

2006

## Ground penetrating RADAR (GPR) based system for nondestructive detection of interior defects in wooden logs

Dayakar Devaru  
*West Virginia University*

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

---

### Recommended Citation

Devaru, Dayakar, "Ground penetrating RADAR (GPR) based system for nondestructive detection of interior defects in wooden logs" (2006). *Graduate Theses, Dissertations, and Problem Reports*. 1764.  
<https://researchrepository.wvu.edu/etd/1764>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact [researchrepository@mail.wvu.edu](mailto:researchrepository@mail.wvu.edu).

# **Ground Penetrating RADAR (GPR) based System for Nondestructive Detection of Interior Defects in Wooden Logs**

**Dayakar Devaru**

**Thesis submitted to the  
College of Engineering and Mineral Resources  
at West Virginia University  
in partial fulfillment of the requirements  
for the degree of**

**Master of Science  
in  
Industrial Engineering**

**Bhaskaran Gopalakrishnan, Ph.D., P.E., Chair  
Udaya B. Halabe, Ph.D., P.E.  
Robert C. Creese, Ph.D., P.E.**

**Department of Industrial and Management Systems Engineering  
Morgantown, West Virginia  
2006**

**Keywords: Ground Penetrating Radar, GPR, MATLAB, Wood, Logs, Rectangular  
Filter, Subsurface Defects, Knots, Decays, Rots, Embedded Metals**

## **ABSTRACT**

### **Ground Penetrating RADAR (GPR) based System for Nondestructive Detection of Interior Defects in Wooden Logs**

**Dayakar Devaru**

A previous study used GPR to scan logs and the scanned data was processed and analyzed using RADAN software. This data processing using RADAN software needs an expert and is time consuming. Also, the output from RADAN software cannot be used to generate manufacturing process instructions since it will be in the form of an image. For online implementation of GPR scanning, a quick automated defect detection system with numerical output is required.

To incorporate automated defect detection system, a MATLAB algorithm has been developed. Validation of the MATLAB algorithm output has been done by comparing with the results of RADAN software. The results from the MATLAB algorithm are in agreement with the results of RADAN software. This algorithm also converts the location of the defect found in the GPR scanned data into X-Y coordinates in meters.

Developed Algorithm can process data to view only internal defects or both internal and surface defects. Noise reduction has been done by removing the bottom reflection of the log and edge effects. Automatic threshold calculation has detected all the major defects in the logs. The depth and length resolution of defect detection are comparable to RADAN software. Conversion of defect location into Y coordinates can be done either using dielectric constant value if known or by detecting the bottom of the log in the scan. Reflections in the scan have been removed by a trial code.

## **ACKNOWLEDGEMENTS**

I am thankful to my AEC Committee Chair and my research advisor, Dr. Bhaskaran Gopalakrishnan for his valuable guidance and encouragement throughout the period of my Masters of Science in Industrial Engineering (M.S.I.E) study and research at WVU. I am sincerely thankful for his patience and time when guiding me. I am thankful for the valuable suggestions provided by Dr. Udaya B. Halabe, and Dr. Robert C. Creese, members of my Advisory and Examining Committee.

I extend my thanks to Mr. Sachin Agrawal for his valuable work and guidance in this project. I also extend my thanks to Division of Forestry, WVU for providing valuable laboratory assistance in sawing the logs used in this research.

I am thankful to my friends, Mr. Vivekananda Dorairaj, Mr. Santosh Hooli and Mr. Dhananjay Rao Hejamadi for their valuable help in MATLAB coding during this project. I am thankful to my family for their moral support to do my higher studies.

I would like to acknowledge the funding provided by the U. S. Department of Agriculture for this research under Grant/Contract No. 2004-34158-14390.

## Table of Contents

<b>ABSTRACT</b> .....	ii
<b>Table of Contents</b> .....	iv
<b>Chapter 1 INTRODUCTION AND LITERATURE REVIEW</b> .....	1
<b>1.1 Description of the Problem</b> .....	1
<b>1.2 Need for Research</b> .....	2
<b>1.3 Research Objectives and Scope</b> .....	2
<b>1.4 Natural Wood Defects that Affect Mechanical Properties of Wood</b> .....	3
<b>1.5 Literature Review</b> .....	8
<b>Chapter 2 NONDESTRUCTIVE TESTING OF WOODEN LOGS USING GPR</b> ..	11
<b>2.1 Experimental Setup of GPR System</b> .....	11
<b>2.2 Data Processing Using RADAN 5.0 Software</b> .....	13
<b>2.3 Defect Detection in Log1 Using RADAN Software</b> .....	17
<b>Chapter 3 DEFECT DETECTION USING MATLAB ALGORITHM</b> .....	21
<b>3.1 Log1 Data Processing Using MATLAB</b> .....	21
<b>3.2 Linear Gain</b> .....	29
<b>3.3 Zero Correction</b> .....	29
<b>3.4 Clipping the Signal</b> .....	31
<b>3.5 Removing Bottom Reflection (Reflection from Wood/Air Interface)</b> .....	32
<b>3.6 Signal Filtering</b> .....	32
<b>3.7 Filters</b> .....	33
<b>3.8 Window Design Techniques for designing filters (Vinay &amp; John 2004)</b> .....	35
<b>3.9 Rectangular Window (Vinay &amp; John 2004)</b> .....	38
<b>3.10 High Pass Rectangular Filter</b> .....	39
<b>3.11 Designing Filters for Applications</b> .....	40
<b>3.12 Filters used in this research</b> .....	40
<b>3.13 Threshold</b> .....	47
<b>3.14 Calculation of depth of the defect detected by MATLAB algorithm</b> .....	50

<b>Chapter 4 DEFECT DETECTION IN LOGS.....</b>	<b>54</b>
<b>4.1 Log2 Data Processing using MATLAB.....</b>	<b>54</b>
<b>4.2 Log3 Data Processing using MATLAB.....</b>	<b>64</b>
<b>4.3 Log5 Data Processing using MATLAB.....</b>	<b>69</b>
<b>4.4 Log6 Data Processing using MATLAB.....</b>	<b>77</b>
<b>Chapter 5 DEFECT DETECTION IN LOG4 USING MATLAB ALGORITHM... 82</b>	<b>82</b>
<b>5.1 Log4 Data Processing using MATLAB.....</b>	<b>82</b>
<b>5.2 Algorithm to remove echoes.....</b>	<b>88</b>
<b>5.3 Classification of defects .....</b>	<b>90</b>
<b>5.4 Threshold Value Selection.....</b>	<b>91</b>
<b>Chapter 6 RESULTS OF THE TEST CONDUCTED ON ALGORITHM .....</b>	<b>93</b>
<b>6.1 Test Log along Mark 5 .....</b>	<b>93</b>
<b>6.2 Test Log along Mark 12 .....</b>	<b>96</b>
<b>6.3 Log11 along Line 5.....</b>	<b>99</b>
<b>Chapter 7 CONCLUSIONS AND FUTURE WORK.....</b>	<b>102</b>
<b>7.1 Conclusions.....</b>	<b>102</b>
<b>7.2 Future Work.....</b>	<b>104</b>
<b>REFERENCES.....</b>	<b>106</b>
<b>APPENDIX.....</b>	<b>108</b>

# Chapter 1

## INTRODUCTION AND LITERATURE REVIEW

### 1.1 Description of the Problem

Presently there are no suitable non-invasive methods for precisely detecting the subsurface defects in wooden logs in real time. Internal defects such as knots, decays, and embedded metals are of greatest concern for lumber production. While defects such as knots and decays (rots) are of major concern related to productivity and yield of high value wood products, embedded metals can damage the saw blade and significantly increase the down time and maintenance costs of saw mills. Currently, a large number of logs end up being discarded by saw mills, or result in low value wood products since they include defects. If these defects can be located ahead of time before the log is sawn, then significant increase in productivity can be achieved by optimizing the sawing process through the active control of saw blade's orientation and or the log orientation. This process can also prevent damage to the saw blade due to embedded metals, thus avoiding downtime and repair costs.

There has been considerable research in the field of nondestructive testing (NDT) of wood for detecting internal defects by using elastic waves. The current elastic wave based technologies such as ultrasound and stress wave used for detecting metals and defects in wood have the disadvantage that they are time consuming and lack the desired accuracy (Halabe et al. 1996). Also, scanning technologies like Computed Tomography, X-ray, Ultrasonic and Nuclear Magnetic Resonance have several disadvantages in terms of technical problems and cost involved (Schad et al. 1996, Ross et al. 1998). The metal detectors currently employed by saw mills cannot precisely determine the location of embedded metals, and they cannot detect other defects.

The major problem is locating the defects accurately and mapping their spatial extent. Once the defect is accurately located, automated systems could be developed for making the sawing operation more effective. Research conducted by Forest and Wildlife Research Center (2005) showed that precise location of defect combined with computer

analysis to determine optimal sawing pattern can increase productivity by 10% for hardwood and 5 to 8% for softwood. A study by Gupta et al. (1998) has demonstrated a gain of up to 21% by sawing logs under different orientations and using different sawing patterns.

In order to address the above problem, Ground Penetrating Radar (GPR) has been used to scan the wooden logs. Ground Penetrating Radar (GPR) has been widely used for civil engineering applications for nondestructive testing (NDT) of bridges and pavements Halabe et al. (1997). Ground Penetrating Radar can acquire data from logs much more rapidly compared to other NDT methods such as ultrasonics and stress wave techniques (Halabe et al. 1995, Muller et al. 2002). A previous study by Agrawal (2005) used GPR to scan logs and the scanned data was processed and analyzed using RADAN software.

## **1.2 Need for Research**

Radar data processing and analysis using RADAN software needs a RADAN expert and is time consuming. RADAN analyzed data will be in the form of an image which helps to see the defects location in the scan by a human operator but it cannot be used directly for generating manufacturing process instructions.

On-line implementation of nondestructive testing requires scanning of logs, processing of data, analyzing of data giving defects' location in X, Y coordinates and generation of manufacturing process instructions in real time. To achieve this objective a radar data processing algorithm has to be developed.

## **1.3 Research Objectives and Scope**

This research proposes to address the issues of nondestructive testing of wooden logs using ground penetrating radar in real time. The objectives of this research are:

1. Design and develop methods to process GPR data for defects.
2. Obtain defects location in 2-Dimensional coordinates.
3. Validation of defects.



This research involves the development of an algorithm in MATLAB software to achieve the above objectives. GPR scanned data used for this research was available for six different logs from a past study (Agrawal 2005). The results of the MATLAB algorithm have to be validated by comparing with the results of the RADAN software.

#### 1.4 Natural Wood Defects that Affect Mechanical Properties of Wood

The natural growth characteristics affect the properties of wood. Natural defects such as rots and knots affect the mechanical properties of the wood and hence result in low value timber products.

**Knots:** Knots are generally the portion of a branch that remain in the tree after the branch falls off or manually cut. They are considered as a major defect in lumber grading, since it adversely affects the use of lumber in construction and other applications. The distorted fibers around the knot lower the strength of wood. The reduction in strength is proportional to the size of a knot. Knots are classified as sound and unsound. An unsound knot is usually rotten. Location of knots is the most important information for grading of lumber. Predicting accurate location of knots is very important for modifying the sawing patterns to increase the lumber value. The knot on the surface does not provide information regarding the amount of fiber distortion or size of the knot inside (Green et al. 1999). The external and the internal knots are shown in Figure 1-1.



(a) External Knot



(b) Internal Knot

Figure 1-1 Knots (Agrawal 2005)

**Decay and Rot:** Decays or rots affect the quality and in turn grade of wood. Decays are generally located in the interior part of the wood where the moisture content will be very

high (Figure 1-2). Wood decay occurs primarily due to fungi, insects, bacteria, and marine bores. The fungi survive on organic materials provided by cell structure of wood. Temperature and moisture content play an important role in the development of decay. The decay slows down at temperature below 10° C and above 35° C. Also, the rate of decay increases at moisture content above the fiber saturation point (average 30%). Wood does not decay under dry conditions.



Figure 1-2 Decay or rot in wood (Agrawal 2005)

Discoloration of wood happens at early stages of fungal attack. Early stages of decay are very difficult to detect as compared to later stages. The decay can be classified into three types: soft-rots, white-rots, and brown-rots. Initial stages of wood decay are called soft-rots. They cause discoloration and stains and are difficult to recognize. White-rots and brown-rots are major decay and can be detected easily. The strength of wood is badly affected by decay. Decay initially affects the toughness of the wood followed by reduction in its strength. Grade of the lumber goes down by the presence of decay. Decay can be prevented by drying the wood and keeping the moisture content below 20% (Green et al. 1999).

**Embedded Metals:** Metals are mainly embedded by humans during the early stage of growth of the tree. These metals cause major problems to saw mills. Figure 1-3(a) shows an inserted metal rod in wood. Figure 1-3(b) shows an embedded metal nail around which the tree has grown. These nails constitute a major type of embedded metals. Embedded metals hit the saw blade during sawing and result in significant downtime and maintenance cost. Presently available testing techniques like metal detectors can detect the presence of metals but not the exact location of them.



Figure 1-3 Embedded metal in wood (Agrawal 2005)

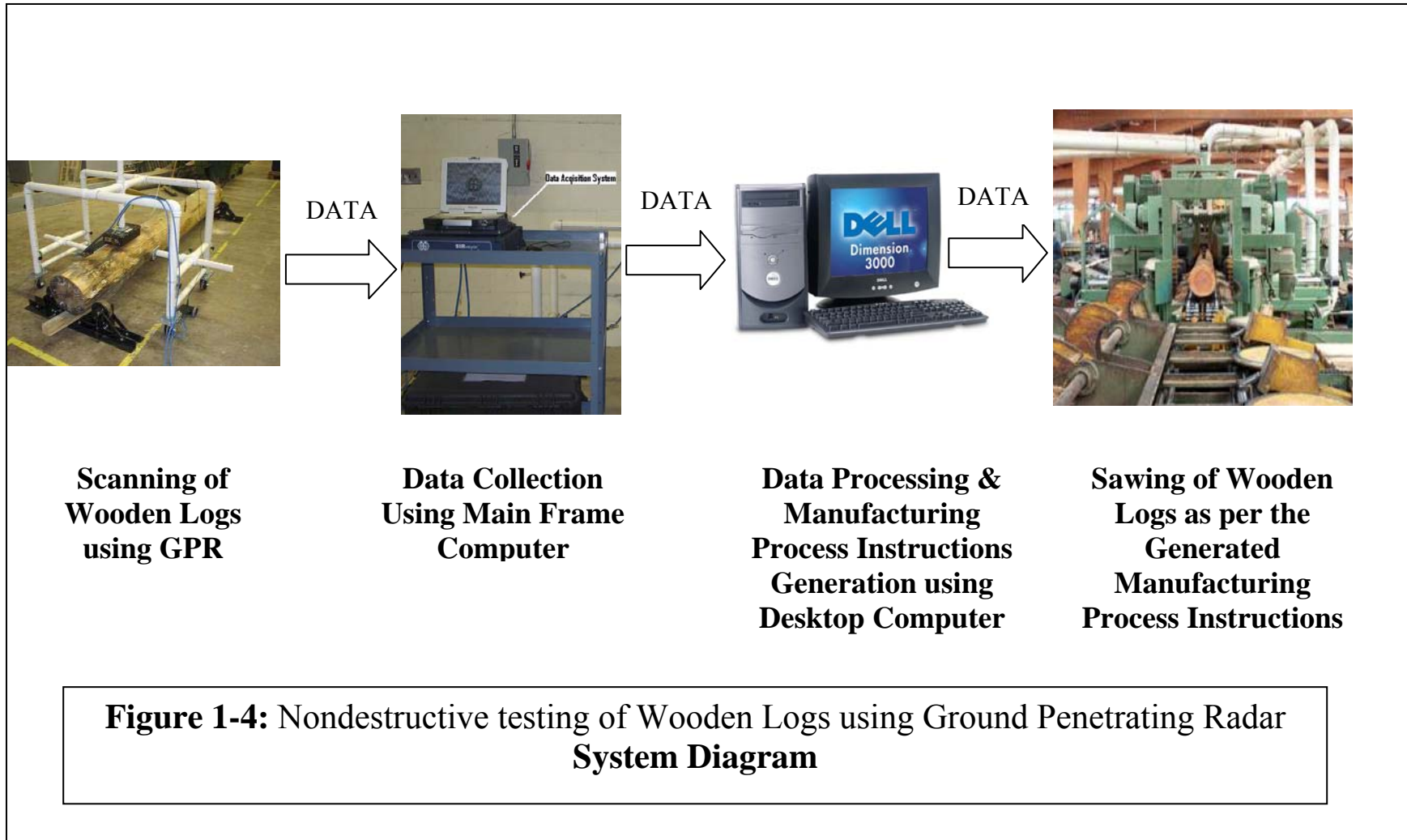
The success of this research can yield significant economic benefits to wood industry.

**Economic benefits:** About 12.2 billion board feet of hardwood timber was harvested in the Eastern United States for the production of lumber in 2000. Over 1 billion board feet of that hardwood timber was harvested in West Virginia. It is conservatively estimated that at least 3 percent of the logs coming from this timber were discarded after a metal detector signaled the presence of embedded metal objects in the log. This metal will be usually a bullet or a nail and if it hits the saw blade, it breaks the blade and results in down time. Thus 61 million board feet (Mmbf) of timber in the Eastern United States and 5 million board feet in West Virginia could not be processed into lumber because of the presence of metal in the logs. The economic value of this material includes the stumpage price paid for the logs as well as the value of the lumber that could have been recovered from the logs if they were not discarded. Since this material was not processed into lumber, one million dollars of timber was lost in West Virginia and 12.2 million dollars was lost in the Eastern United States. Use of the GPR system to pinpoint the location of embedded metal objects would allow the production of lumber from these logs. Thus, an additional 3.75 million dollars in West Virginia and 45.75 million dollars in the Eastern United States would have been resulted from the production of lumber from the presently discarded logs. The overall economic impact of possible savings in the year 2000 from waste reduction was estimated to be at least 58 million dollars in the Eastern United States and almost 5 million dollars in West Virginia.

