Sig_Diff(y2_base1(1:ls,:),y2_c6(1:ls,:));
diffc1y2(:,6) = Sig_Diff(y2_base1(1:ls,:),y2_c7(1:ls,:));
diffc1y2(:,7) = Sig_Diff(y2_base1(1:ls,:),y2_c8(1:ls,:));
diffc1y2(:,8) = Sig_Diff(y2_base1(1:ls,:),y2_c9(1:ls,:));
diffc1y2(:,9) = Sig_Diff(y2_base1(1:ls,:),y2_c10(1:ls,:));

```

```

% Finding the difference between z components for Base profile vs conceal
% profile
diffc1z2 = zeros(ls,9);
diffc1z2(:,1) = Sig_Diff(z2_base1(1:ls,:),z2_c2(1:ls,:));
diffc1z2(:,2) = Sig_Diff(z2_base1(1:ls,:),z2_c3(1:ls,:));
diffc1z2(:,3) = Sig_Diff(z2_base1(1:ls,:),z2_c4(1:ls,:));
diffc1z2(:,4) = Sig_Diff(z2_base1(1:ls,:),z2_c5(1:ls,:));
diffc1z2(:,5) = Sig_Diff(z2_base1(1:ls,:),z2_c6(1:ls,:));
diffc1z2(:,6) = Sig_Diff(z2_base1(1:ls,:),z2_c7(1:ls,:));
diffc1z2(:,7) = Sig_Diff(z2_base1(1:ls,:),z2_c8(1:ls,:));
diffc1z2(:,8) = Sig_Diff(z2_base1(1:ls,:),z2_c9(1:ls,:));
diffc1z2(:,9) = Sig_Diff(z2_base1(1:ls,:),z2_c10(1:ls,:));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[db,mb,lb] =
dimcheck(x3_base1,x3_base2,x3_base3,x3_base4,x3_base5,x3_base6,x3_base7,x3_base8,x3_base9,x3_base10);
[dc,mc,lc] = dimcheck(x3_c1,x3_c2,x3_c3,x3_c4,x3_c5,x3_c6,x3_c7,x3_c8,x3_c9,x3_c10);
ls = min([lc;lb]);
clear db mb dc mc
% Sensor 3
% Finding the difference between x components for Base profile vs conceal
% profile
diffc1x3 = zeros(ls,9);
diffc1x3(:,1) = Sig_Diff(x3_base1(1:ls,:),x3_c2(1:ls,:));
diffc1x3(:,2) = Sig_Diff(x3_base1(1:ls,:),x3_c3(1:ls,:));
diffc1x3(:,3) = Sig_Diff(x3_base1(1:ls,:),x3_c4(1:ls,:));
diffc1x3(:,4) = Sig_Diff(x3_base1(1:ls,:),x3_c5(1:ls,:));
diffc1x3(:,5) = Sig_Diff(x3_base1(1:ls,:),x3_c6(1:ls,:));
diffc1x3(:,6) = Sig_Diff(x3_base1(1:ls,:),x3_c7(1:ls,:));
diffc1x3(:,7) = Sig_Diff(x3_base1(1:ls,:),x3_c8(1:ls,:));
diffc1x3(:,8) = Sig_Diff(x3_base1(1:ls,:),x3_c9(1:ls,:));
diffc1x3(:,9) = Sig_Diff(x3_base1(1:ls,:),x3_c10(1:ls,:));

% Finding the difference between y components for Base profile vs conceal
% profile
diffc1y3 = zeros(ls,9);
diffc1y3(:,1) = Sig_Diff(y3_base1,y3_c2);
diffc1y3(:,2) = Sig_Diff(y3_base1,y3_c3);
diffc1y3(:,3) = Sig_Diff(y3_base1,y3_c4);
diffc1y3(:,4) = Sig_Diff(y3_base1,y3_c5);
diffc1y3(:,5) = Sig_Diff(y3_base1,y3_c6);
diffc1y3(:,6) = Sig_Diff(y3_base1,y3_c7);
diffc1y3(:,7) = Sig_Diff(y3_base1,y3_c8);
diffc1y3(:,8) = Sig_Diff(y3_base1,y3_c9);
diffc1y3(:,9) = Sig_Diff(y3_base1,y3_c10);

% Finding the difference between z components for Base profile vs conceal
% profile
diffc1z3 = zeros(ls,9);

```

```

diffc1z3(:,1) = Sig_Diff(z3_base1,z3_c2);
diffc1z3(:,2) = Sig_Diff(z3_base1,z3_c3);
diffc1z3(:,3) = Sig_Diff(z3_base1,z3_c4);
diffc1z3(:,4) = Sig_Diff(z3_base1,z3_c5);
diffc1z3(:,5) = Sig_Diff(z3_base1,z3_c6);
diffc1z3(:,6) = Sig_Diff(z3_base1,z3_c7);
diffc1z3(:,7) = Sig_Diff(z3_base1,z3_c8);
diffc1z3(:,8) = Sig_Diff(z3_base1,z3_c9);
diffc1z3(:,9) = Sig_Diff(z3_base1,z3_c10);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

**%Sensor 1**

**%Implementing the function 2 for the base profiles**

```

diffb1_sq(:,1) = diffb1x1(:,1).^2 + diffb1y1(:,1).^2 + diffb1z1(:,1).^2;
diffb1_sq(:,2) = diffb1x1(:,2).^2 + diffb1y1(:,2).^2 + diffb1z1(:,2).^2;
diffb1_sq(:,3) = diffb1x1(:,3).^2 + diffb1y1(:,3).^2 + diffb1z1(:,3).^2;
diffb1_sq(:,4) = diffb1x1(:,4).^2 + diffb1y1(:,4).^2 + diffb1z1(:,4).^2;
diffb1_sq(:,5) = diffb1x1(:,5).^2 + diffb1y1(:,5).^2 + diffb1z1(:,5).^2;
diffb1_sq(:,6) = diffb1x1(:,6).^2 + diffb1y1(:,6).^2 + diffb1z1(:,6).^2;
diffb1_sq(:,7) = diffb1x1(:,7).^2 + diffb1y1(:,7).^2 + diffb1z1(:,7).^2;
diffb1_sq(:,8) = diffb1x1(:,8).^2 + diffb1y1(:,8).^2 + diffb1z1(:,8).^2;
diffb1_sq(:,9) = diffb1x1(:,9).^2 + diffb1y1(:,9).^2 + diffb1z1(:,9).^2;