It is anticipated that significant yield improvement can also be obtained from integrating the GPR system into the process of converting logs into lumber. GPR helps to cut the log optimally and improve the quality and yield of the lumber. These yield improvements are in addition to the reduction in waste from conversion of presently discarded logs into lumber. Assuming a 5 percent shift in the grades (Gupta et al. 1998) resulting from implementing the GPR system would result in conversion of #1 Common lumber to higher value FAS lumber equal to an estimated 41 million dollars in the Eastern United States and 3.5 million dollars in West Virginia.

**Energy benefits:** Overall energy savings of 5 percent can easily be achieved using the proposed GPR based intelligent system. This energy savings comes from not processing some of the defective logs after defect detection. In a typical wood processing facility in West Virginia, the average electricity consumption is 2400 MMBtu per year, or \$50,000 annually. Projecting 5 percent energy savings for the 200 saw mills, a net energy savings of about 24,000 MMBtu per year can be obtained at the minimum. This translates to savings of \$500,000 per year in West Virginia alone. The 200 saw mills in WV average 3 million kWh of energy usage per sawmill with an average production of 9 million board feet per year. Conservatively estimating savings of 3 percent specifically for process avoidance due to defect detection, savings of 270,000 board feet of wood not being processed by electrical motors can be expected. This energy savings will be 81,000 kWh per year for one saw mill and the projected energy savings for 200 saw mills in WV is likely to be 16 million kWh annually, based on 0.3 kWh per board feet. If similar energy savings are projected for all sawmills producing hardwood lumber in the United States, the total energy savings is estimated at 1.62 billion kWh annually.

**System Diagram:** Figure 1-4 shows the typical system diagram of the nondestructive testing of wooden logs using ground penetrating radar. Wooden logs are scanned using GPR antennas and the data is collected using mainframe computer. This data has to be transferred to a desktop computer for processing. Data is processed and analyzed using MATLAB algorithm. The analyzed data has to be used to generate manufacturing process instructions.



### **Cost of Testing Equipment and Implementation:**

The cost of a two-antenna GPR system is \$40,000. An 8-antenna system costs around \$160,000. The cost of the GPR system depends on number of antennas required to scan the log. MATLAB software, processing program and processing computer costs around \$1000 (NextTag.com 2006). Cost for making the set-up for moving the log, housing for the antenna, diameter measurement equipment etc. in a factory setting will cost around \$10,000.

### **1.5 Literature Review**

Canpolar, Inc. (1987) conducted tests to assess Impulse Radar to detect decay in hardwoods. A-cubed pulseEKKO I impulse radar system with a center frequency of 700 MHz was used to scan twelve bolts of Aspen with diameter ranging from 0.1 to 0.12 m. The electrical characteristic of defect-free and decayed wood radar response was different. Decayed sample had more ringing (echoes) in the radar signal.

Muller (2002), tested timber girders from an existing and a demolished bridge using GPR. These girders had circular cross-sections with diameter varying between 0.35 m to 0.40 m. EM wave range (two-way travel time) for the GPR recorded data was 11 ns. The GPR testing used a ground coupled dipole antenna with central frequency of 1.2 GHz manufactured by Geophysical Survey Systems, Inc. (GSSI). Other NDT techniques like ultrasonic and gamma ray transmissions were also used for testing. Of all the techniques GPR was found to be the most reliable NDT method for assessing internal defects in wood.

In 1994, Detection Science and Forest Product Laboratory (FPL) tried GPR for nondestructive testing of wood. Ground coupled antennas with center frequency of 1.2 GHz were used for this research (Detection Sciences, Inc. 1994). Sixteen specimens of 1.22 m length and 0.3 m diameter were tested in 3 longitudinal planes to detect defects and interaction of radar with log specimens. It was found that uniform, high grade logs can be easily distinguished from logs having more knots and internal defects.

Detection Sciences, Inc. (1994) demonstrated the feasibility of inspecting wooden beams using impulse radar. Few beams were tested using impulse radar in Hasbro Toy Factory in Pawtucket, Rhode Island. The radar system was capable of detecting internal rot. A small hand held radar antenna with center frequency of 900 MHz was used for this research. It was concluded that wood with no defects will result in a relatively uniform travel time of the signal. On the other hand, portions of the wood with decay or rot will show an increase in travel time and result in a non-uniform signal output.

Craig and Kelvin (2004) of Forest research and development division tested the ability of GPR, with 500 MHz, 800 MHz and 1 GHz antennas, to detect tree roots and determine root size by burying roots in a 32 m<sup>3</sup> pit containing damp sand. Within this test bed, tree roots were buried in two configurations: (1) roots of various diameters (1–10 cm) were buried at a single depth (50 cm); and (2) roots of similar diameter (about 5 cm) were buried at various depths (15–155 cm). Radar scanning was done in the direction perpendicular to the buried roots. Radar profile normalization, filtration and migration were undertaken based on standard algorithms. All antennas produced characteristic reflection hyperbolas on the radar profiles allowing visual identification of most of the root locations. The 800 MHz antenna resulted in the clearest radar profiles. An unsupervised, maximum-convexity migration algorithm was used to focus information contained in the hyperbolas back to a point. This resulted in a significant gain in clarity with roots appearing as discrete shapes, thereby reducing confusion due to overlapping of hyperbolas when many roots are detected.

Further, parameters extracted from the resultant waveform through the center of a root were correlated with the root diameter. The 500 MHz antenna showed good waveform parameter and root correlation than the other two antennas. A multiple regression model based on the extracted parameters was calibrated on half of the data ( $R^2 = 0.89$ ). This model produced good predictions when tested on the remaining data. Root diameters were predicted with a root mean squared error of 0.6 cm, allowing detection and quantification of roots as small as 1 cm in diameter. The advantage of this processing technique is that it produces results independently of signal strength.

Quan Zhu and Leslie M. Collins (2005) proposed GPR as an alternative to classical electromagnetic induction techniques for the landmine detection problem. The Wichmann/Niitek GPR system provided a good platform for a novel GPR-based antitank mine detection and classification algorithm development due to its extremely high Signal to Noise Ratio. The mines formed hyperbolas in the time-domain data record of GPR scan. These hyperbolas were extracted using an algorithm that had two steps: feature extraction and classification. Preprocessing was also considered to remove both stationary effects and non-stationary drift of the data and to improve the contrast of the desired hyperbolas. The feature extraction involved fitting of a polynomial to the maximum points of the reflected signal from the mine. The feature classification involved testing of the fitted polynomial for hyperbola. The receiver operating characteristic results shows that the polynomial fitting method is better than hidden Markov models.

Agrawal (2005) of Department of Civil and Environmental Engineering at West Virginia University used GPR to detect the internal defects such as rots, knots and metals in wooden logs. Subsurface Interface Radar (SIR) System manufactured by GSSI with antenna central frequency of 900 MHz was used for this research. Six wooden logs were scanned using this GPR system and the scanned data was processed using RADAN software. The study was successful in detecting the defects like metals, knots and rots with a defect depth resolution of 0.04 m and length resolution of 0.05 m.

**Conclusion:** The above literature review indicates that extensive research is still needed in terms of mapping the extent of subsurface defects in wooden logs. Most of the algorithms discussed in the above literature review do not apply to wooden logs. The data processing method used in Agrawal (2005) is manual method and the results can be interpreted only by an RADAN expert. Extensive research is needed to develop an automated algorithm which can detect defects in radar scanned data and give result in a machine readable format.



## Chapter 2

### NONDESTRUCTIVE TESTING OF WOODEN LOGS USING GPR

#### 2.1 Experimental Setup of GPR System

The GPR system used for this particular research was assembled by Agrawal (2005) and utilizes the Subsurface Interface Radar (SIR) System, manufactured by Geophysical Survey System, Inc. (GSSI). This GPR system incorporates ground coupled antennas for deeper penetration and has the added advantage that data collection and processing can be done simultaneously thus making it the best nondestructive scanning technology for on-line implementation. The GPR equipment consists of a data acquisition (DAQ) mainframe system and two 900 MHz antennas. The data acquisitions system houses the control unit and laptop computer based storage and display devices. The computer includes data acquisition as well as RADAN 5.0 data processing software. The transmitter and receiver are both incorporated in the same antenna, and both antennas are identical. These 900 MHz antennas provide penetration depths of up to 1 m even in moist logs and at the same time provide the best possible resolution for this penetration depth. A movable antenna deployment frame was built in-house, and a surveyor wheel with built-in optical encoder was attached to it for acquiring distance information (Agrawal 2005). The general setup of the scanning arrangement is shown in Figure 2-1.

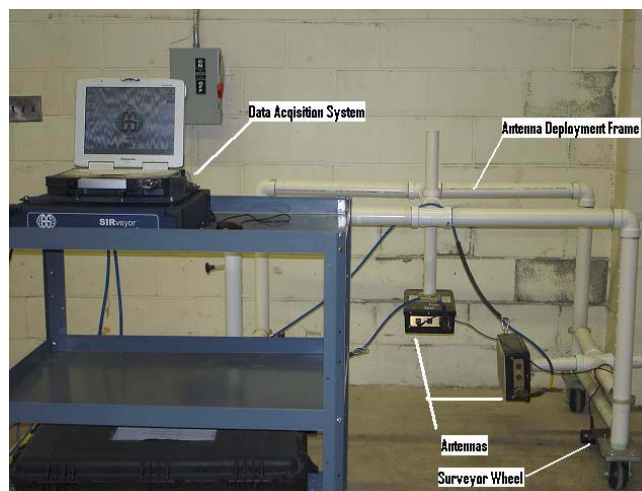


Figure 2-1 General setup of the GPR data acquisition system used for scanning logs  
(Agrawal 2005)

All the data was collected in the survey wheel mode since this mode provides precise information about distance along the length of the log in the GPR data (Agrawal, 2005). In this mode, data is collected based on the rotation of the survey wheel with the sampling rate set by the user. There would be no data collection if there is no rotation of the survey wheel (Agrawal, 2005).

The various data collection settings and parameters are shown in a header file in Figure 2-2 (Agrawal, 2005). It is very important to first set the antenna configuration name to the correct channel, namely channel 1, channel 2, or multi-channel (using both channels). The data collection parameters including scans/second, scans/meter, and meter/mark (marker setting) were user-specified inputs that respectively affected how many scans of GPR data are collected in a second, how many scans are collected based on distance traveled, and how many visual marks will be placed at a specified distance. Other user defined parameters include time range (ns), samples/scan and bits/sample that affect the sample depth and resolution of the data, and also the signal quality. The other parameters that had to be specified were dielectric constant, antenna transmit rate, and configuration (Comp) of the transmitter and receiver antennas. Typical values for the data collection parameters for this study are shown in Figure 2-2 (Agrawal, 2005).

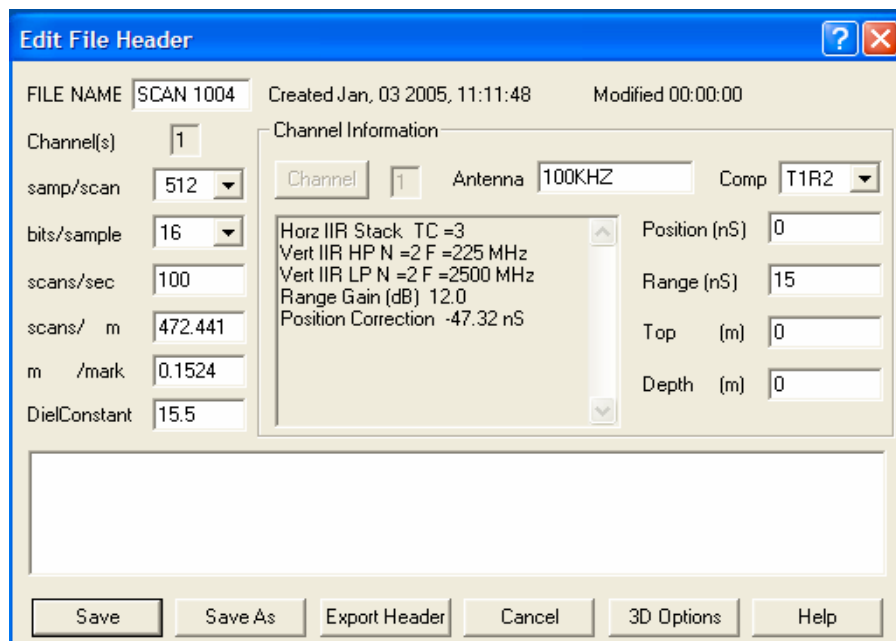


Figure 2-2 GPR data collection parameters (Agrawal 2005)

The Top and Depth parameters shown in Figure 2-2 are processing parameters used to specify the top location and depth of the full GPR scan, if known. These parameters were not used in this study since the computation of the dielectric constant usually results in more accurate depth estimations. The dielectric constant was computed by Agrawal (2005) based on the log diameter and the travel time to the bottom of the log.

## 2.2 Data Processing Using RADAN 5.0 Software

GPR scanned data of wooden logs has been processed using RADAN 5.0 software by researchers in the Department of Civil and Environmental Engineering of West Virginia University (Agrawal 2005). RADAN 5.0 is one of the commonly used software packages for processing GPR scanned data of bridges, pavements etc., (Halabe, U. B., Petro, S. H., Ganga Rao 1995). The post-processing of the GPR data is necessary to enhance the features (signal echoes) in radar data scan and reduce the noise. This greatly helps in detecting the subsurface defects and identifying the bottom of the log. Figure 2-3 shows the typical radar signal without any processing.

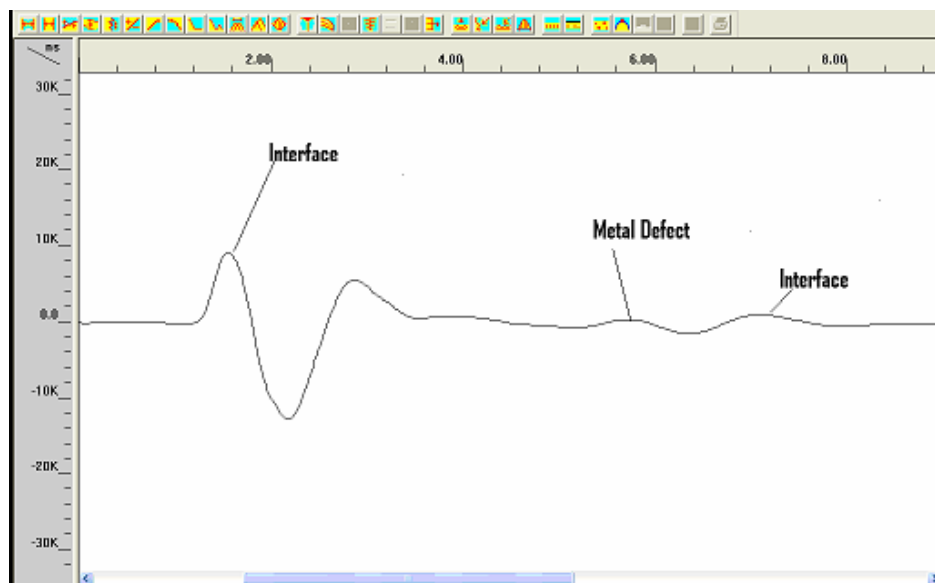


Figure 2-3 Typical radar signal observed using GPR oscilloscope mode (Agrawal 2005)

Basic data post-processing involved some fundamental manipulation of raw data to enhance the data for easier data interpretation. The basic steps involved in post-processing are described in detail by Agrawal (2005) and are as follows:

1. Range gain
2. Zero correction
3. Background removal and Noise filtering.

### **Range Gain**

Radar signals are prone to attenuation with increasing depth. Because of attenuation, the information at greater depths is not as clear and as reliable as the data near the surface of the antenna. RADAN software allows increasing time (or depth) dependent gain which compensates for amplitude reduction with depth. One of the major problems is that the variation of attenuation with depth is not uniform. In general, the attenuation of radar signals is low near the surface but very high towards the center of the logs due to high moisture content in the center (Agrawal 2005).

RADAN provides three different types of gains: Automatic Gain, Linear Gain, and Exponential Gain. In this research, a linear gain was applied to the raw GPR data since it provided significant gain for deeper and more attenuated echoes. The echo from the bottom of the log and some of the internal features became more obvious after the gain was applied (Agrawal 2005).

### **Zero Correction**

Zero correction is a process that is used to vertically adjust the position of the whole GPR scan in the data window so that the depth can be measured with respect to the ground surface. This correction involves shifting the first positive peak of the direct wave from the antenna (i.e., reflection from the antenna-log interface) in such a way that it becomes centered at the top edge of the data window, which then corresponds to the ground surface. After the zero correction is applied, the “Position” parameter in the file header has to be changed to a value of zero. This process allows estimation of the correct depth of any observed feature after the correct dielectric constant is used (Agrawal 2005).

### **Background removal and Noise Filtering**

After applying the zero correction the data must be filtered (Agrawal 2005). Filtering removes direct coupling surface reflections, flat-lying ringing system noise and

snow like peaks. The flat-lying ringing system noise is characterized by flat-lying horizontal bands, usually with lower frequency than that of the real reflections in the data set. This type of noise is usually most prevalent when the range is set near the maximum limits for antenna. Another type of noise is the high frequency noise which result in “snow-like” noise in the data, which is most prevalent when the range is set near the maximum limits for antenna or when large amount of gain is used.

The noise in the acquired data can obscure real reflection near the surface or at greater depth. These noises can be removed by using background removal (horizontal high pass filtering), vertical high frequency filter, and vertical low frequency filter. These features are all available under the same tool called Finite Impulse Response (FIR) filter. There are filters called Boxcar and Triangle filters under FIR filter in RADAN 5.0 software. The Boxcar filter is a rectangular window function while the Triangle filter emphasizes the center of the filter more heavily than the ends of the filter. The Boxcar filter was chosen for this research since it provided a uniform window function. The GPR scan looks considerably clearer after the FIR filter was applied (Agrawal 2005).

Background removal is done by applying Horizontal High Pass (Background Removal) Filter. The background removal number is set to one less than the number of sample points in each waveform ( $512 - 1 = 511$  points). The Horizontal filter also provides a Stacking option, which is a Horizontal Low Pass filter designed to remove high frequency “snow like” noise. The Vertical High Pass Filter with a cut-off frequency corresponding to about  $1/3$  the center frequency of the antenna ( $1/3 \times 900 \text{ MHz} = 300 \text{ MHz}$ ) is used if the desired horizontal features are of higher frequency content (i.e., low frequency noise is removed) (Agrawal 2005).

The Vertical Low Pass filter was set to a cut-off frequency that is twice the center frequency of the antenna ( $2 \times 900 \text{ MHz} = 1800 \text{ MHz}$ ) (Agrawal 2005). Low pass filter reduces the high frequency “snow like” noise. Figure 2-4 shows the sample specifications of all the data processing operations done on the radar data. Figure 2-5 shows the radar signal after applying range gain, zero correction, background removal and FIR low pass and high pass filters (Agrawal 2005).

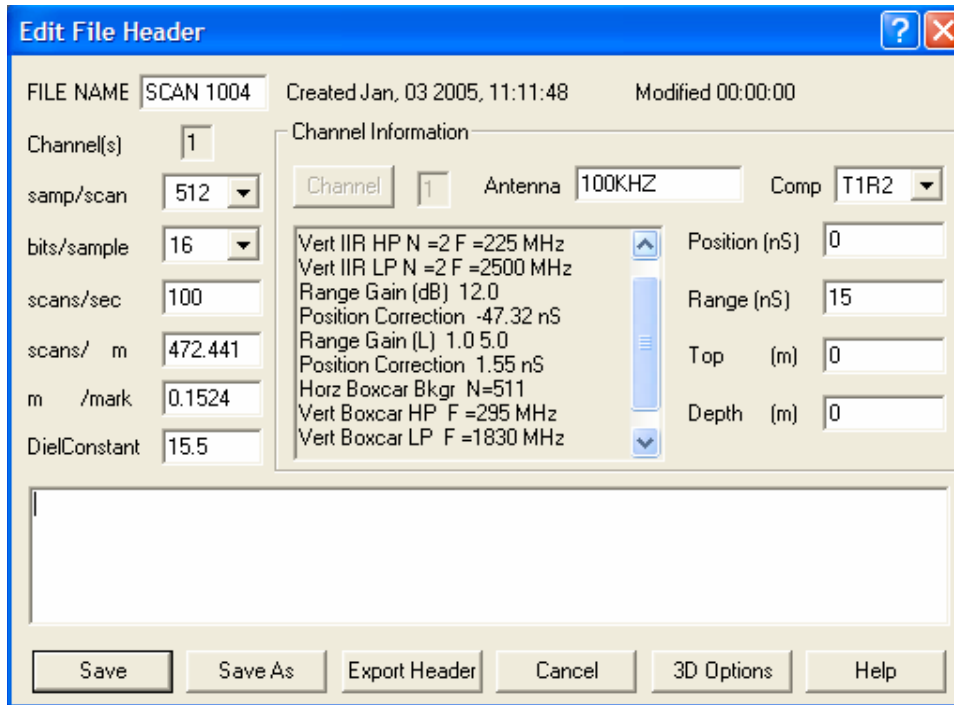


Figure 2-4 Sample Specifications of Data Processing Operations (Agrawal 2005)

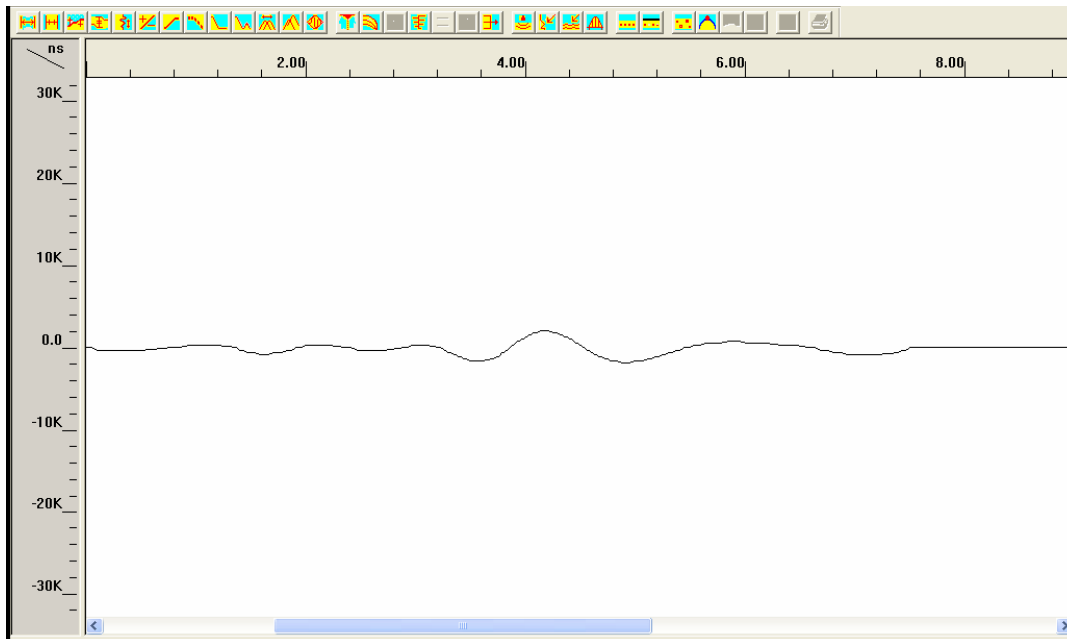


Figure 2-5 A typical radar signal (observed in oscilloscope mode) after applying range gain, zero correction, background removal and FIR low pass and high pass filters (Agrawal 2005)

### 2.3 Defect Detection in Log1 Using RADAN Software

This section describes the process of defect detection in logs using RADAN software that was carried out in a previous study by Agrawal (2005). The characteristics of Log1 are given in Table 2-1. The test setup is shown in Figure 2-6 which utilized two antennas simultaneously (one of the antennas is hidden).

Table 2-1 Characteristics of Log1 (Agrawal 2005)

Log no.	Species	Length (m)	Diameter (m)		Moisture Content (%)			Dielectric Constant
			Butt end	Small end	Surface	Max.	Average	
1	Yellow Poplar	2.45	0.35	0.33	16	36	24	12



Figure 2-6 Antenna configuration for Log1 (Agrawal 2005)

The actual radar signals were converted by the software to grey scale scans (Figure 2-7) where the intensity in the scan is a measure of signal amplitude with white bands for positive signal peaks and black bands for negative signal peaks (Agrawal 2005). The top set of white, black and white band indicates the transmit signal (antenna to log coupling). After the data was processed, all the internal defects were predicted based on location of the signal clutter (changes in signal bands) in the processed data shown in Figure 2-8. The log was sawed at positions where the processed data indicated changes in signal characteristics, which were attributed to presence of internal defects (Agrawal 2005). The cut circular cross sections are shown in Figure 2-9 where the circles in the center indicate rot initiation and the circles on the periphery indicate knots. There

was a good correlation between the changes observed in the signal characteristics in the GPR scan and the actual location of defects observed after sawing the log.

The predicted rot initiation was at about 0.16 m as seen from Figure 2-8 and the actual depth measured after sawing was 0.15 to 0.16 m. The knot (Defect 1) in Figure 2-9 (a) shows up in the GPR scan (Figure 2-8) at 0.26 m depth whereas the actual defect is at about 0.25 m. For knots on the side (Defect 2 in Figure 2-9(b)), the distance to the knot is at an angle, and the echoes from the two knots came successively. Reflection from the first knot (Defect 2) arrived at 0.18 m and merged into the reflection from rot initiation. Reflection from the second knot has merged into the bottom reflection, which makes it difficult to resolve the second knot from the GPR scan. The measured distances of the two knots in Figure 2-9(b) from the top antenna were 0.18 m and 0.29 m. For Defect 3, the predicted depth from the GPR scan (Figure 2-8) was 0.22 m where as the actual depth (Figure 2-9(c)) was 0.23 m. For Defect 4, the predicted depth from the GPR scan (Figure 2-8) was 0.2 m where as the actual depth (Figure 2-9(c)) was 0.23 m. The above results show that the resolution of depth prediction is 0.01 to 0.04 m. The resolution value is equal to approximately half the wavelength, which comes to 0.048 m for 900 MHz radar signal in a log with dielectric constant of 12. The positions of the defects along the length of the log were predicted within 0.05m (Agrawal 2005).

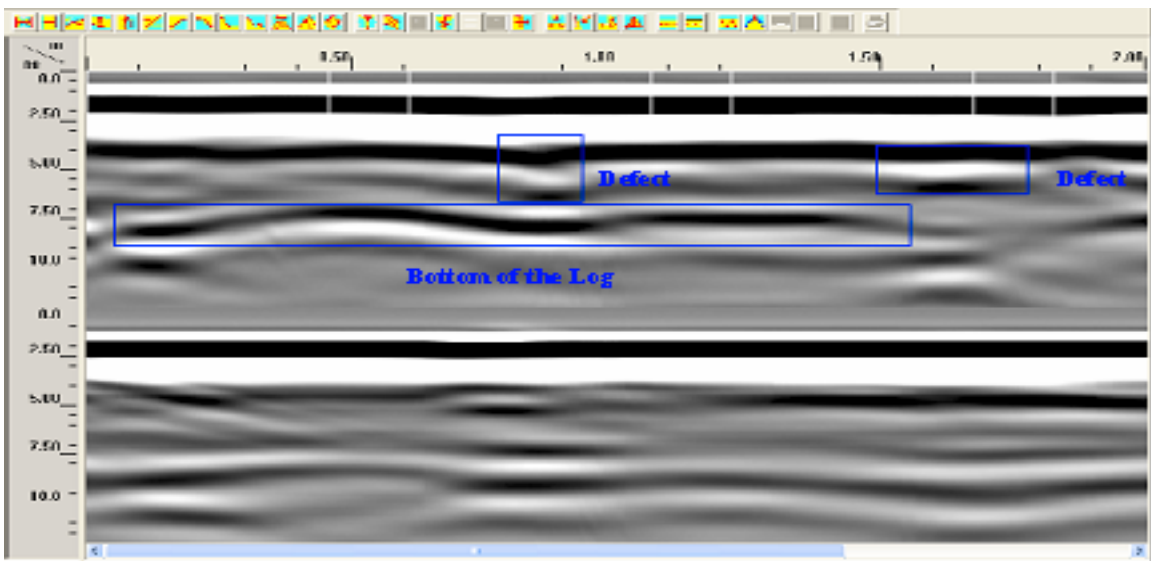


Figure 2-7 GPR Raw data of Log1, horizontal scale indicating the travel distance along the length of the log (m) and the vertical scale indicating travel time (ns) (Agrawal 2005)



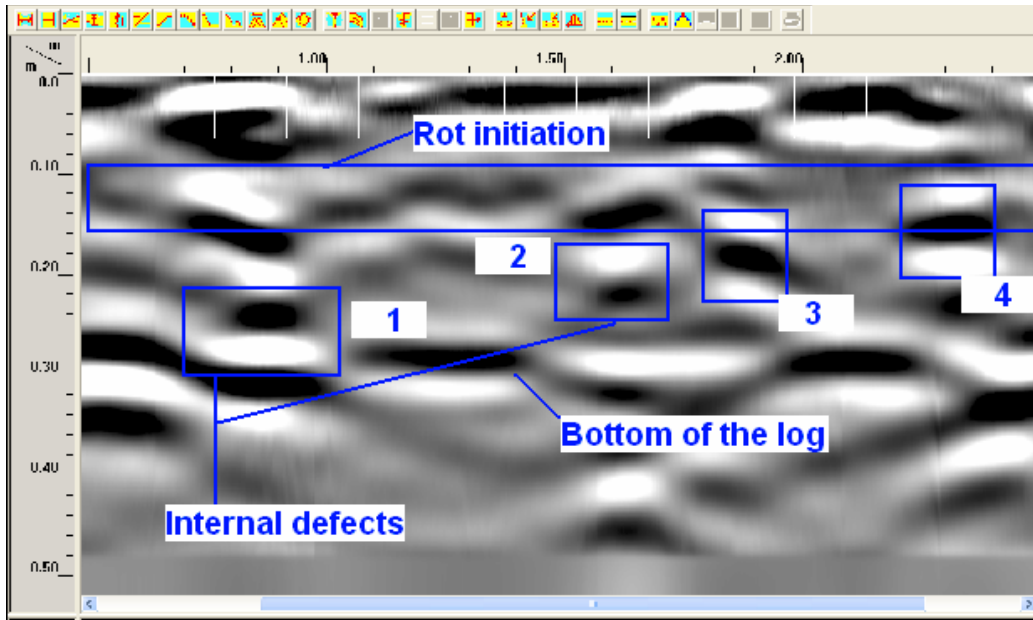


Figure 2-8 Processed GPR data of Log1 indicating the internal features (Agrawal 2005)



(a) Defect 1

(b) Defect 2

(c) Defect 3

(d) Defect 4

(Circles at the center indicate rot initiation and circles at the periphery indicate knots)



(e) Visible external knot

Figure 2-9 Cut section profile of Log1 indicating presence of defects (Agrawal 2005)

**Conclusion:** It can be seen that for the above data processing method a RADAN expert is required to process and analyze data. Also, the processed data is in the form of an image without precise numerical information about the location and size of the defects. This result cannot be used directly to generate manufacturing process instructions.

## Chapter 3

### DEFECT DETECTION USING MATLAB ALGORITHM

#### 3.1 Log1 Data Processing Using MATLAB

MATLAB is one of the sophisticated software packages available for image processing. It is built to handle very high graphics applications. It comprises of inbuilt tool boxes like Image processing, Signal processing, Neural Networks, Filter Design, Curve fitting, Statistics etc., that are highly useful in dealing with complicated real life problems. GPR Data has been processed using MATLAB algorithm and the processing involved application of the following major steps.

1. Linear Gain
2. Zero Correction
3. Clipping the signal
4. Signal Filtering
  - Low Pass Filter
  - High Pass Filter
5. Threshold

Figure 3-1 shows the MATLAB algorithm developed for radar data processing.

**Outline of MATLAB Algorithm:** MATLAB algorithm works on a desktop computer where MATLAB is available on the system. Radar data in the form of .dzt file from SIR system (Radar system) needs to be transferred into this desktop computer for processing. The other option is to process the radar data on the SIR system laptop itself. MATLAB needs to be present in the laptop to do this.

Radar data in the form of .dzt has to be converted into ASCII before processing. MATLAB algorithm loads this converted data file onto MATLAB. The scans in the data file will be realigned to match the RADAN output. Initial noise in the data will be removed. A grand average of the entire radar scan is calculated. Every value in the radar scan is subtracted by this grand average. 138 columns from both the sides of the scan are

copied and added onto the radar scan. Linear gain is applied to improve signal clarity. Zero Correction is done to ease the depth calculation of the defects.

Algorithm branches out into two options here. The 1<sup>st</sup> option is for seeing only internal defects and the 2<sup>nd</sup> option is to see both internal and surface defects. Options have been created to minimize the noise in the final output. More noise can be eliminated if interested only in seeing internal defects. In 1<sup>st</sup> option, the signal is clipped till 2<sup>nd</sup> maximum. In 2<sup>nd</sup> option, signal is clipped till 1<sup>st</sup> minimum. Reflection from the bottom of the log is removed if found. Edge effect is removed from the scan. Then, Low pass and High pass filters are applied. After this the program can go in two directions. If the remove reflections option is not clicked, then automatic threshold value is calculated. Automatic threshold is applied to detect the defects. There is also option for entering threshold value manually for defect detection. Location of the detected defects is converted into X-Y coordinates in meters. The Y coordinates conversion can take place only in two situations, either if the value of dielectric constant is known or the bottom of the log is found. If the remove reflections option is clicked, then an initial threshold is applied to separate the echoes in the scan. Then these echoes are removed leaving the first reflection. Small chunks leftover in the scan are removed further. Final threshold is applied to detect the defects. Location of the detected defects is converted into X-Y coordinates in meters.

Figure 3-2 shows a typical radar signal/scan. The first sinusoid is formed by the reflection of electromagnetic waves from the air/wood interface. The electromagnetic waves form a sinusoidal reflection when they pass through an interface. The strength of the reflection from an interface depends on the difference in the dielectric constants of the media forming the interface. The formation of the sinusoid and the difference in the strength of the reflected signal has been made use of in this research for detection of defects. Air/wood interface will have a stronger valley where as wood/air interface will have a stronger peak in the sinusoid. Figure 3-2 also shows initial noise in a radar signal. The initial noise occurs due to the self reflection of the radar waves just in front of the antenna. The initial noise will be present in the first three data points of the signal. Figure 3-2 also shows the location of the useful information in the radar signal.