```

**%Implementing the function 2 for the base profiles vs conceal profiles**

```

diffc1_sq(:,1) = diffc1x1(:,1).^2 + diffc1y1(:,1).^2 + diffc1z1(:,1).^2;
diffc1_sq(:,2) = diffc1x1(:,2).^2 + diffc1y1(:,2).^2 + diffc1z1(:,2).^2;
diffc1_sq(:,3) = diffc1x1(:,3).^2 + diffc1y1(:,3).^2 + diffc1z1(:,3).^2;
diffc1_sq(:,4) = diffc1x1(:,4).^2 + diffc1y1(:,4).^2 + diffc1z1(:,4).^2;
diffc1_sq(:,5) = diffc1x1(:,5).^2 + diffc1y1(:,5).^2 + diffc1z1(:,5).^2;
diffc1_sq(:,6) = diffc1x1(:,6).^2 + diffc1y1(:,6).^2 + diffc1z1(:,6).^2;
diffc1_sq(:,7) = diffc1x1(:,7).^2 + diffc1y1(:,7).^2 + diffc1z1(:,7).^2;
diffc1_sq(:,8) = diffc1x1(:,8).^2 + diffc1y1(:,8).^2 + diffc1z1(:,8).^2;
diffc1_sq(:,9) = diffc1x1(:,9).^2 + diffc1y1(:,9).^2 + diffc1z1(:,9).^2;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

**%Sensor 2**

**%Implementing the function 2 for the base profiles**

```

diffb2_sq(:,1) = diffb1x2(:,1).^2 + diffb1y2(:,1).^2 + diffb1z2(:,1).^2;
diffb2_sq(:,2) = diffb1x2(:,2).^2 + diffb1y2(:,2).^2 + diffb1z2(:,2).^2;
diffb2_sq(:,3) = diffb1x2(:,3).^2 + diffb1y2(:,3).^2 + diffb1z2(:,3).^2;
diffb2_sq(:,4) = diffb1x2(:,4).^2 + diffb1y2(:,4).^2 + diffb1z2(:,4).^2;
diffb2_sq(:,5) = diffb1x2(:,5).^2 + diffb1y2(:,5).^2 + diffb1z2(:,5).^2;
diffb2_sq(:,6) = diffb1x2(:,6).^2 + diffb1y2(:,6).^2 + diffb1z2(:,6).^2;
diffb2_sq(:,7) = diffb1x2(:,7).^2 + diffb1y2(:,7).^2 + diffb1z2(:,7).^2;
diffb2_sq(:,8) = diffb1x2(:,8).^2 + diffb1y2(:,8).^2 + diffb1z2(:,8).^2;
diffb2_sq(:,9) = diffb1x2(:,9).^2 + diffb1y2(:,9).^2 + diffb1z2(:,9).^2;

```

**%Implementing the function 2 for the base profiles vs conceal profiles**

```

diffc2_sq(:,1) = diffc1x2(:,1).^2 + diffc1y2(:,1).^2 + diffc1z2(:,1).^2;
diffc2_sq(:,2) = diffc1x2(:,2).^2 + diffc1y2(:,2).^2 + diffc1z2(:,2).^2;

```



```

diffc2_sq(:,3) = diffc1x2(:,3).^2 + diffc1y2(:,3).^2 + diffc1z2(:,3).^2;
diffc2_sq(:,4) = diffc1x2(:,4).^2 + diffc1y2(:,4).^2 + diffc1z2(:,4).^2;
diffc2_sq(:,5) = diffc1x2(:,5).^2 + diffc1y2(:,5).^2 + diffc1z2(:,5).^2;
diffc2_sq(:,6) = diffc1x2(:,6).^2 + diffc1y2(:,6).^2 + diffc1z2(:,6).^2;
diffc2_sq(:,7) = diffc1x2(:,7).^2 + diffc1y2(:,7).^2 + diffc1z2(:,7).^2;
diffc2_sq(:,8) = diffc1x2(:,8).^2 + diffc1y2(:,8).^2 + diffc1z2(:,8).^2;
diffc2_sq(:,9) = diffc1x2(:,9).^2 + diffc1y2(:,9).^2 + diffc1z2(:,9).^2;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 3
%Implementing the function 2 for the base profiles
diffb3_sq(:,1) = diffb1x3(:,1).^2 + diffb1y3(:,1).^2 + diffb1z3(:,1).^2;
diffb3_sq(:,2) = diffb1x3(:,2).^2 + diffb1y3(:,2).^2 + diffb1z3(:,2).^2;
diffb3_sq(:,3) = diffb1x3(:,3).^2 + diffb1y3(:,3).^2 + diffb1z3(:,3).^2;
diffb3_sq(:,4) = diffb1x3(:,4).^2 + diffb1y3(:,4).^2 + diffb1z3(:,4).^2;
diffb3_sq(:,5) = diffb1x3(:,5).^2 + diffb1y3(:,5).^2 + diffb1z3(:,5).^2;
diffb3_sq(:,6) = diffb1x3(:,6).^2 + diffb1y3(:,6).^2 + diffb1z3(:,6).^2;
diffb3_sq(:,7) = diffb1x3(:,7).^2 + diffb1y3(:,7).^2 + diffb1z3(:,7).^2;
diffb3_sq(:,8) = diffb1x3(:,8).^2 + diffb1y3(:,8).^2 + diffb1z3(:,8).^2;
diffb3_sq(:,9) = diffb1x3(:,9).^2 + diffb1y3(:,9).^2 + diffb1z3(:,9).^2;

%Implementing the function 2 for the base profiles vs conceal profiles
diffc3_sq(:,1) = diffc1x3(:,1).^2 + diffc1y3(:,1).^2 + diffc1z3(:,1).^2;
diffc3_sq(:,2) = diffc1x3(:,2).^2 + diffc1y3(:,2).^2 + diffc1z3(:,2).^2;
diffc3_sq(:,3) = diffc1x3(:,3).^2 + diffc1y3(:,3).^2 + diffc1z3(:,3).^2;
diffc3_sq(:,4) = diffc1x3(:,4).^2 + diffc1y3(:,4).^2 + diffc1z3(:,4).^2;
diffc3_sq(:,5) = diffc1x3(:,5).^2 + diffc1y3(:,5).^2 + diffc1z3(:,5).^2;
diffc3_sq(:,6) = diffc1x3(:,6).^2 + diffc1y3(:,6).^2 + diffc1z3(:,6).^2;
diffc3_sq(:,7) = diffc1x3(:,7).^2 + diffc1y3(:,7).^2 + diffc1z3(:,7).^2;
diffc3_sq(:,8) = diffc1x3(:,8).^2 + diffc1y3(:,8).^2 + diffc1z3(:,8).^2;
diffc3_sq(:,9) = diffc1x3(:,9).^2 + diffc1y3(:,9).^2 + diffc1z3(:,9).^2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 1
%Finding the area under the curve of (Delta B)^2 for base profiles
adiffb1_sq(1,:) = trapz(diffb1_sq(:,1));
adiffb1_sq(2,:) = trapz(diffb1_sq(:,2));
adiffb1_sq(3,:) = trapz(diffb1_sq(:,3));
adiffb1_sq(4,:) = trapz(diffb1_sq(:,4));
adiffb1_sq(5,:) = trapz(diffb1_sq(:,5));
adiffb1_sq(6,:) = trapz(diffb1_sq(:,6));
adiffb1_sq(7,:) = trapz(diffb1_sq(:,7));
adiffb1_sq(8,:) = trapz(diffb1_sq(:,8));
adiffb1_sq(9,:) = trapz(diffb1_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffb1_sq(1,:) = max(diffb1_sq(:,1));
maxdiffb1_sq(2,:) = max(diffb1_sq(:,2));
maxdiffb1_sq(3,:) = max(diffb1_sq(:,3));

```