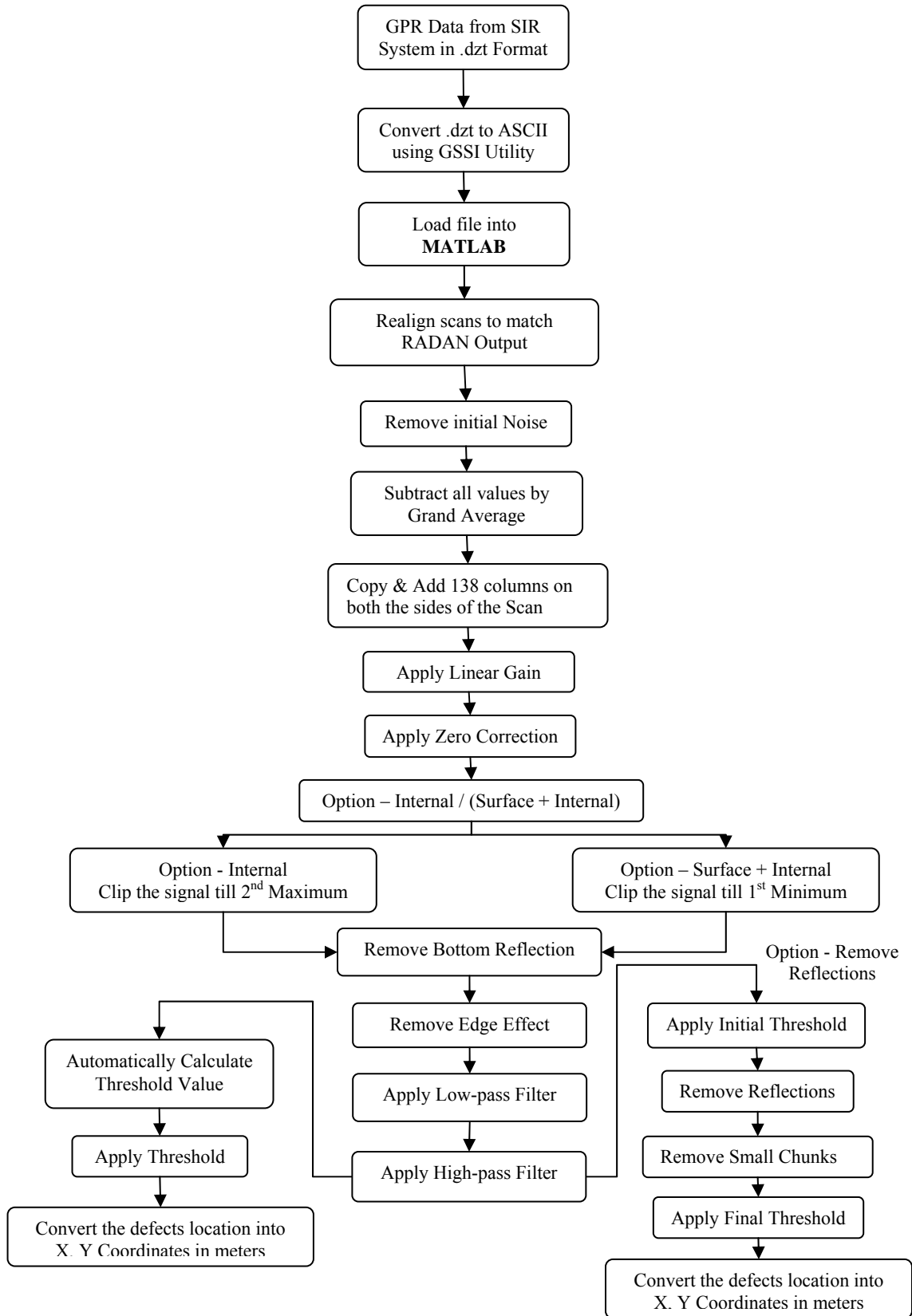


Figure 3-1 MATLAB Algorithm for Radar Data Processing

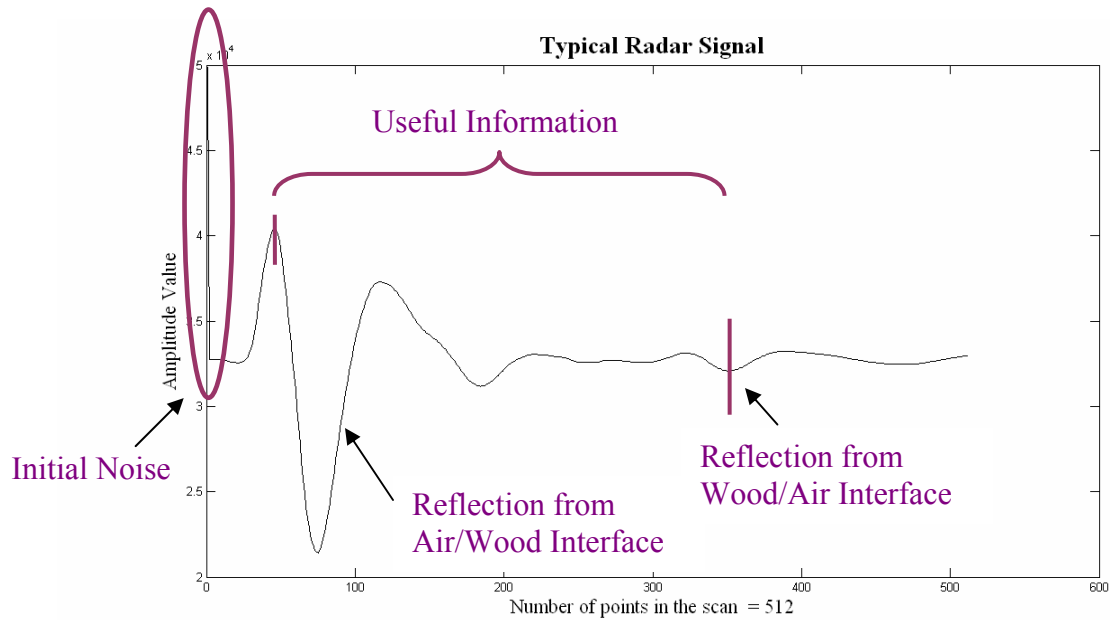


Figure 3-2 Typical Radar Signal showing reflections from interfaces, location of useful information and initial noise

The following assumptions have been made in building the algorithm. With reference to Figure 3-2,

1. The initial noise will be present only in 1<sup>st</sup> 3 rows of the signal
2. Reflection from Air/Wood Interface will always be present in first 200 points of the signal.
3. Reflection from Wood/Air interface will be present after 200<sup>th</sup> point in the radar signal after zero correction.
4. Reflection from Wood/Air interface is assumed to be present when the average amplitude value of its minimum is less than -1000.
5. The edge effect will be present in the first and last 50 scans of the data.

Figure 3-3 shows the surf plot of the raw data of Log1 obtained using MATLAB. The reflection from the air/wood interface is clearly visible and the other features are not prominent. It should be noted that this plot shows the travel time along the horizontal axis. Figure 3-4 shows the top view of Figure 3-3 with 90° clockwise rotation. Figure 3-4 is the typical representation of a radar scan. There are 1,311 individual signals in the entire plot. Individual signals are arranged next to each other in the longitudinal direction of the wooden log to form the entire scan. Each scan is done perpendicular to the length

of the log. Each scan has 512 data points. Both Figures 3-3 and 3-4 do not show the defect patterns. Individual scans in Figure 3-4 are starting from right to left where as the actual scan is done from left to right. This scan is reversed when compared to the output of RADAN software. This scan has been realigned in Figures 3-5 and 3-6. This realignment can be observed by comparing Figures 3-4 and 3-6.

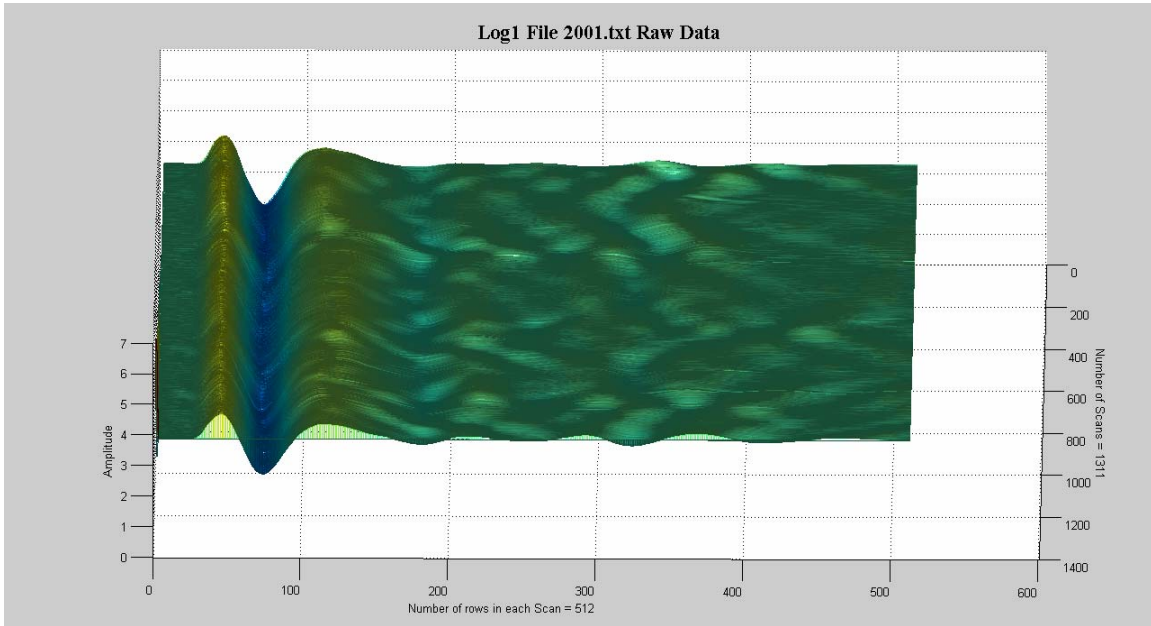


Figure 3-3 MATLAB 'Surf' plot of raw data of Log1

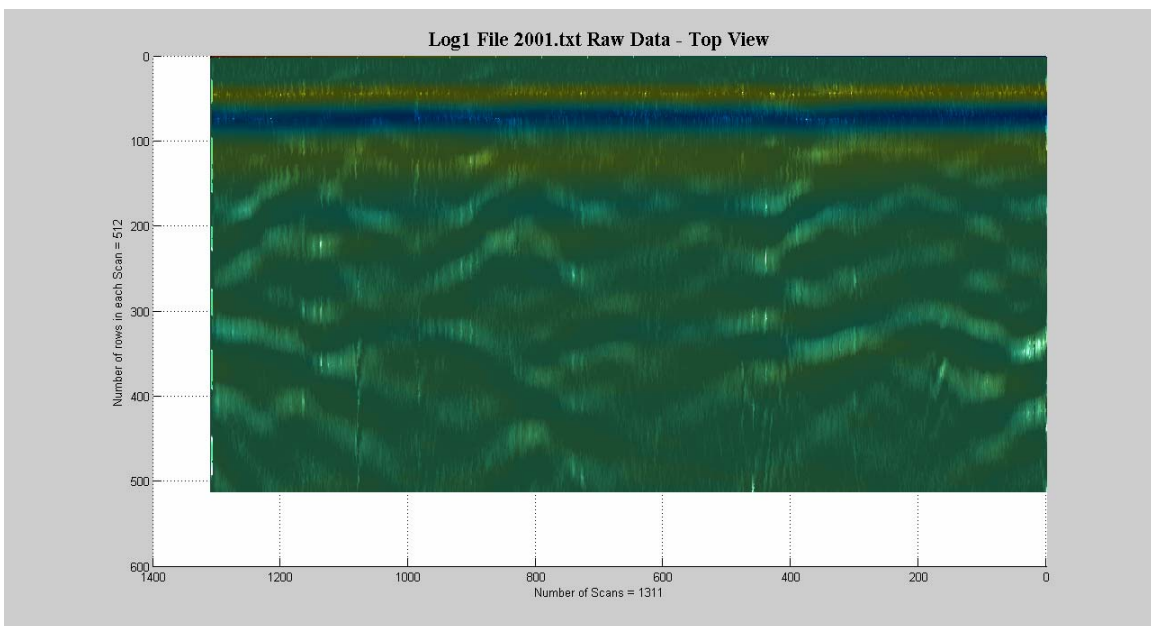


Figure 3-4 Top view of Figure 3-3 with 90° clockwise rotation

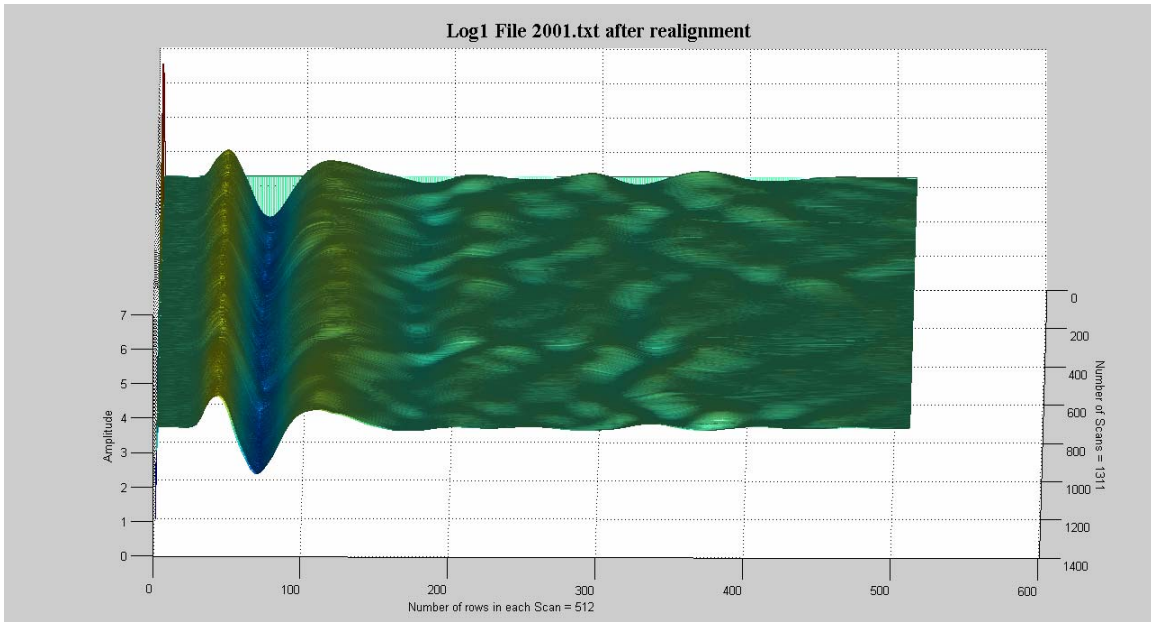


Figure 3-5 MATLAB ‘Surf’ plot of Log1 raw data after realignment

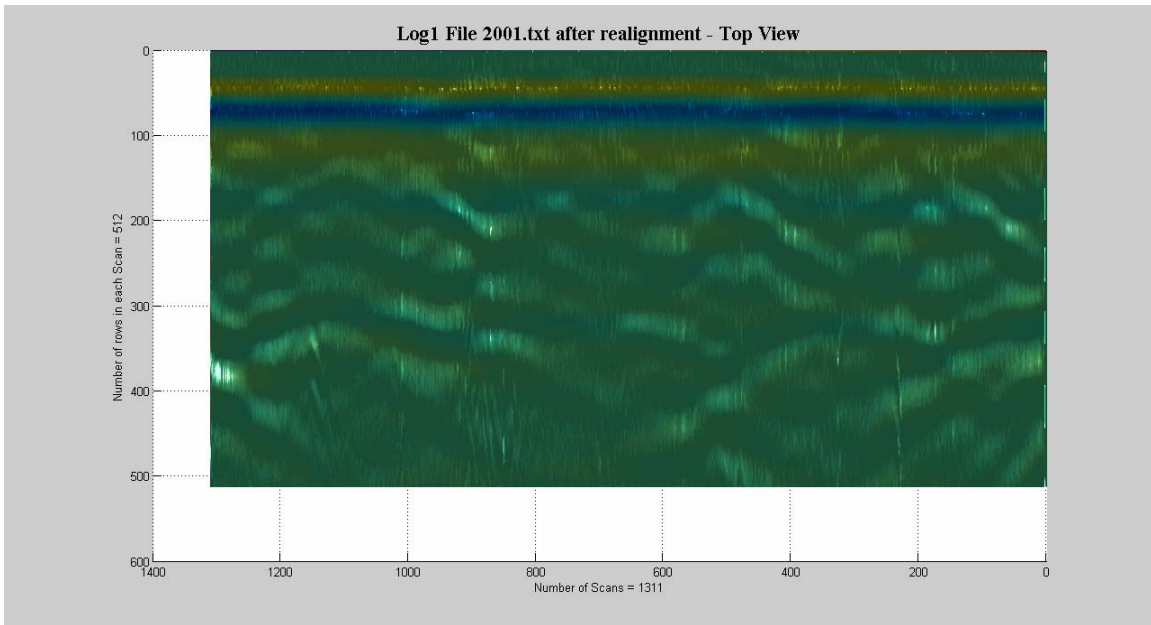


Figure 3-6 Top view of Figure 3-5 with 90° clockwise rotation

By setting the values of the first three data points of all the signals to 32,750, the initial noise in the signal can be removed. The value 32,750 is an approximate average amplitude value of the radar signal. Radar scan becomes clearer after removing the initial noise from the signal and patterns start appearing as seen in Figures 3-7 and 3-8.



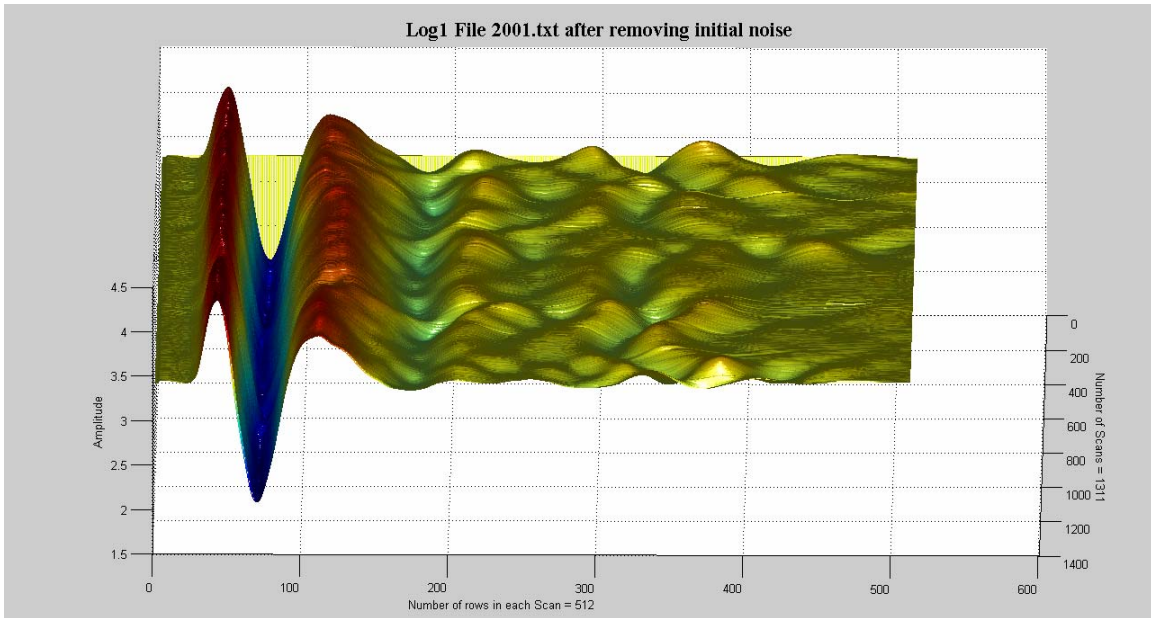


Figure 3-7 Surf Plot of Log1 after removing initial noise

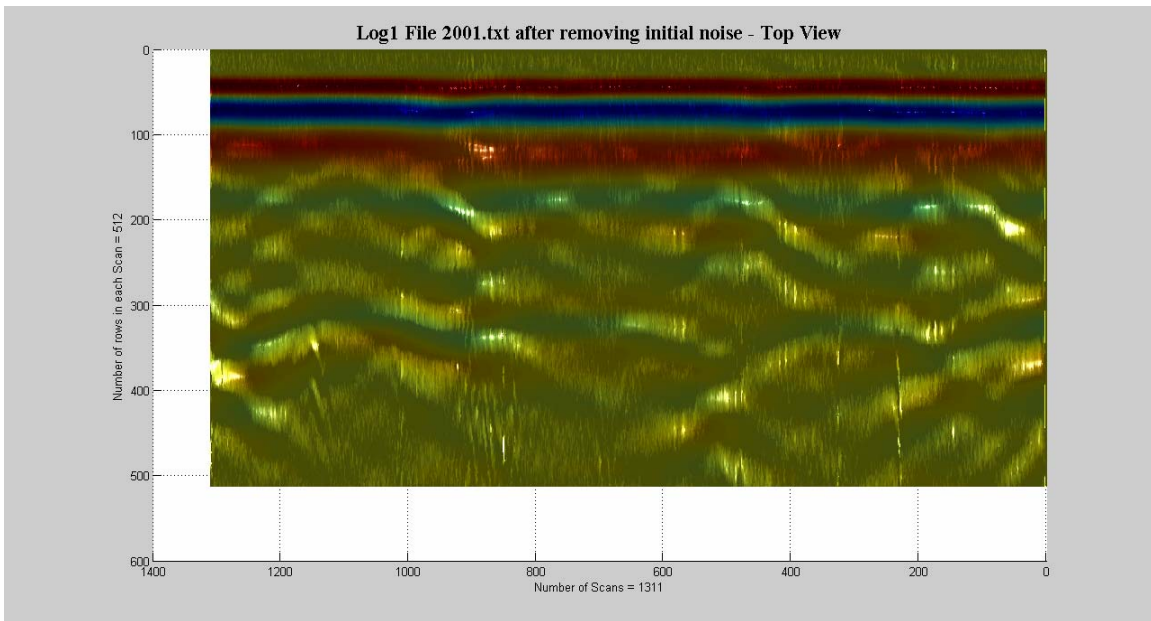


Figure 3-8 Top view of Figure 3-7 with 90° clockwise rotation

The radar scan loses columns equal to the length of the filter plus one after filtering. The length of the Low pass filter is 17 and the length of the High pass filter is 257 which add up to 274. Adding two to this number gives 276 and half of 276 is 138. So, In order to make up for these columns, 138 columns have been copied and added on

both the sides of the scan. Figures 3-9 and 3-10 shows the radar scan after adding these columns.

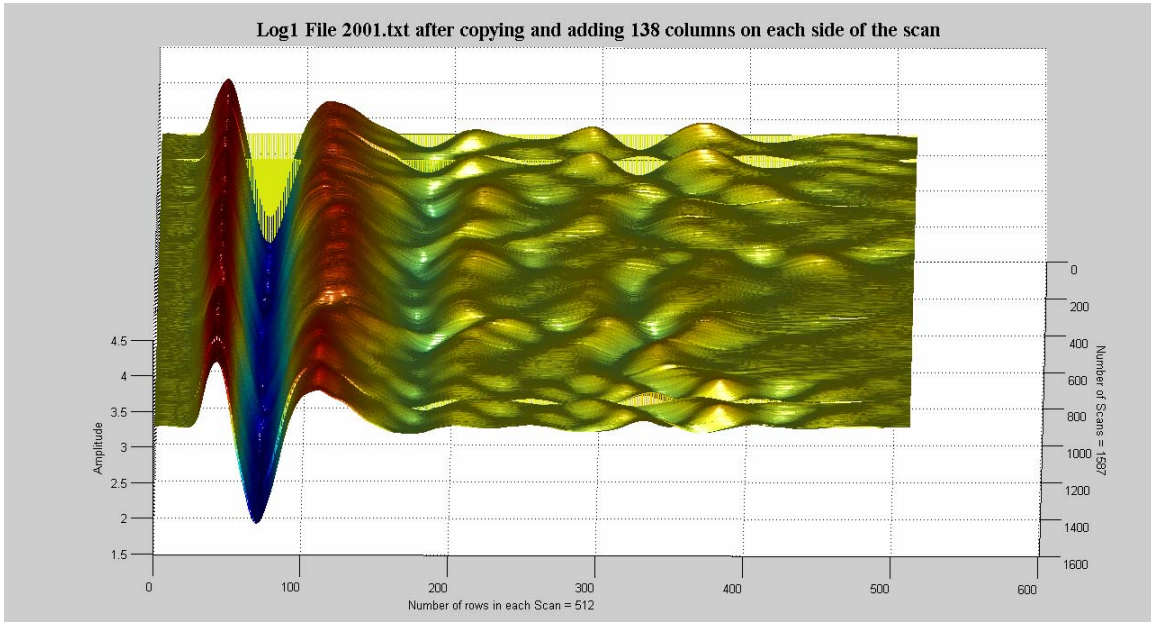


Figure 3-9 Surf Plot of Log1 after adding 138 columns on each side of the scan

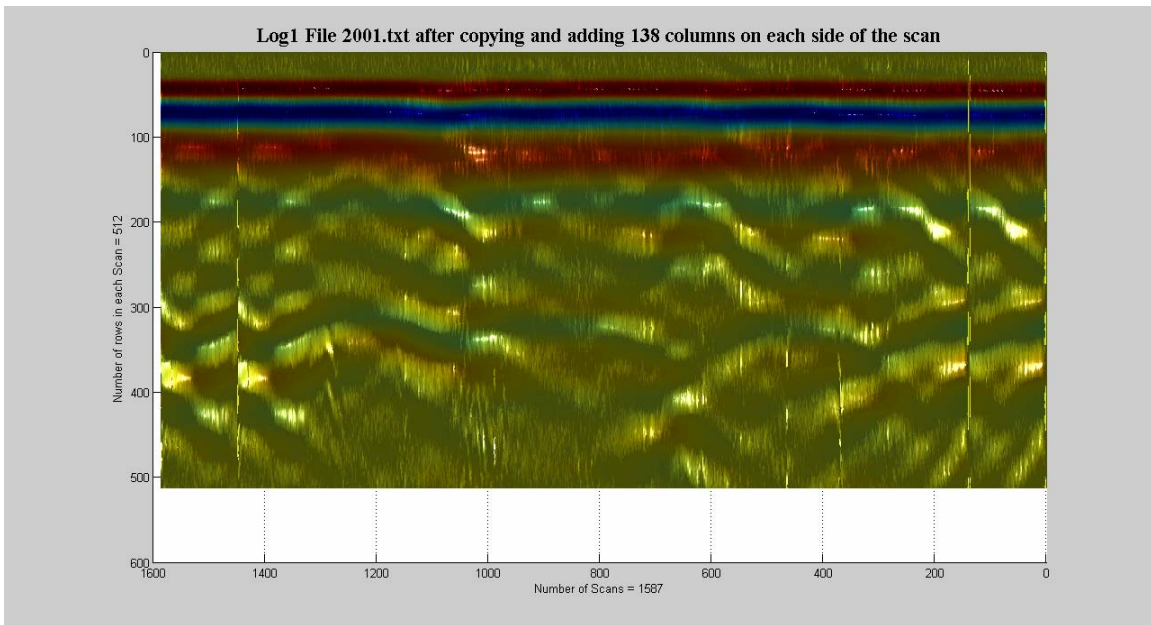


Figure 3-10 Top view of Figure 3-9 with 90° clockwise rotation

### 3.2 Linear Gain

Radar signals are prone to attenuation with increasing depth. Because of attenuation, the information at greater depths is not as clear and as reliable as the data near the surface of the antenna. A linear gain increasing with time (or depth) has been used to compensate for amplitude reduction with depth. Figure 3-11 shows the signal before and after applying linear gain. The echo from the bottom of the log and some of the internal features became more obvious after the gain was applied. The gain ranged from 1 at the beginning of the signal to 1.08 at the end of the signal for the length of 512 points.

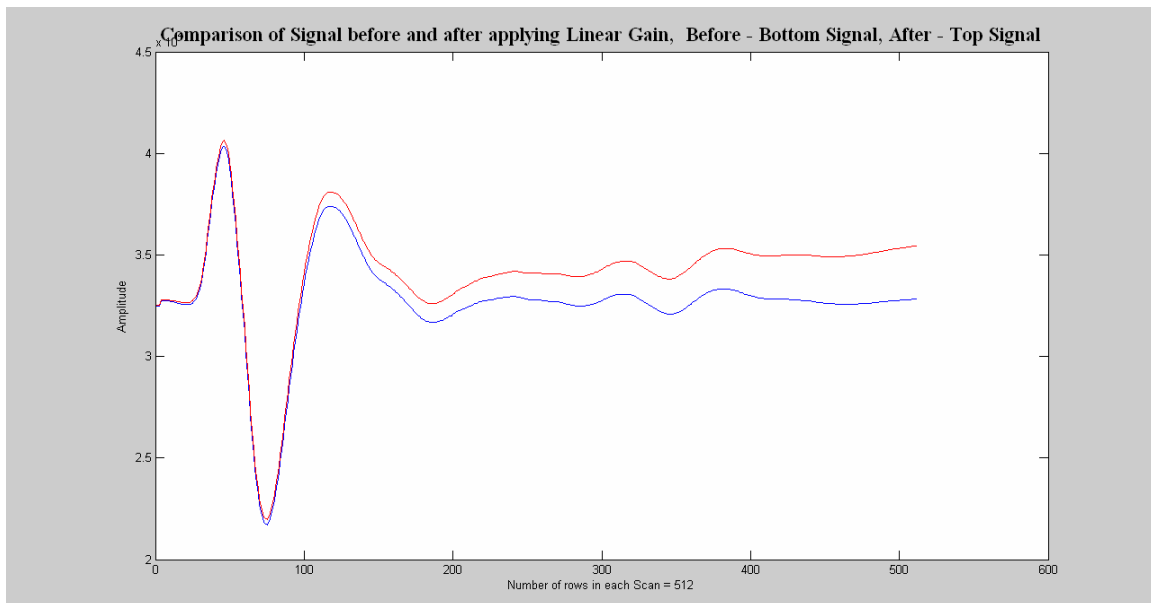


Figure 3-11 Comparison of radar signal before and after applying linear gain

### 3.3 Zero Correction

Zero correction is a process that is used to vertically adjust the position of the whole GPR scan so that the depth can be measured with respect to the wood surface. This correction involves deleting the signal till the first positive peak (i.e., reflection from the air/log interface) which then corresponds to the wood surface. This process allows estimation of the correct depth of any observed feature after the correct dielectric constant is used. Figure 3-12 shows the radar scan after applying zero correction. To apply zero correction, the average of positions of all the signals maximum will be found and all the data points till that average maximum position will be deleted from the scan.

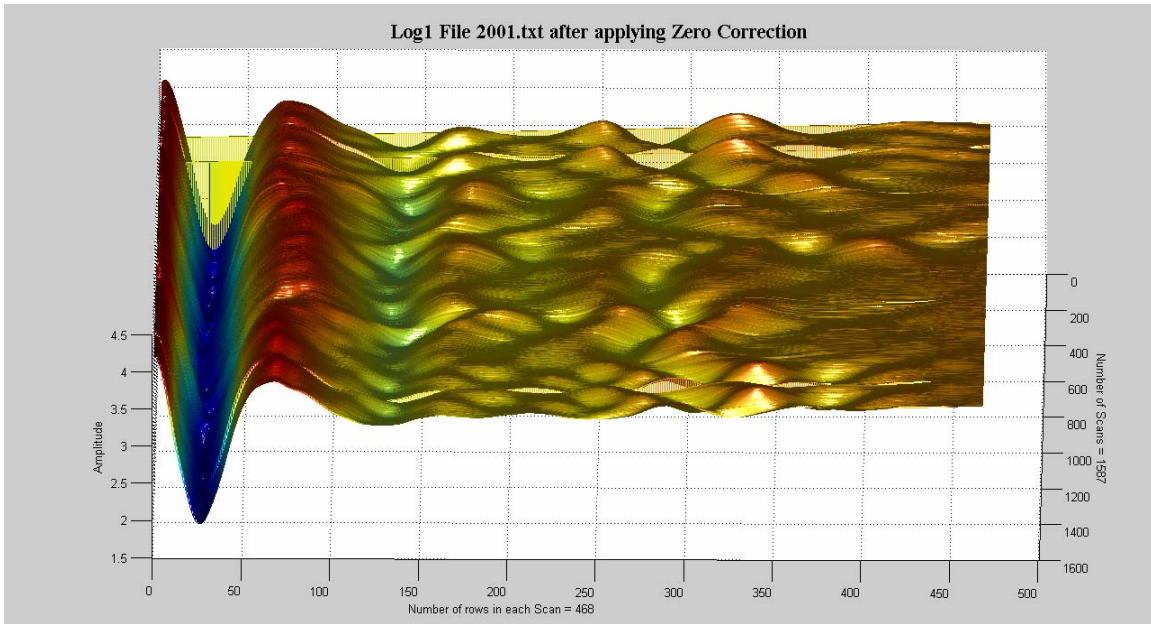


Figure 3-12 Radar scan after applying zero correction

Figure 3-13 also shows the radar scan after applying zero correction. But this zero correction is different from previous zero correction. To apply this zero correction, the position of each signal maximum will be found and all the data points till that signal maximum position will be deleted. So, this zero correction results in aligning all the signal maximums (peaks) in line.

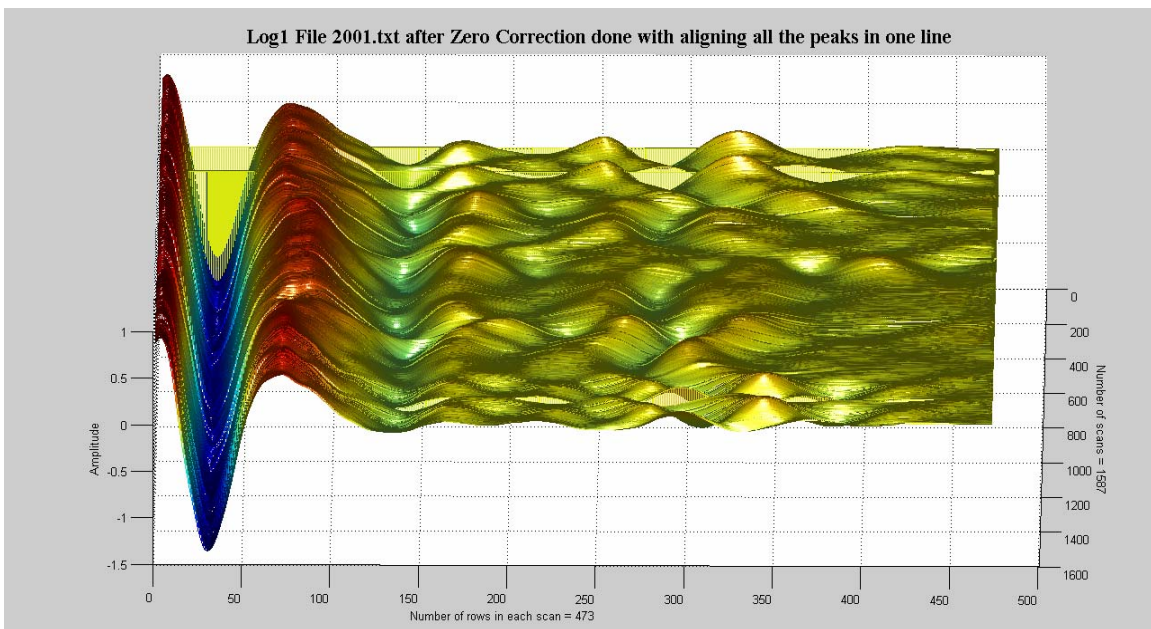


Figure 3-13 Radar scan after applying zero correction with aligning all the peaks in line

### 3.4 Clipping the Signal

Signal clipping is required to eliminate noise from the signal. Signal can be clipped till any point based on our application. Signal clipping is to make the value of the signal equal to zero. To see only internal defects in the scan, the signal can be clipped till 2<sup>nd</sup> maximum (with respect to Figure 3-2) and the noise caused due to air / wood interface reflection can be eliminated to a greater extent. To see the surface defects also, the signal has to be clipped only till 1<sup>st</sup> minimum (with respect to Figure 3-2).