```

maxdiffb1_sq(4,:) = max(diffb1_sq(:,4));
maxdiffb1_sq(5,:) = max(diffb1_sq(:,5));
maxdiffb1_sq(6,:) = max(diffb1_sq(:,6));
maxdiffb1_sq(7,:) = max(diffb1_sq(:,7));
maxdiffb1_sq(8,:) = max(diffb1_sq(:,8));
maxdiffb1_sq(9,:) = max(diffb1_sq(:,9));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 2
%Finding the area under the curve of (Delta B)^2 for base profiles
adiffb2_sq(1,:) = trapz(diffb2_sq(:,1));
adiffb2_sq(2,:) = trapz(diffb2_sq(:,2));
adiffb2_sq(3,:) = trapz(diffb2_sq(:,3));
adiffb2_sq(4,:) = trapz(diffb2_sq(:,4));
adiffb2_sq(5,:) = trapz(diffb2_sq(:,5));
adiffb2_sq(6,:) = trapz(diffb2_sq(:,6));
adiffb2_sq(7,:) = trapz(diffb2_sq(:,7));
adiffb2_sq(8,:) = trapz(diffb2_sq(:,8));
adiffb2_sq(9,:) = trapz(diffb2_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffb2_sq(1,:) = max(diffb2_sq(:,1));
maxdiffb2_sq(2,:) = max(diffb2_sq(:,2));
maxdiffb2_sq(3,:) = max(diffb2_sq(:,3));
maxdiffb2_sq(4,:) = max(diffb2_sq(:,4));
maxdiffb2_sq(5,:) = max(diffb2_sq(:,5));
maxdiffb2_sq(6,:) = max(diffb2_sq(:,6));
maxdiffb2_sq(7,:) = max(diffb2_sq(:,7));
maxdiffb2_sq(8,:) = max(diffb2_sq(:,8));
maxdiffb2_sq(9,:) = max(diffb2_sq(:,9));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 3
%Finding the area under the curve of (Delta B)^2 for base profiles
adiffb3_sq(1,:) = trapz(diffb3_sq(:,1));
adiffb3_sq(2,:) = trapz(diffb3_sq(:,2));
adiffb3_sq(3,:) = trapz(diffb3_sq(:,3));
adiffb3_sq(4,:) = trapz(diffb3_sq(:,4));
adiffb3_sq(5,:) = trapz(diffb3_sq(:,5));
adiffb3_sq(6,:) = trapz(diffb3_sq(:,6));
adiffb3_sq(7,:) = trapz(diffb3_sq(:,7));
adiffb3_sq(8,:) = trapz(diffb3_sq(:,8));
adiffb3_sq(9,:) = trapz(diffb3_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffb3_sq(1,:) = max(diffb3_sq(:,1));
maxdiffb3_sq(2,:) = max(diffb3_sq(:,2));
maxdiffb3_sq(3,:) = max(diffb3_sq(:,3));
maxdiffb3_sq(4,:) = max(diffb3_sq(:,4));

```

```

maxdiffb3_sq(5,:) = max(diffb3_sq(:,5));
maxdiffb3_sq(6,:) = max(diffb3_sq(:,6));
maxdiffb3_sq(7,:) = max(diffb3_sq(:,7));
maxdiffb3_sq(8,:) = max(diffb3_sq(:,8));
maxdiffb3_sq(9,:) = max(diffb3_sq(:,9));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 1
%Finding the area under the curve of (Delta B)^2 for base profiles vs
%conceal
adiffc1_sq(1,:) = trapz(diffc1_sq(:,1));
adiffc1_sq(2,:) = trapz(diffc1_sq(:,2));
adiffc1_sq(3,:) = trapz(diffc1_sq(:,3));
adiffc1_sq(4,:) = trapz(diffc1_sq(:,4));
adiffc1_sq(5,:) = trapz(diffc1_sq(:,5));
adiffc1_sq(6,:) = trapz(diffc1_sq(:,6));
adiffc1_sq(7,:) = trapz(diffc1_sq(:,7));
adiffc1_sq(8,:) = trapz(diffc1_sq(:,8));
adiffc1_sq(9,:) = trapz(diffc1_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffc1_sq(1,:) = max(diffc1_sq(:,1));
maxdiffc1_sq(2,:) = max(diffc1_sq(:,2));
maxdiffc1_sq(3,:) = max(diffc1_sq(:,3));
maxdiffc1_sq(4,:) = max(diffc1_sq(:,4));
maxdiffc1_sq(5,:) = max(diffc1_sq(:,5));
maxdiffc1_sq(6,:) = max(diffc1_sq(:,6));
maxdiffc1_sq(7,:) = max(diffc1_sq(:,7));
maxdiffc1_sq(8,:) = max(diffc1_sq(:,8));
maxdiffc1_sq(9,:) = max(diffc1_sq(:,9));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 2
%Finding the area under the curve of (Delta B)^2 for base profiles vs
%conceal
adiffc2_sq(1,:) = trapz(diffc2_sq(:,1));
adiffc2_sq(2,:) = trapz(diffc2_sq(:,2));
adiffc2_sq(3,:) = trapz(diffc2_sq(:,3));
adiffc2_sq(4,:) = trapz(diffc2_sq(:,4));
adiffc2_sq(5,:) = trapz(diffc2_sq(:,5));
adiffc2_sq(6,:) = trapz(diffc2_sq(:,6));
adiffc2_sq(7,:) = trapz(diffc2_sq(:,7));
adiffc2_sq(8,:) = trapz(diffc2_sq(:,8));
adiffc2_sq(9,:) = trapz(diffc2_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffc2_sq(1,:) = max(diffc2_sq(:,1));
maxdiffc2_sq(2,:) = max(diffc2_sq(:,2));
maxdiffc2_sq(3,:) = max(diffc2_sq(:,3));

```