For clipping till 2<sup>nd</sup> maximum, first the average of the entire scan has to be brought to zero. This is done by subtracting the entire scan by the whole average. Whole average is the average of all the data points in the entire scan. Average of around  $1587 \times 468 = 742,716$  data points (numbers). Then, the minimum with in the first 200 points of each column will be found in the signal and the signal will be clipped till the first positive number to the right of this minimum. The next step is to find the positions of maximums with in the first 200 points of each column and take the average of these positions. Then clip the signal till this average. Figure 3-14 and 3-15 shows the radar scan of Log1 after clipping the signal till the 2<sup>nd</sup> maximum. Defect patterns appear clearer in this scan compared to previous radar scans.

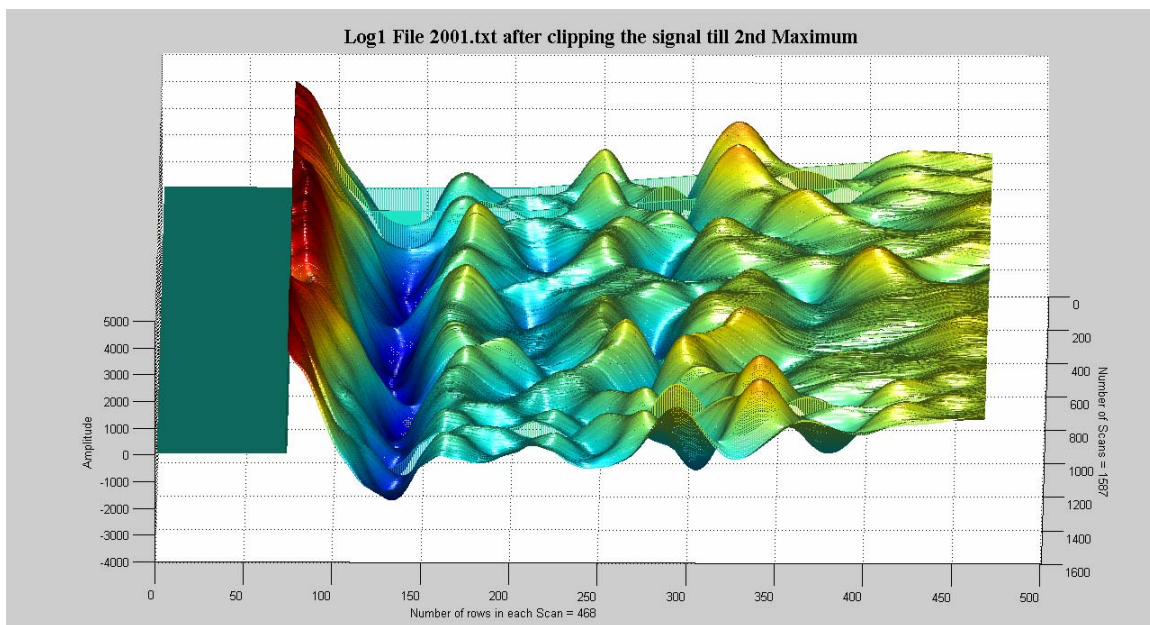


Figure 3-14 Radar scan of Log1 after clipping till the 2<sup>nd</sup> Maximum

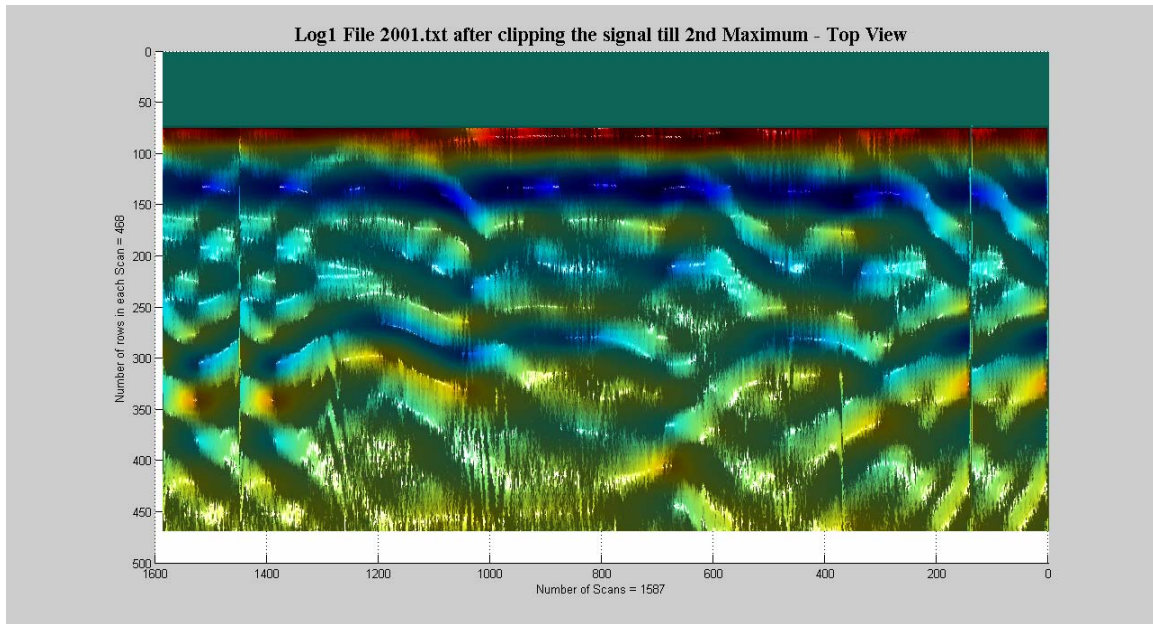


Figure 3-15 Top view of Figure 3-14 with 90° clockwise rotation

### 3.5 Removing Bottom Reflection (Reflection from Wood/Air Interface)

Removing reflection from bottom of the log eliminates unwanted noise from the signal. A good bottom reflection will have the lowest amplitude value after 200<sup>th</sup> point in the zero correction applied signal. Also, a significant bottom reflection will have the minimum amplitude value lower than -1000. To remove bottom reflection, first the minimum amplitude value after 250<sup>th</sup> point in each signal is found. An average of all these minimum amplitude values is found. Then, this average minimum amplitude value is compared with -1000 value. If this amplitude value turns out to be lower than -1000, then the average of these minimum amplitude value positions is found. Then the signal is clipped from this average minimum amplitude value position to the end. Figure 3-16 and 3-17 shows radar scan after removing bottom reflection.

### 3.6 Signal Filtering

The radar scan in Figure 3-15 is not showing the correct defect patterns. It is superimposed with unwanted noise. The noise in the signal comes from the unwanted frequencies. The unwanted frequencies lie below 300 MHz and above 1800 MHz

(Agrawal 2005). This noise has to be filtered out from the signal to get the useful information. Noise filtering can be done by using filters.

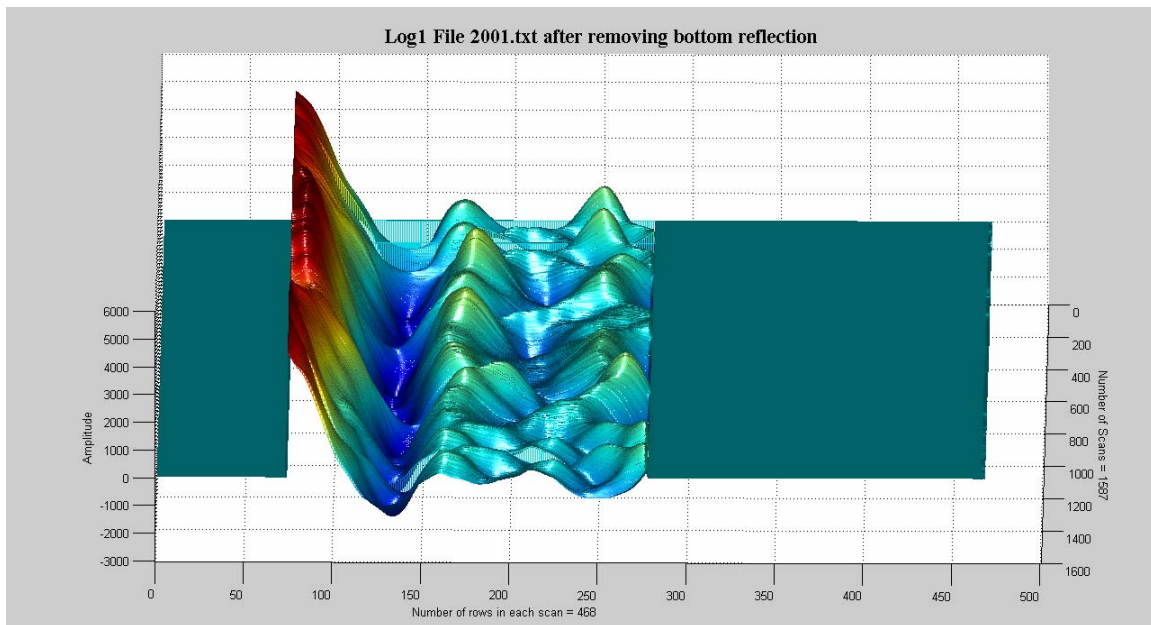


Figure 3-16 Radar Scan after removing Bottom Reflection

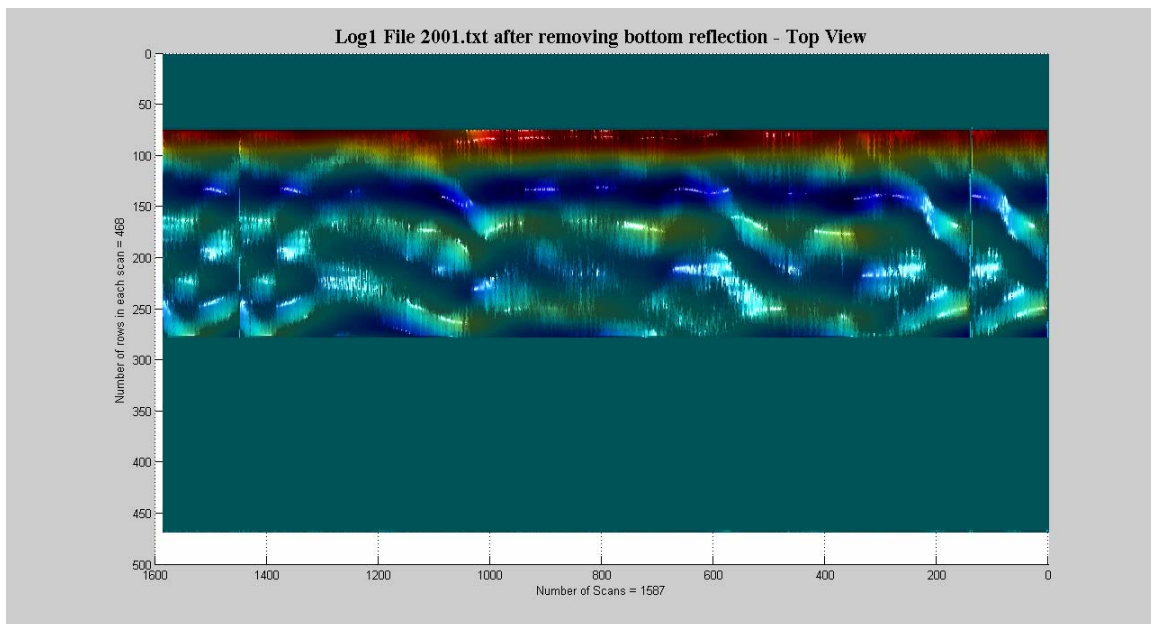


Figure 3-17 Top view of Figure 3-16 with 90° clockwise rotation

### 3.7 Filters

Filters are signal conditioners. Filters function by accepting an input signal, blocking pre-specified frequency components and passing the original signal minus the

filtered components to the output. For example, a typical phone line acts as a filter that limits frequencies to a range considerably smaller than the range of frequencies human beings can hear. This is the reason why listening to music over the phone is not as pleasing to the ear as listening to it directly.

A digital filter takes a digital input, gives a digital output and consists of digital components. In a typical digital filtering application, software running on a digital signal processor (DSP) reads input samples from an Analog to Digital (A/D) converter, performs the mathematical manipulations dictated by theory for the required filter type and outputs the result via a Digital to Analog converter (D/A). In this research, Finite Impulse response (FIR) digital filter has been used.

An analog filter by contrast operates directly on the analog inputs and is built entirely with analog components such as resistors, capacitors and inductors.

There are many types of filters, but the most common are low-pass, high-pass, band pass, and band stop. A low pass filter (LPF) allows only low frequency signals (below some specified cutoff) to its output, so it can be used to eliminate high frequencies. A low pass filter is handy in that regard for limiting the uppermost range of frequencies in an audio signal; it's the type of filter that a phone line resembles.

A high pass filter just does the opposite by rejecting only frequency components below some threshold. An example of high pass application is cutting out the audible 60Hz AC power "hum", which can be picked up as noise accompanying almost any signal in the U.S.

The designer of a cell phone or any other sort of wireless transmitter would typically place an analog band pass filter in its output RF stage to ensure that only output signals within its narrow, government-authorized range of the frequency spectrum are transmitted. Engineers can use band stop filters which pass both low and high frequencies and blocks a predefined range of frequencies in the middle.



### 3.8 Window Design Techniques for designing filters (Vinay & John 2004)

The basic idea behind the window design is to choose a proper ideal frequency-selective filter (which always has a non causal, infinite-duration impulse response) and then truncate (or window) its impulse response to obtain a linear-phase and causal Finite Impulse Response (FIR) filter. Because the impulse response required to implement the ideal filter is infinitely long, it is not possible to design an ideal FIR filter. Finite-length approximations of the ideal impulse response lead to ripples in the pass-band and the stop-band, as well as a nonzero transition width between the pass-band and the stop-band (Figure 3-18).

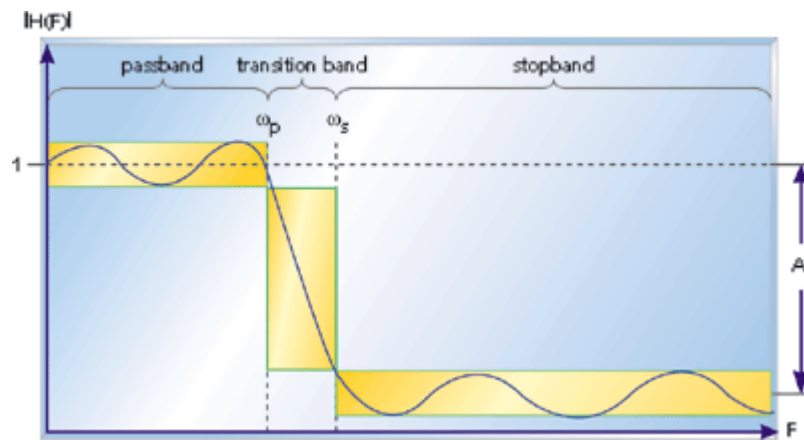


Figure 3-18 The response of a low-pass filter to various input frequencies

The emphasis in this method is on selecting an appropriate windowing function and an appropriate ideal filter. We will denote an ideal frequency-selective filter by  $H_d(e^{j\omega})$ , which has a unity magnitude gain and linear-phase characteristics over its pass-band, and zero response over its stop-band. An ideal LPF of bandwidth  $\omega_c < \pi$  is given by

$$H_d(e^{j\omega}) = \begin{cases} 1.e^{-j\alpha\omega}, & 0 \leq |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases} \quad (3.1)$$

Where  $\omega_c$  is also called the *cutoff* frequency and  $\alpha$  is called the sample *delay* (note that from the DTFT properties,  $e^{-j\alpha\omega}$  implies shift in the positive  $n$  direction or delay). The impulse response of this filter is of infinite duration and is given by

$$h_d(n) = F^{-1} \left[ H_d(e^{j\omega}) \right] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega$$

$$h_d(n) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} 1 \cdot e^{-j\alpha\omega} e^{j\omega n} d\omega$$

$$h_d(n) = \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}, \quad -\infty < n < \infty \quad (3.2)$$

Note that  $h_d(n)$  is symmetric with respect to  $\alpha$ , a fact useful for linear-phase FIR filters. To obtain an FIR filter from  $h_d(n)$ , one has to truncate  $h_d(n)$  on both sides. To obtain a causal and linear-phase FIR filter  $h(n)$  of length  $M$ , we must have

$$h(n) = \begin{cases} h_d(n), & 0 \leq n \leq M-1 \\ 0, & \text{elsewhere} \end{cases} \quad \text{and} \quad \alpha = \frac{M-1}{2} \quad (3.3)$$

This operation is called “windowing”. In general,  $h(n)$  can be thought of as being formed by the product of  $h_d(n)$  and a window function  $w(n)$  as follows

$$h(n) = h_d(n)w(n) \quad (3.4)$$

where,

$$w(n) = \begin{cases} \text{some symmetric function with respect to } \alpha \text{ over } 0 \leq n \leq M-1 \\ 0, \text{ otherwise} \end{cases}$$

Depending on how we define  $w(n)$  above, we obtain different window designs. For example, in (3.3) above

$$w(n) = \begin{cases} 1, & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} = R_M(n) \quad (3.5)$$

which is the *rectangular window*

*Convolution:* A convolution is an integral that expresses the amount of overlap of one function say ‘H’ as it is shifted over another function say ‘W’. It therefore “blends” one function with another.

In the frequency domain the causal FIR filter response  $H(e^{j\omega})$  is given by the periodic convolution of  $H_d(e^{j\omega})$  and the window response  $W(e^{j\omega})$ ; that is,

$$H(e^{j\omega}) = H_d(e^{j\omega}) \otimes W(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} W(e^{j\lambda}) H_d(e^{j(\omega-\lambda)}) d\lambda \quad (3.6)$$

This is shown pictorially in Figure 3-19 for a typical window response, from which we have the following observations:

1. Since the window  $w(n)$  has a finite length equal to  $M$ , its response has a *peaky main lobe* whose width is proportional to  $1/M$ , and has side lobes of smaller heights.
2. The periodic convolution (3.6) produces a smeared version of the ideal response  $H_d(e^{j\omega})$ .
3. The main lobe produces a transition band in  $H(e^{j\omega})$  whose width is responsible for the transition width. This width is then proportional to  $1/M$ . The wider the main lobe, the wider will be the transition width.
4. The side lobes produce ripples that have similar shapes in both the pass-band and stop-band.

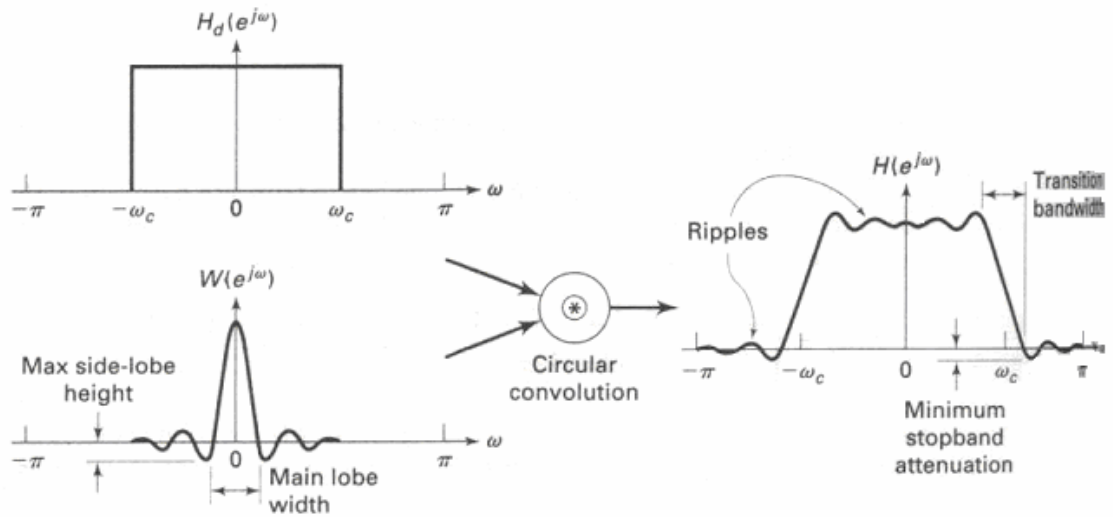


Figure 3-19 Windowing operation in the frequency domain

### 3.9 Rectangular Window (Vinay & John 2004)

This is the simplest window function. Rectangular window averaging has, historically, been the method of choice for smoothing transient waveforms. Rectangular window averaging adds  $N$  adjacent samples, divides the sum by  $N$ , and then writes that values into the  $N$ th sample location. This technique is basically a finite impulse response (FIR) low-pass digital filter of  $N$  taps with uniform weighting. It is defined by

$$w(n) = \begin{cases} 1, & 0 \leq n \leq M-1 \\ 0, & \text{otherwise} \end{cases} \quad (3.7)$$

Its frequency response function is

$$W(e^{j\omega}) = \left[ \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \right] e^{-j\omega \frac{M-1}{2}} \quad (3.8)$$

Amplitude response is,

$$W_r(\omega) = \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

From (3.6) the accumulated amplitude response  $H_r(\omega)$  is given by,

$$H_r(\omega) \square \frac{1}{2\pi} \int_{-\pi}^{\omega+\omega_c} W_r(\lambda) d\lambda = \frac{1}{2\pi} \int_{-\pi}^{\omega+\omega_c} \frac{\sin\left(\frac{\omega M}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} d\lambda, \quad M \square 1 \quad (3.9)$$

This implies that the running integral of the window amplitude response (or *accumulated* amplitude response) is necessary in the accurate analysis of the transition bandwidth and the stop-band attenuation. Figure 3-20 shows the rectangular window function  $w(n)$ , its amplitude response  $W(\omega)$ , the amplitude response in dB, and the accumulated amplitude response (3.9) in dB.

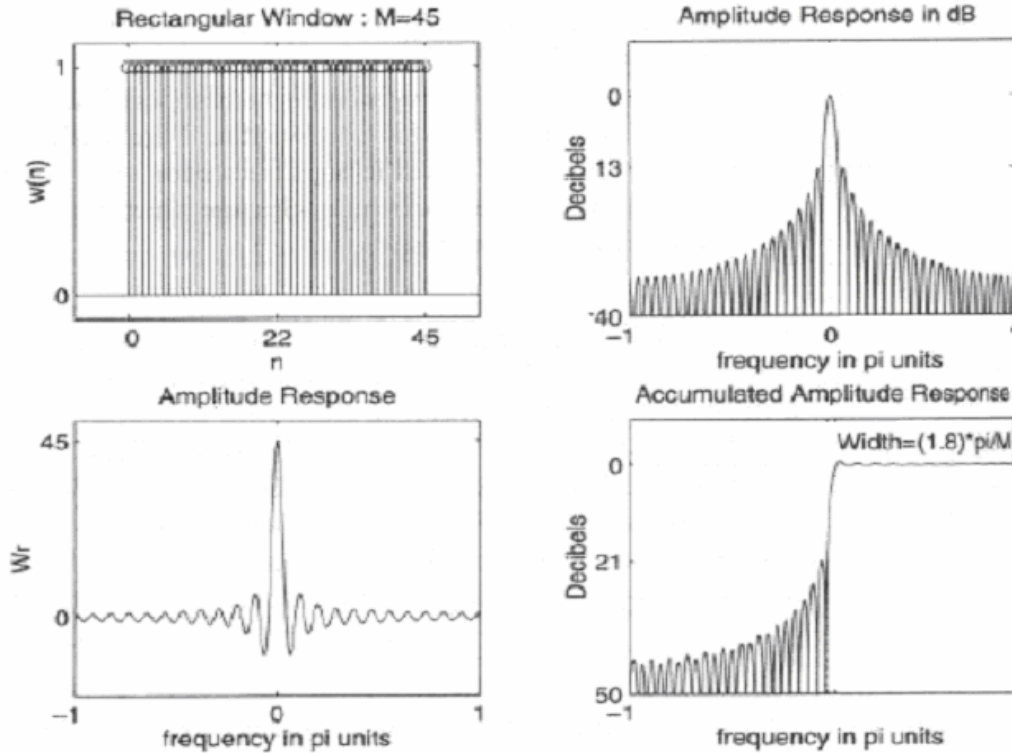


Figure 3-20 Amplitude response of rectangular window for  $M = 45$

### 3.10 High Pass Rectangular Filter

This filter applies a high pass filter i.e., it filters the data by suppressing low frequency data and allows high frequency data to pass through. This is done by convolving an  $N \times M$  rectangular window through the data, where  $N$  and  $M$  are odd integers. The average of the rectangular window is subtracted from the middle pixel. The result of the subtraction is the high frequency information. Thus,  $output(i,j) = input(i,j) - average(i,j,N,M)$  where  $i$  and  $j$  are the sample line position in the data,  $N$  and  $M$  are the size of the rectangular window, and  $average(i,j,N,M)$  is the average of the  $N \times M$  centered at  $i,j$ . A small rectangular window (e.g,  $3 \times 3$ ) will allow fine details to pass thru and will significantly suppress the low frequency. A large rectangular window (e.g,  $101 \times 101$ ) will enhance larger features and allow more low frequency information to pass through. In general, features which are half the size of the rectangular window will pass through the filter.

### 3.11 Designing Filters for Applications

Designing filters for applications is the process of selecting the filter's length and coefficients. The goal is to set those parameters such that certain desired stop-band and pass-band parameters will result from running the filter. Most engineers utilize a program such as MATLAB to do this. Filter Design and Analysis tool (FDATool) in MATLAB is one of the best tools available for doing filter design. But whatever tool is used, the results of the design effort should be the same:

1. A frequency response plot, like the one shown in Figure 3-18, which verifies that the filter meets the desired specifications, including ripple and transition bandwidth.
2. The filter's length/order or equivalently the length of the truncated impulse response and coefficients
3. The transition width and tolerance on pass-band and stop-band ripples
4. The Pass band ripple and the stop band attenuation.

The longer the filter (more taps) the more finely the response can be tuned. FIR filter simply produces a weighted average of its  $N$  most recent input samples. All of the magic is in the coefficients, which dictate the actual output for a given pattern of input samples.

### 3.12 Filters used in this research

In this research, low pass and high pass rectangular filters have been used for filtering the data. Filter Design and Analysis tool of MATLAB has been used for designing the filter. The following parameters were entered in the FDATool for designing the *Low pass filter*.

1. Filter type = Low Pass
2. Design method = FIR Window
3. Filter Order = 16
4. Window = Rectangular
5. Frequency specifications Units = Normalized (0 to 1)
6. Cutoff frequency,  $\omega_c = 0.13 \pi$  radians / sample

For the above inputs the impulse response of the designed filter is shown in Figure 3-21.

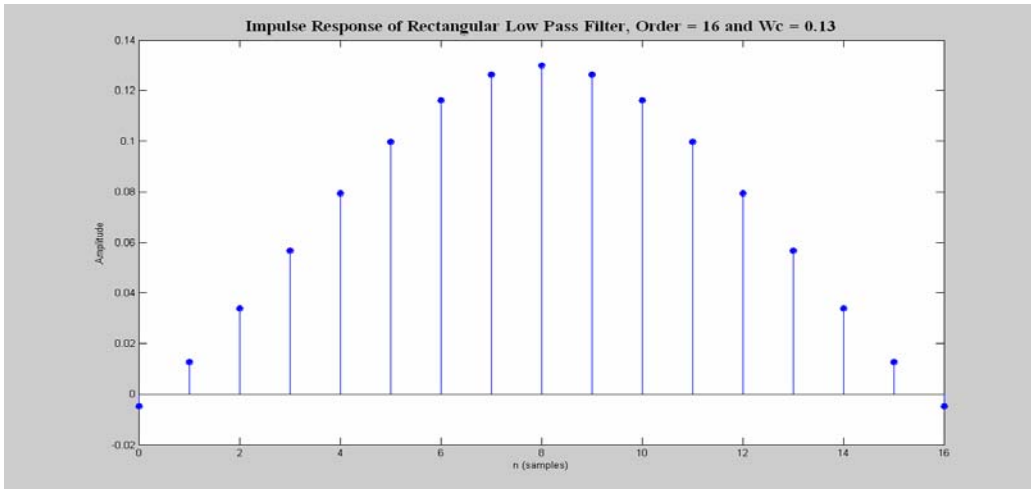


Figure 3-21 Impulse response of Rectangular Low Pass Filter, Order = 16 and  $\omega_c = 0.13$

Impulse response shows that the designed filter is symmetric about the point 8 and the number of points in the filter is odd i.e.,  $M = 17$ . Coefficients of the designed low pass filter are shown in Table 3-1.

Table 3-1 Coefficients of the designed low pass filter

1	-0.0049869	5	0.07942	9	0.13	13	0.07942	17	-0.0049869
2	0.012687	6	0.099831	10	0.12642	14	0.056723		
3	0.033816	7	0.11602	11	0.11602	15	0.033816		
4	0.056723	8	0.12642	12	0.099831	16	0.012687		

Figure 3-22 shows the magnitude and phase response of the designed filter.

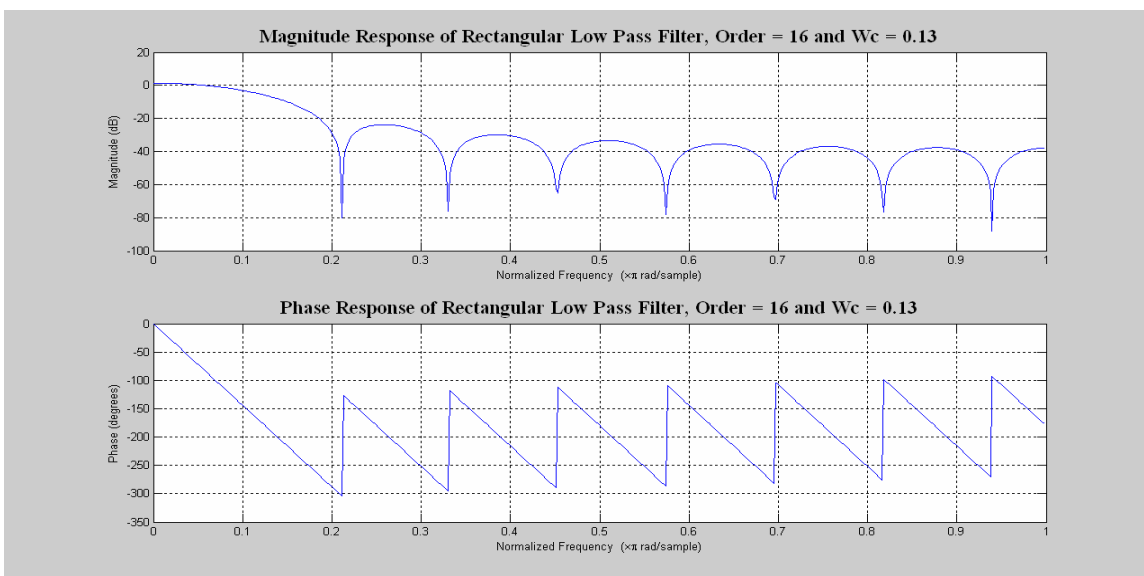


Figure 3-22 Magnitude and Phase response of Rectangular Low Pass Filter

Magnitude response shows the response of the signal at different frequencies. From this response the cutoff frequency  $\omega_c$  is around  $0.08 \pi$  radians / sample. Even though the cutoff frequency entered for the filter design was  $0.13 \pi$  radians / sample, due to the transition band the actual cutoff frequency has shifted to around  $0.08 \pi$  radians / sample. The attenuation of the signal in the stop band region is above 20 dB. The transition band width ranges from  $0.8 \pi$  radians / sample to  $1.8 \pi$  radians / sample. The cutoff frequency is in digital normalized angular frequency units. The method to obtain digital normalized angular frequency from Hz is explained below.

For Log1, Sampling time = 12ns

i.e., 512 samples or data points are collected in 12ns

Time between samples (Time to take one sample),  $T_s = \frac{12}{512} = 0.0234375 \text{ ns}$

For low-pass filter, desired cutoff frequency  $f = 1,800 \text{ MHz}$  (Agrawal 2005),

Angular Frequency =  $\Omega = 2\pi f$  (3.10)

$$= 2 \times 3.14 \times 1,800 \times 10^6 \text{ Hz} = 11,310 \times 10^6 \text{ Hz}$$

Digital (normalized angular) frequency =  $\omega = \frac{\Omega}{F_s} = \Omega T_s$  (3.11)  $\left( \text{Since } F_s = \frac{1}{T_s} \right)$

$$= 11,310 \times 10^6 \times 0.0234375 \times 10^{-9}$$

$$= 0.26508 \text{ radians/sample}$$

$$\text{or} = 0.08437 \pi \text{ radians / sample}$$

From this frequency calculation, it can be observed that the cutoff frequency obtained from the magnitude response Figure 3-22 matches with the desired cutoff frequency.

High pass filter used in this research is a standard rectangular filter of order 256. 256 was selected as the filter order since it is half the total number of elements in each scan i.e., 512. Filter is designed by defining a column matrix of 257 negative ones divided by the filter order 256. Then set the value of the middle element equal to one. The value of the filter coefficients except the middle one is -0.00391.



Figure 3-23 shows the impulse response of the designed filter.

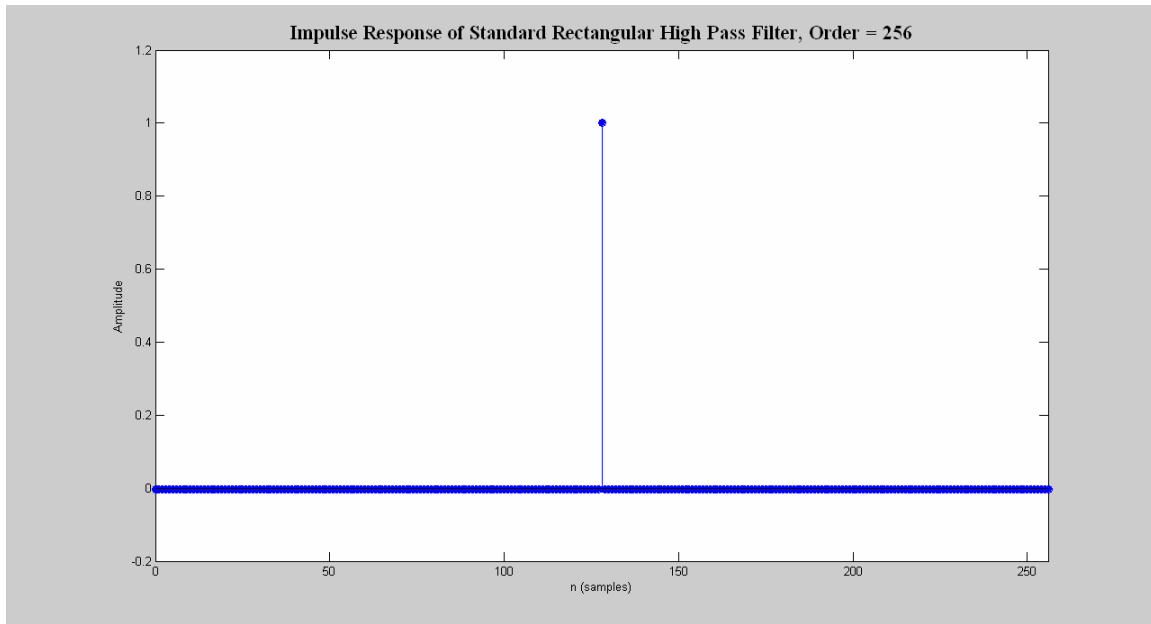


Figure 3-23 Impulse response of Rectangular High Pass Filter, Order = 256

Impulse response shows that the designed filter is symmetric about the point 129 and the number of points in the filter is odd i.e.,  $M = 257$ .