```

maxdiffc2_sq(4,:) = max(diffc2_sq(:,4));
maxdiffc2_sq(5,:) = max(diffc2_sq(:,5));
maxdiffc2_sq(6,:) = max(diffc2_sq(:,6));
maxdiffc2_sq(7,:) = max(diffc2_sq(:,7));
maxdiffc2_sq(8,:) = max(diffc2_sq(:,8));
maxdiffc2_sq(9,:) = max(diffc2_sq(:,9));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Sensor 3
%Finding the area under the curve of (Delta B)^2 for base profiles vs
%conceal
adiffc3_sq(1,:) = trapz(diffc3_sq(:,1));
adiffc3_sq(2,:) = trapz(diffc3_sq(:,2));
adiffc3_sq(3,:) = trapz(diffc3_sq(:,3));
adiffc3_sq(4,:) = trapz(diffc3_sq(:,4));
adiffc3_sq(5,:) = trapz(diffc3_sq(:,5));
adiffc3_sq(6,:) = trapz(diffc3_sq(:,6));
adiffc3_sq(7,:) = trapz(diffc3_sq(:,7));
adiffc3_sq(8,:) = trapz(diffc3_sq(:,8));
adiffc3_sq(9,:) = trapz(diffc3_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffc3_sq(1,:) = max(diffc3_sq(:,1));
maxdiffc3_sq(2,:) = max(diffc3_sq(:,2));
maxdiffc3_sq(3,:) = max(diffc3_sq(:,3));
maxdiffc3_sq(4,:) = max(diffc3_sq(:,4));
maxdiffc3_sq(5,:) = max(diffc3_sq(:,5));
maxdiffc3_sq(6,:) = max(diffc3_sq(:,6));
maxdiffc3_sq(7,:) = max(diffc3_sq(:,7));
maxdiffc3_sq(8,:) = max(diffc3_sq(:,8));
maxdiffc3_sq(9,:) = max(diffc3_sq(:,9));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[db,mb,lb] =
dimcheck(x1_base1,x1_base2,x1_base3,x1_base4,x1_base5,x1_base6,x1_base7,x1_base8,x1_base9,x1_base10);
[dc,mc,lc] =
dimcheck(x1_base1do,x1_base2do,x1_base3do,x1_base4do,x1_base5do,x1_base6do,x1_base7do,x1_base8do,x1_base9do,x1_base10do);
ls = min([lc;lb]);
clear db mb dc mc
% Sensor 1
% Finding the difference between x components for Base profile vs conceal
% profile
diffdox1 = zeros(ls,9);
diffdox1(:,1) = Sig_Diff(x1_base1(1:ls,:),x1_base2do(1:ls,:));
diffdox1(:,2) = Sig_Diff(x1_base1(1:ls,:),x1_base3do(1:ls,:));
diffdox1(:,3) = Sig_Diff(x1_base1(1:ls,:),x1_base4do(1:ls,:));

```

```

diffdcox1(:,4) = Sig_Diff(x1_base1(1:ls,:),x1_base5do(1:ls,:));
diffdcox1(:,5) = Sig_Diff(x1_base1(1:ls,:),x1_base6do(1:ls,:));
diffdcox1(:,6) = Sig_Diff(x1_base1(1:ls,:),x1_base7do(1:ls,:));
diffdcox1(:,7) = Sig_Diff(x1_base1(1:ls,:),x1_base8do(1:ls,:));
diffdcox1(:,8) = Sig_Diff(x1_base1(1:ls,:),x1_base9do(1:ls,:));
diffdcox1(:,9) = Sig_Diff(x1_base1(1:ls,:),x1_base10do(1:ls,:));

```

**% Finding the difference between y components for Base profile vs conceal  
% profile**

```

diffdoy1 = zeros(ls,9);
diffdoy1(:,1) = Sig_Diff(y1_base1(1:ls,:),y1_base2do(1:ls,:));
diffdoy1(:,2) = Sig_Diff(y1_base1(1:ls,:),y1_base3do(1:ls,:));
diffdoy1(:,3) = Sig_Diff(y1_base1(1:ls,:),y1_base4do(1:ls,:));
diffdoy1(:,4) = Sig_Diff(y1_base1(1:ls,:),y1_base5do(1:ls,:));
diffdoy1(:,5) = Sig_Diff(y1_base1(1:ls,:),y1_base6do(1:ls,:));
diffdoy1(:,6) = Sig_Diff(y1_base1(1:ls,:),y1_base7do(1:ls,:));
diffdoy1(:,7) = Sig_Diff(y1_base1(1:ls,:),y1_base8do(1:ls,:));
diffdoy1(:,8) = Sig_Diff(y1_base1(1:ls,:),y1_base9do(1:ls,:));
diffdoy1(:,9) = Sig_Diff(y1_base1(1:ls,:),y1_base10do(1:ls,:));

```

**% Finding the difference between z components for Base profile vs conceal  
% profile**

```

diffdoz1 = zeros(ls,9);
diffdoz1(:,1) = Sig_Diff(z1_base1(1:ls,:),z1_base2do(1:ls,:));
diffdoz1(:,2) = Sig_Diff(z1_base1(1:ls,:),z1_base3do(1:ls,:));
diffdoz1(:,3) = Sig_Diff(z1_base1(1:ls,:),z1_base4do(1:ls,:));
diffdoz1(:,4) = Sig_Diff(z1_base1(1:ls,:),z1_base5do(1:ls,:));
diffdoz1(:,5) = Sig_Diff(z1_base1(1:ls,:),z1_base6do(1:ls,:));
diffdoz1(:,6) = Sig_Diff(z1_base1(1:ls,:),z1_base7do(1:ls,:));
diffdoz1(:,7) = Sig_Diff(z1_base1(1:ls,:),z1_base8do(1:ls,:));
diffdoz1(:,8) = Sig_Diff(z1_base1(1:ls,:),z1_base9do(1:ls,:));
diffdoz1(:,9) = Sig_Diff(z1_base1(1:ls,:),z1_base10do(1:ls,:));

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

[db,mb,lb] =
dimcheck(x2_base1,x2_base2,x2_base3,x2_base4,x2_base5,x2_base6,x2_base7,x2_base8,x2_base9,x2_base10);
[dc,mc,lc] =
dimcheck(x2_base1do,x2_base2do,x2_base3do,x2_base4do,x2_base5do,x2_base6do,x2_base7do,x2_base8do,x2_base9do,x2_base10do);
ls = min([lc;lb]);
clear db mb dc mc

```

**% Sensor 2**

**% Finding the difference between x components for Base profile vs conceal  
% profile**

```

diffdcox2 = zeros(ls,9);
diffdcox2(:,1) = Sig_Diff(x2_base1(1:ls,:),x2_base2do(1:ls,:));
diffdcox2(:,2) = Sig_Diff(x2_base1(1:ls,:),x2_base3do(1:ls,:));
diffdcox2(:,3) = Sig_Diff(x2_base1(1:ls,:),x2_base4do(1:ls,:));
diffdcox2(:,4) = Sig_Diff(x2_base1(1:ls,:),x2_base5do(1:ls,:));

```

```

diffdxx2(:,5) = Sig_Diff(x2_base1(1:ls,:),x2_base6do(1:ls,:));
diffdxx2(:,6) = Sig_Diff(x2_base1(1:ls,:),x2_base7do(1:ls,:));
diffdxx2(:,7) = Sig_Diff(x2_base1(1:ls,:),x2_base8do(1:ls,:));
diffdxx2(:,8) = Sig_Diff(x2_base1(1:ls,:),x2_base9do(1:ls,:));
diffdxx2(:,9) = Sig_Diff(x2_base1(1:ls,:),x2_base10do(1:ls,:));

```

**% Finding the difference between y components for Base profile vs conceal**

**% profile**

```

diffdoy2 = zeros(ls,9);
diffdoy2(:,1) = Sig_Diff(y2_base1(1:ls,:),y2_base2do(1:ls,:));
diffdoy2(:,2) = Sig_Diff(y2_base1(1:ls,:),y2_base3do(1:ls,:));
diffdoy2(:,3) = Sig_Diff(y2_base1(1:ls,:),y2_base4do(1:ls,:));
diffdoy2(:,4) = Sig_Diff(y2_base1(1:ls,:),y2_base5do(1:ls,:));
diffdoy2(:,5) = Sig_Diff(y2_base1(1:ls,:),y2_base6do(1:ls,:));
diffdoy2(:,6) = Sig_Diff(y2_base1(1:ls,:),y2_base7do(1:ls,:));
diffdoy2(:,7) = Sig_Diff(y2_base1(1:ls,:),y2_base8do(1:ls,:));
diffdoy2(:,8) = Sig_Diff(y2_base1(1:ls,:),y2_base9do(1:ls,:));
diffdoy2(:,9) = Sig_Diff(y2_base1(1:ls,:),y2_base10do(1:ls,:));

```

**% Finding the difference between z components for Base profile vs conceal**

**% profile**

```

diffdoz2 = zeros(ls,9);
diffdoz2(:,1) = Sig_Diff(z2_base1(1:ls,:),z2_base2do(1:ls,:));
diffdoz2(:,2) = Sig_Diff(z2_base1(1:ls,:),z2_base3do(1:ls,:));
diffdoz2(:,3) = Sig_Diff(z2_base1(1:ls,:),z2_base4do(1:ls,:));
diffdoz2(:,4) = Sig_Diff(z2_base1(1:ls,:),z2_base5do(1:ls,:));
diffdoz2(:,5) = Sig_Diff(z2_base1(1:ls,:),z2_base6do(1:ls,:));
diffdoz2(:,6) = Sig_Diff(z2_base1(1:ls,:),z2_base7do(1:ls,:));
diffdoz2(:,7) = Sig_Diff(z2_base1(1:ls,:),z2_base8do(1:ls,:));
diffdoz2(:,8) = Sig_Diff(z2_base1(1:ls,:),z2_base9do(1:ls,:));
diffdoz2(:,9) = Sig_Diff(z2_base1(1:ls,:),z2_base10do(1:ls,:));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[db,mb,lb] =
dimcheck(x3_base1,x3_base2,x3_base3,x3_base4,x3_base5,x3_base6,x3_base7,x3_base8,x3_base9,x3_base10);
[dc,mc,lc] =
dimcheck(x3_base1do,x3_base2do,x3_base3do,x3_base4do,x3_base5do,x3_base6do,x3_base7do,x3_base8do,x3_base9do,x3_base10do);
ls = min([lc;lb]);
clear db mb dc mc

```

**% Sensor 3**

**% Finding the difference between x components for Base profile vs conceal**

**% profile**

```

diffdxx3 = zeros(ls,9);
diffdxx3(:,1) = Sig_Diff(x3_base1(1:ls,:),x3_base2do(1:ls,:));
diffdxx3(:,2) = Sig_Diff(x3_base1(1:ls,:),x3_base3do(1:ls,:));
diffdxx3(:,3) = Sig_Diff(x3_base1(1:ls,:),x3_base4do(1:ls,:));
diffdxx3(:,4) = Sig_Diff(x3_base1(1:ls,:),x3_base5do(1:ls,:));
diffdxx3(:,5) = Sig_Diff(x3_base1(1:ls,:),x3_base6do(1:ls,:));
diffdxx3(:,6) = Sig_Diff(x3_base1(1:ls,:),x3_base7do(1:ls,:));

```

```

diffdov3(:,7) = Sig_Diff(x3_base1(1:ls,:),x3_base8do(1:ls,:));
diffdov3(:,8) = Sig_Diff(x3_base1(1:ls,:),x3_base9do(1:ls,:));
diffdov3(:,9) = Sig_Diff(x3_base1(1:ls,:),x3_base10do(1:ls,:));

```

% Finding the difference between y components for Base profile vs conceal

% profile

```

diffdoy3 = zeros(ls,9);
diffdoy3(:,1) = Sig_Diff(y3_base1,y3_base2do);
diffdoy3(:,2) = Sig_Diff(y3_base1,y3_base3do);
diffdoy3(:,3) = Sig_Diff(y3_base1,y3_base4do);
diffdoy3(:,4) = Sig_Diff(y3_base1,y3_base5do);
diffdoy3(:,5) = Sig_Diff(y3_base1,y3_base6do);
diffdoy3(:,6) = Sig_Diff(y3_base1,y3_base7do);
diffdoy3(:,7) = Sig_Diff(y3_base1,y3_base8do);
diffdoy3(:,8) = Sig_Diff(y3_base1,y3_base9do);
diffdoy3(:,9) = Sig_Diff(y3_base1,y3_base10do);

```

% Finding the difference between z components for Base profile vs conceal

% profile

```

diffdoz3 = zeros(ls,9);
diffdoz3(:,1) = Sig_Diff(z3_base1,z3_base2do);
diffdoz3(:,2) = Sig_Diff(z3_base1,z3_base3do);
diffdoz3(:,3) = Sig_Diff(z3_base1,z3_base4do);
diffdoz3(:,4) = Sig_Diff(z3_base1,z3_base5do);
diffdoz3(:,5) = Sig_Diff(z3_base1,z3_base6do);
diffdoz3(:,6) = Sig_Diff(z3_base1,z3_base7do);
diffdoz3(:,7) = Sig_Diff(z3_base1,z3_base8do);
diffdoz3(:,8) = Sig_Diff(z3_base1,z3_base9do);
diffdoz3(:,9) = Sig_Diff(z3_base1,z3_base10do);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

%Sensor 1

%Implementing the function 2 for the base profiles vs conceal profiles

```

diffdo1_sq(:,1) = diffdov1(:,1).^2 + diffdoy1(:,1).^2 + diffdoz1(:,1).^2;
diffdo1_sq(:,2) = diffdov1(:,2).^2 + diffdoy1(:,2).^2 + diffdoz1(:,2).^2;
diffdo1_sq(:,3) = diffdov1(:,3).^2 + diffdoy1(:,3).^2 + diffdoz1(:,3).^2;
diffdo1_sq(:,4) = diffdov1(:,4).^2 + diffdoy1(:,4).^2 + diffdoz1(:,4).^2;
diffdo1_sq(:,5) = diffdov1(:,5).^2 + diffdoy1(:,5).^2 + diffdoz1(:,5).^2;
diffdo1_sq(:,6) = diffdov1(:,6).^2 + diffdoy1(:,6).^2 + diffdoz1(:,6).^2;
diffdo1_sq(:,7) = diffdov1(:,7).^2 + diffdoy1(:,7).^2 + diffdoz1(:,7).^2;
diffdo1_sq(:,8) = diffdov1(:,8).^2 + diffdoy1(:,8).^2 + diffdoz1(:,8).^2;
diffdo1_sq(:,9) = diffdov1(:,9).^2 + diffdoy1(:,9).^2 + diffdoz1(:,9).^2;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

%Sensor 2

%Implementing the function 2 for the base profiles vs conceal profiles

```

diffdo2_sq(:,1) = diffdox2(:,1).^2 + diffdoy2(:,1).^2 + diffdoz2(:,1).^2;
diffdo2_sq(:,2) = diffdox2(:,2).^2 + diffdoy2(:,2).^2 + diffdoz2(:,2).^2;
diffdo2_sq(:,3) = diffdox2(:,3).^2 + diffdoy2(:,3).^2 + diffdoz2(:,3).^2;
diffdo2_sq(:,4) = diffdox2(:,4).^2 + diffdoy2(:,4).^2 + diffdoz2(:,4).^2;
diffdo2_sq(:,5) = diffdox2(:,5).^2 + diffdoy2(:,5).^2 + diffdoz2(:,5).^2;
diffdo2_sq(:,6) = diffdox2(:,6).^2 + diffdoy2(:,6).^2 + diffdoz2(:,6).^2;
diffdo2_sq(:,7) = diffdox2(:,7).^2 + diffdoy2(:,7).^2 + diffdoz2(:,7).^2;
diffdo2_sq(:,8) = diffdox2(:,8).^2 + diffdoy2(:,8).^2 + diffdoz2(:,8).^2;
diffdo2_sq(:,9) = diffdox2(:,9).^2 + diffdoy2(:,9).^2 + diffdoz2(:,9).^2;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Sensor 3

```

**%Implementing the function 2 for the base profiles vs conceal profiles**

```

diffdo3_sq(:,1) = diffdox3(:,1).^2 + diffdoy3(:,1).^2 + diffdoz3(:,1).^2;
diffdo3_sq(:,2) = diffdox3(:,2).^2 + diffdoy3(:,2).^2 + diffdoz3(:,2).^2;
diffdo3_sq(:,3) = diffdox3(:,3).^2 + diffdoy3(:,3).^2 + diffdoz3(:,3).^2;
diffdo3_sq(:,4) = diffdox3(:,4).^2 + diffdoy3(:,4).^2 + diffdoz3(:,4).^2;
diffdo3_sq(:,5) = diffdox3(:,5).^2 + diffdoy3(:,5).^2 + diffdoz3(:,5).^2;
diffdo3_sq(:,6) = diffdox3(:,6).^2 + diffdoy3(:,6).^2 + diffdoz3(:,6).^2;
diffdo3_sq(:,7) = diffdox3(:,7).^2 + diffdoy3(:,7).^2 + diffdoz3(:,7).^2;
diffdo3_sq(:,8) = diffdox3(:,8).^2 + diffdoy3(:,8).^2 + diffdoz3(:,8).^2;
diffdo3_sq(:,9) = diffdox3(:,9).^2 + diffdoy3(:,9).^2 + diffdoz3(:,9).^2;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

**%Sensor 1**

**%Finding the area under the curve of (Delta B)^2 for base profiles vs**

**%conceal**

```

adiffdo1_sq(1,:) = trapz(diffdo1_sq(:,1));
adiffdo1_sq(2,:) = trapz(diffdo1_sq(:,2));
adiffdo1_sq(3,:) = trapz(diffdo1_sq(:,3));
adiffdo1_sq(4,:) = trapz(diffdo1_sq(:,4));
adiffdo1_sq(5,:) = trapz(diffdo1_sq(:,5));
adiffdo1_sq(6,:) = trapz(diffdo1_sq(:,6));
adiffdo1_sq(7,:) = trapz(diffdo1_sq(:,7));
adiffdo1_sq(8,:) = trapz(diffdo1_sq(:,8));
adiffdo1_sq(9,:) = trapz(diffdo1_sq(:,9));

```

**%Finding the maximum value of (Delta B)^2 for base profiles vs**

**%conceal profiles**

```

maxdiffdo1_sq(1,:) = max(diffdo1_sq(:,1));
maxdiffdo1_sq(2,:) = max(diffdo1_sq(:,2));
maxdiffdo1_sq(3,:) = max(diffdo1_sq(:,3));
maxdiffdo1_sq(4,:) = max(diffdo1_sq(:,4));
maxdiffdo1_sq(5,:) = max(diffdo1_sq(:,5));
maxdiffdo1_sq(6,:) = max(diffdo1_sq(:,6));
maxdiffdo1_sq(7,:) = max(diffdo1_sq(:,7));
maxdiffdo1_sq(8,:) = max(diffdo1_sq(:,8));
maxdiffdo1_sq(9,:) = max(diffdo1_sq(:,9));

```

```

%%%%%%%%%%%%%%
%%%%%%%%%%%%%%Sensor 2
%Finding the area under the curve of (Delta B)^2 for base profiles vs
%conceal
adiffdo2_sq(1,:) = trapz(diffdo2_sq(:,1));
adiffdo2_sq(2,:) = trapz(diffdo2_sq(:,2));
adiffdo2_sq(3,:) = trapz(diffdo2_sq(:,3));
adiffdo2_sq(4,:) = trapz(diffdo2_sq(:,4));
adiffdo2_sq(5,:) = trapz(diffdo2_sq(:,5));
adiffdo2_sq(6,:) = trapz(diffdo2_sq(:,6));
adiffdo2_sq(7,:) = trapz(diffdo2_sq(:,7));
adiffdo2_sq(8,:) = trapz(diffdo2_sq(:,8));
adiffdo2_sq(9,:) = trapz(diffdo2_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffdo2_sq(1,:) = max(diffdo2_sq(:,1));
maxdiffdo2_sq(2,:) = max(diffdo2_sq(:,2));
maxdiffdo2_sq(3,:) = max(diffdo2_sq(:,3));
maxdiffdo2_sq(4,:) = max(diffdo2_sq(:,4));
maxdiffdo2_sq(5,:) = max(diffdo2_sq(:,5));
maxdiffdo2_sq(6,:) = max(diffdo2_sq(:,6));
maxdiffdo2_sq(7,:) = max(diffdo2_sq(:,7));
maxdiffdo2_sq(8,:) = max(diffdo2_sq(:,8));
maxdiffdo2_sq(9,:) = max(diffdo2_sq(:,9));

%%%%%%%%%%%%%%
%%%%%%%%%%%%%%
%Sensor 3
%Finding the area under the curve of (Delta B)^2 for base profiles vs
%conceal
adiffdo3_sq(1,:) = trapz(diffdo3_sq(:,1));
adiffdo3_sq(2,:) = trapz(diffdo3_sq(:,2));
adiffdo3_sq(3,:) = trapz(diffdo3_sq(:,3));
adiffdo3_sq(4,:) = trapz(diffdo3_sq(:,4));
adiffdo3_sq(5,:) = trapz(diffdo3_sq(:,5));
adiffdo3_sq(6,:) = trapz(diffdo3_sq(:,6));
adiffdo3_sq(7,:) = trapz(diffdo3_sq(:,7));
adiffdo3_sq(8,:) = trapz(diffdo3_sq(:,8));
adiffdo3_sq(9,:) = trapz(diffdo3_sq(:,9));

%Finding the maximum value of (Delta B)^2 for base profiles vs
%conceal profiles
maxdiffdo3_sq(1,:) = max(diffdo3_sq(:,1));
maxdiffdo3_sq(2,:) = max(diffdo3_sq(:,2));
maxdiffdo3_sq(3,:) = max(diffdo3_sq(:,3));
maxdiffdo3_sq(4,:) = max(diffdo3_sq(:,4));
maxdiffdo3_sq(5,:) = max(diffdo3_sq(:,5));
maxdiffdo3_sq(6,:) = max(diffdo3_sq(:,6));
maxdiffdo3_sq(7,:) = max(diffdo3_sq(:,7));
maxdiffdo3_sq(8,:) = max(diffdo3_sq(:,8));

```



```

maxdiffdo3_sq(9,:) = max(diffdo3_sq(:,9));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sensor 1 - Circle
% Sensor 2 - Triangle
% Sensor 3 - Square
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Function 2
figure(1)
scatter(adiffb1_sq,maxdiffb1_sq,'b','o','SizeData',msize);
hold on
title('Sensor 1-3 Classification Graph - Function 2');
xlabel('Area');
ylabel('Max Value');
scatter(adiffc1_sq,maxdiffc1_sq,'r','o','SizeData',msize);
scatter(adiffb2_sq,maxdiffb2_sq,'b','^','SizeData',msize);
scatter(adiffc2_sq,maxdiffc2_sq,'r','^','SizeData',msize);
scatter(adiffb3_sq,maxdiffb3_sq,'b','s','SizeData',msize);
scatter(adiffc3_sq,maxdiffc3_sq,'r','s','SizeData',msize);

if (DO == 1)
scatter(adiffdo1_sq,maxdiffdo1_sq,'g','o','SizeData',msize);
scatter(adiffdo2_sq,maxdiffdo2_sq,'g','^','SizeData',msize);
scatter(adiffdo3_sq,maxdiffdo3_sq,'g','s','SizeData',msize);

bf2_i = [adiffdo1_sq maxdiffdo1_sq; adiffdo2_sq maxdiffdo2_sq; adiffdo3_sq maxdiffdo3_sq];
end

switch lower(segment)
case 'top'
%Upper Segment Case
bf2 = [adiffb1_sq maxdiffb1_sq;adiffb2_sq maxdiffb2_sq;adiffb3_sq maxdiffb3_sq];
cf2 = [adiffc1_sq maxdiffc1_sq;adiffc2_sq maxdiffc2_sq];
case 'bottom'
%Lower Segment Case
bf2 = [adiffb1_sq maxdiffb1_sq;adiffb2_sq maxdiffb2_sq;adiffb3_sq maxdiffb3_sq];
cf2 = [adiffc3_sq maxdiffc3_sq];
%[adiffc1_sq maxdiffc1_sq;adiffc2_sq maxdiffc2_sq];
otherwise
disp('Unknown Segment');
end

sigma_bf2 = cov(bf2);
sigma_cf2 = cov(cf2);
sigma_commonf2 = (length(bf2)*sigma_bf2 + length(cf2)*sigma_cf2)/(length(bf2)+length(cf2)-
2);

meanbf2 = mean(bf2);
meancf2 = mean(cf2);

```

```
scatter(meanbf2(1),meanbf2(2),'h','MarkerFaceColor','b','MarkerEdgeColor','k','SizeData',mmsize
)
scatter(meancf2(1),meancf2(2),'h','MarkerFaceColor','r','MarkerEdgeColor','k','SizeData',mmsize)
```

*%Calculating the mahalanobi distance from the means of C1 and C2*

```
mahal_sq_D12_f2 = (meanbf2'-meancf2')*(inv(sigma_commonf2))*(meanbf2' - meancf2');
mahal_D12_f2 = sqrt(mahal_sq_D12_f2);
```

*%Calculating the mahalanobi distance from each point to its own class*

```
mahal_C1_f2 = mahal(bf2,bf2);
```

```
mahal_C2_f2 = mahal(cf2,cf2);
```

*%the max mahalanobi distance for C1 and C2 for this particular sample*

```
mahal_D1_f2_max = sqrt(max(mahal_C1_f2));
```

```
mahal_D2_f2_max = sqrt(max(mahal_C2_f2));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

*% This section calculates the indexes*

```
observed = bf2; % These are the observed points
```

```
observed2 = cf2; % These are the observed points with concealed objects
```

*% This is the mahalanobi distance from the observed points to class 1*

```
%mahal_classify_D1_f2 = sqrt(mahal(observed,bf2)); %Method 1
```

*% This is the mahalanobi distance from the observed points to class 2*

```
%mahal_classify_D2_f2 = sqrt(mahal(observed,cf2)); %Method 1
```

*%Method 2*

```
mahal_classify_D1_f2 = zeros(length(observed),1);
```

```
mahal_classify_D2_f2 = zeros(length(observed),1);
```

```
for m2 = 1:length(observed)
```

```
mahal_classify_D1_f2(m2,1) = sqrt((observed(m2,:).'-
meanbf2')*(inv(sigma_commonf2))*(observed(m2,:).' - meanbf2')); %Method 2
```

```
mahal_classify_D2_f2(m2,1) = sqrt((observed(m2,:).'-
meancf2')*(inv(sigma_commonf2))*(observed(m2,:).' - meancf2')); %Method 2
```

```
end
```

```
mahal_classify_D1_f2_c = zeros(length(observed2),1);
```

```
mahal_classify_D2_f2_c = zeros(length(observed2),1);
```

```
for mc2 = 1:length(observed2)
```

```
mahal_classify_D1_f2_c(mc2,1) = sqrt((observed2(mc2,:).'-
meanbf2')*(inv(sigma_commonf2))*(observed2(mc2,:).' - meanbf2')); %Method 2
```

```
mahal_classify_D2_f2_c(mc2,1) = sqrt((observed2(mc2,:).'-
meancf2')*(inv(sigma_commonf2))*(observed2(mc2,:).' - meancf2')); %Method 2
```

```
end
```

*% Calculating the max mahal distance of C!*

*% clear mahal\_D1\_f2*

```

% mahal_D1_f2 = zeros(length(bf2),1);
% for x = 1:1:length(bf2);
% mahal_D1_f2(x,1) = sqrt((bf2(x,:)'-meanbf2')*(inv(sigma_commonf2))*(bf2(x,:)' -
meanbf2'));
% end
% display('new')
% display(max(mahal_D1_f2))
%Defining the vector size of threshold index and limit index
%Base Profiles
threshold_index_f2 = zeros(length(observed),1);
limit_index_f2 = zeros(length(observed),1);

for obs = 1:1:length(observed)
%threshold index for classification
threshold_index_f2(obs,1) = mahal_classify_D2_f2(obs,1)/(mahal_D12_f2 -
mahal_D1_f2_max);
%limit index
limit_index_f2(obs,1) = ((mahal_D12_f2/2) -
mahal_classify_D1_f2(obs,1))/(mahal_classify_D2_f2(obs,1)-(mahal_D12_f2/2));
end

%Conceal Objects
threshold_index_f2_c = zeros(length(observed2),1);
limit_index_f2_c = zeros(length(observed2),1);

for obs2 = 1:1:length(observed2)
%threshold index for classification
threshold_index_f2_c(obs2,1) = mahal_classify_D2_f2_c(obs2,1)/(mahal_D12_f2 -
mahal_D1_f2_max);
%limit index
limit_index_f2_c(obs2,1) = ((mahal_D12_f2/2) -
mahal_classify_D1_f2_c(obs2,1))/(mahal_classify_D2_f2_c(obs2,1)-(mahal_D12_f2/2));
end

if(DO==1)
observed3 = bf2_i; % These are the observed points with Spikes

mahal_classify_D1_f2_i = zeros(length(observed3),1);
mahal_classify_D2_f2_i = zeros(length(observed3),1);

for mci = 1:1:length(observed3)
mahal_classify_D1_f2_i(mci,1) = sqrt((observed3(mci,:).'-
meanbf2')*(inv(sigma_commonf2))*(observed3(mci,:).'- meanbf2')); %Method 2
mahal_classify_D2_f2_i(mci,1) = sqrt((observed3(mci,:).'-
meancf2')*(inv(sigma_commonf2))*(observed3(mci,:).'- meancf2')); %Method 2
end

%offsets or spikes
threshold_index_f2_i = zeros(length(observed3),1);
limit_index_f2_i = zeros(length(observed3),1);

```

```

for obs3 = 1:length(observed3)
%threshold index for classification
threshold_index_f2_i(obs3,1) = mahal_classify_D2_f2_i(obs3,1)/(mahal_D12_f2 -
mahal_D1_f2_max);
%limit index
limit_index_f2_i(obs3,1) = ((mahal_D12_f2/2) -
mahal_classify_D1_f2_i(obs3,1))/(mahal_classify_D2_f2_i(obs3,1)-(mahal_D12_f2/2));
end

end

figure(2)
scatter(limit_index_f2,threshold_index_f2,'b','SizeData',msize)
hold on
xlabel('Limit Index');
ylabel('Threshold Index')
scatter(limit_index_f2_c,threshold_index_f2_c,'r','SizeData',msize);

if(DO==1)
switch lower(purpose)
case('offset')
scatter(limit_index_f2_i,threshold_index_f2_i,'g','SizeData',msize);
case('spike')
scatter(limit_index_f2_i,threshold_index_f2_i,'k','SizeData',msize);
otherwise
end
end
line([-1.5,1.5],[1,1]);
line([0,0],[0,3]);

```

## MatLab Functions

```
function E = B_Squared(x, y, z)
%ver 1.1 - Nar Vang
%This function computes the energy of the signal at that instant
%The input parameters are x, y, and z components of the magnetic sensor out
%This function should be used when the signals offset has been removed and
%when x, y and z have the same vector length
E = zeros(length(x),1);
for i=1:length(x)
    E(i) = x(i,:)^2 + y(i,:)^2 + z(i,:)^2;
end
end
```

```
function [fX] = bfilter(D,Wn)
% Version 0.1 Stephane Roussel
% Version 0.2 Nar Vang
%
%
%Filter function: implements a 2nd orders Butterworth digital lowpass
%filter. The input is a set of data collected by the magnetic sensor that
%is; it's a matrix nx3 where n is the number of data point taken by each
%channel, and where the first column corresponds to time, the second
%corresponds to Vx and the third to Vy.
%
%Added the third component to the filter
%
% Set up the Butterworth filter of Nth order and a cutoff frequency of Wn
N = 2;
[B,A] = butter(N,Wn,'low');

%Compute the initial conditions for a single component
ic1 = (D(1)+D(2))/2;
z2 = B(3)*ic1-A(3)*ic1;
z1 = B(2)*ic1+z2-A(2)*ic1;
zi1 = [z1;z2];

% Filtered the data that was passed to the function with filter
fX = filter(B,A,D,zi1);
```

```
function [s] = bsquared_delta(x,y)
%This function computes the sum of the difference of B^2 signals of x and y
[a b l] = dimcheck2(x,y);
d = x(1:l,:)-y(1:l,:);
s = sum(abs(d));
end
```

```
function [dim min_dim_index_r l] = dimcheck2(a,b)
```

```

%v1.0 - Nar Vang
% This function checks the dimension of vectors a through j and stores the
% vector dimensions into a vector in sequential order as seen above.

%creates a zero vector of size 10
dim = zeros(2,1);

%stores the vector length of each input vectors into a specific index in
%dim variable
dim(1) = length(a);
dim(2) = length(b);

%output the row index of the smallest length
[min_dim_index_r] = find(dim == min(dim));

%outputs the length of the smallest vector
l = dim(min_dim_index_r);

%output is the length vector the vector inputs and the index of the minimum
%length
end

```

```

function diff = Sig_Diff(x, y)
%ver1.2 - Nar Vang
%
%Added the vector size chop
%
%This function takes in to vector array of the same length and takes the
%absolute difference of the two signal values at that index. The method
%outputs the difference of the signal values into the diff array
x_dim = length(x);
y_dim = length(y);
if (x_dim - y_dim > 0)
    x_chop = x_dim - y_dim;
    x_dummy = x(1:(x_dim - x_chop));
    y_dummy = y;
elseif (x_dim - y_dim < 0)
    y_chop = y_dim - x_dim;
    y_dummy = y(1:(y_dim - y_chop));
    x_dummy = x;
else
    x_dummy = x;
    y_dummy = y;
end

diff = zeros(length(x_dummy),1);
for i = 1:length(x_dummy)
    diff(i) = abs(x(i)-y(i));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end

```

```

function E = Sig_Ener(x, y, z)
%ver 1.1 - Nar Vang
%This function computes the energy of the signal at that instant
%The input parameters are x, y, and z components of the magnetic sensor out
%This function should be used when the signals offset has been removed and
%when x, y and z have the same vector length
E = zeros(length(x),1);
for i=1:length(x)
    E(i) = x(i)^2 + y(i)^2 + z(i)^2;
end
end

```

```

function S = Sig_Sum(x, y, z)
%ver 1.1 - Nar Vang
%This function computes the energy of the signal at that instant
%The input parameters are x, y, and z components of the magnetic sensor out
%This function should be used when the signals offset has been removed and
%when x, y and z have the same vector length
S = zeros(length(x),1);
for i=1:length(x)
    S(i) = abs(x(i)) + abs(y(i)) + abs(z(i));
end
end

```