Figure 3-24 shows the magnitude and phase response of the designed filter.

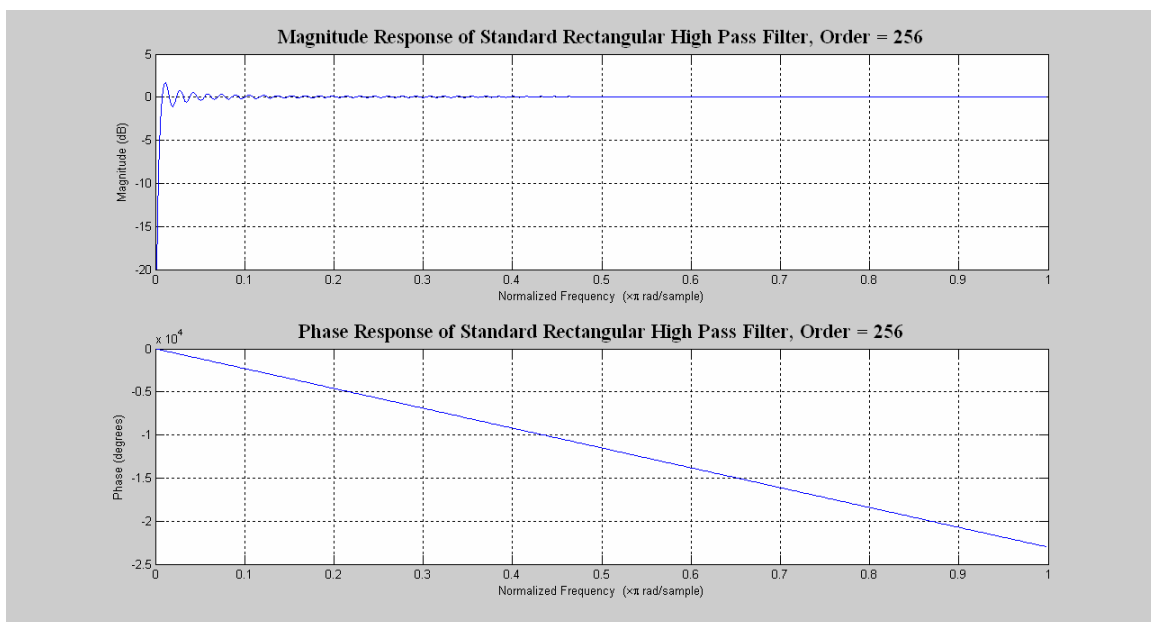


Figure 3-24 Magnitude and Phase response of Rectangular High Pass Filter

Magnitude response shows the response of the signal at different frequencies. From this response the cutoff frequency  $\omega_c$  is around  $0.01 \pi$  radians / sample. The attenuation of the signal in the stop band region is around 20 dB. The transition band width ranges from  $0.001 \pi$  radians / sample to  $0.01 \pi$  radians / sample. The cutoff frequency is in digital normalized angular frequency units. The method to obtain digital normalized angular frequency from Hz is explained below.

For Log1, Sampling time = 12ns

i.e., 512 samples or data points are collected in 12ns

Time between samples (Time to take one sample),  $T_s = \frac{12}{512} = 0.0234375 \text{ ns}$

For high-pass filter, desired cutoff frequency  $f = 300 \text{ MHz}$  (Agrawal 2005),

Angular Frequency =  $\Omega = 2\pi f = 2 \times 3.14 \times 300 \times 10^6 \text{ Hz}$

$$= 1,885 \times 10^6 \text{ Hz}$$

Digital (normalized angular) frequency =  $\omega = \frac{\Omega}{F_s} = \Omega T_s$  (Since  $F_s = \frac{1}{T_s}$ )

$$= 1,885 \times 10^6 \times 0.0234375 \times 10^{-9}$$

$$= 0.04417 \text{ radians/sample}$$

$$\text{or} = 0.01406 \pi \text{ radians / sample}$$

From this frequency calculation, it can be observed that the cutoff frequency obtained from the magnitude response Figure 3-24 matches with the desired cutoff frequency.

Figure 3-25 shows the surf plot of the radar scanned data of Log1 after applying low pass filter. The notable factor is that the image has been smoothened after the elimination of high frequencies.

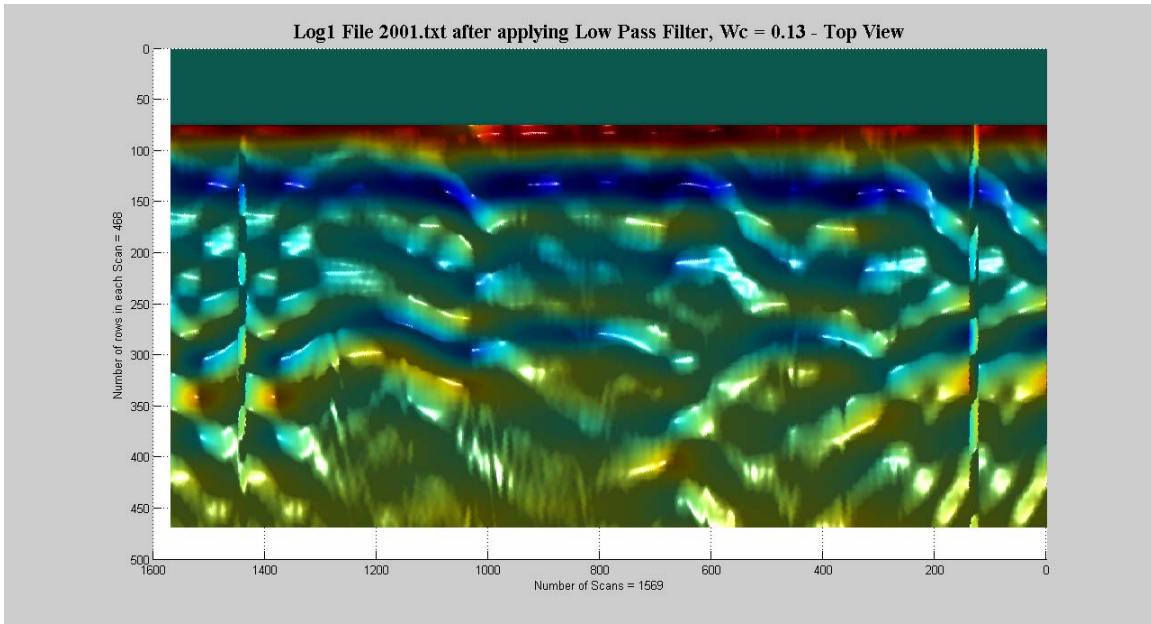


Figure 3-25 Radar scan after applying Low Pass Filter

Figure 3-26 and 3-27 shows the surf plot of the radar scanned data of Log1 after applying high-pass filter.

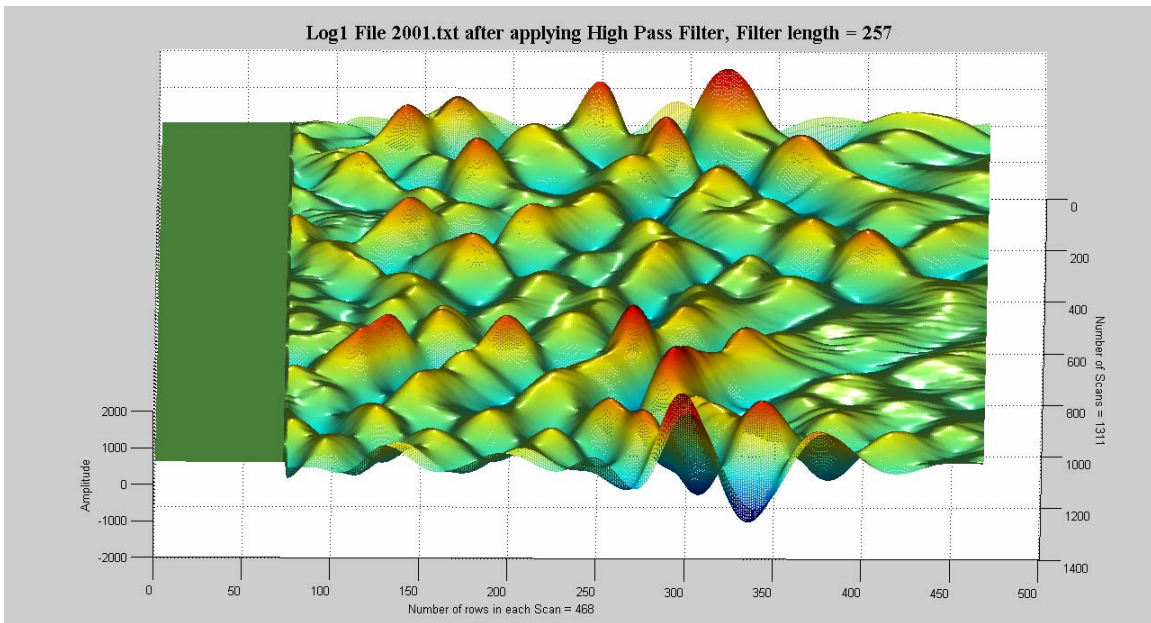


Figure 3-26 Radar scan after applying High Pass Filter

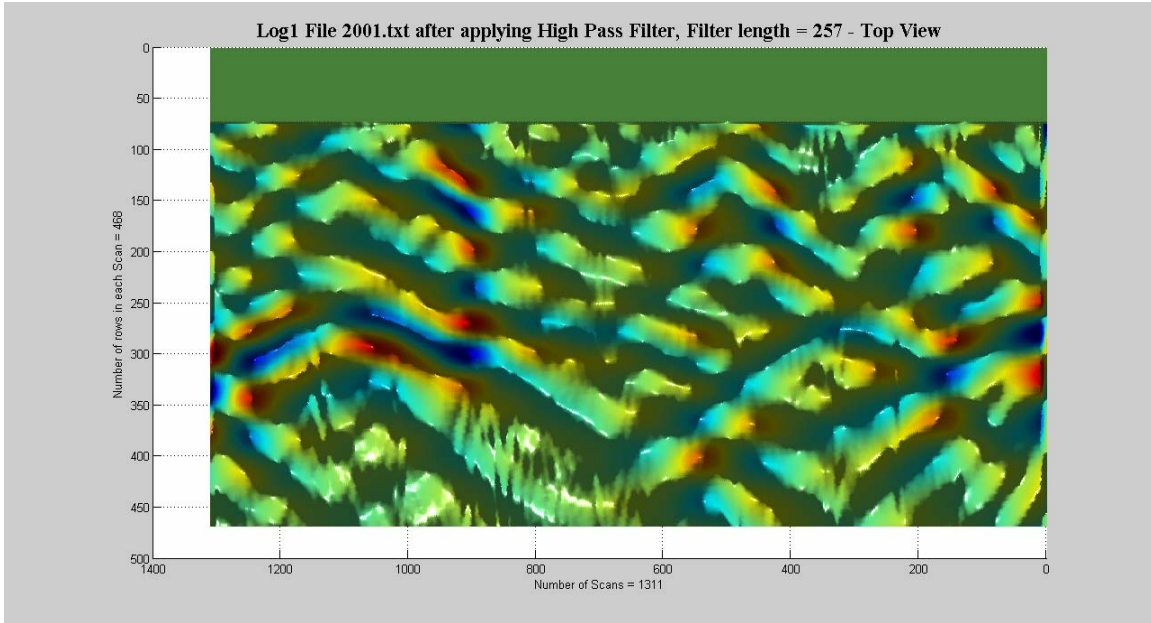


Figure 3-27 Top view of Figure 3-26 with 90° clockwise rotation

Data processed using RADAN 5.0 software (Figure 2-8) has been given along with the high pass filter applied radar scan (Figure 3-27) for comparison. It can be seen from the Figures 3-27 and 2-8 that the results of MATLAB algorithm are in agreement with the results of RADAN software.

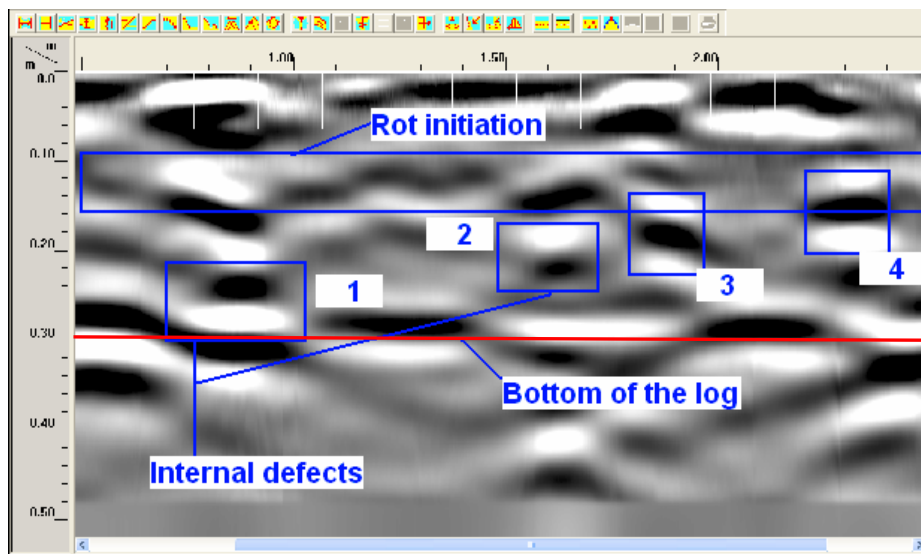


Figure 2-8 RADAN Processed GPR data of Log1 (Agrawal 2005)

Figure 3-28 shows the bottom reflection removed radar scan after applying high-pass filter.

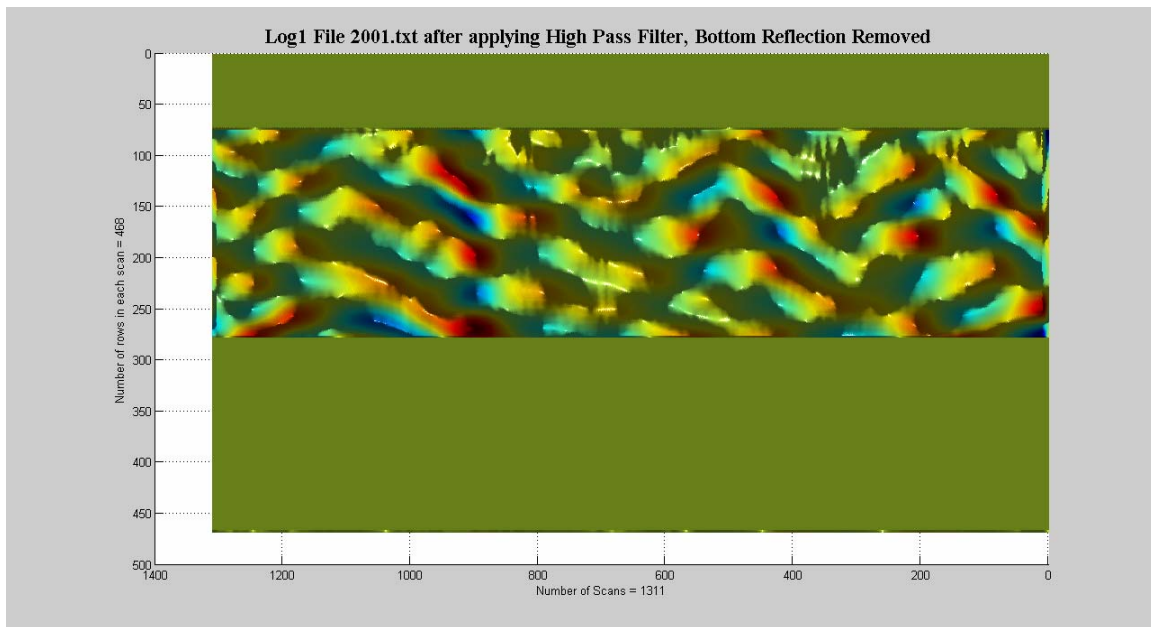


Figure 3-28 Radar scan after applying High Pass Filter and removing Bottom Reflection

### 3.13 Threshold

Threshold is one of the methods to select the defects region on the radar scan. Signal amplitudes above a particular threshold value have to be selected and the remaining data has to be discarded. Threshold has been done for value = 700 for Log1. This threshold value was arrived through trial and error method. Figure 3-29 shows the radar scan after applying threshold. This scan contains both defects and noise. Defects have been identified in order to compare with the RADAN output. But in real time application, even the noise has to be considered as defect. This noise has been eliminated in the subsequent scans by removing bottom reflection and edge effect.

Figure 3-30 also shows the radar scan after applying threshold of value = 700. But this scan is the one with zero correction done by aligning all the peaks in one line. This scan contains both defects and noise. This scan is little different from Figure 3-29. The difference can be noticed in the defect detected as rot initiation. Defects have been identified in this scan in order to compare with the RADAN output.

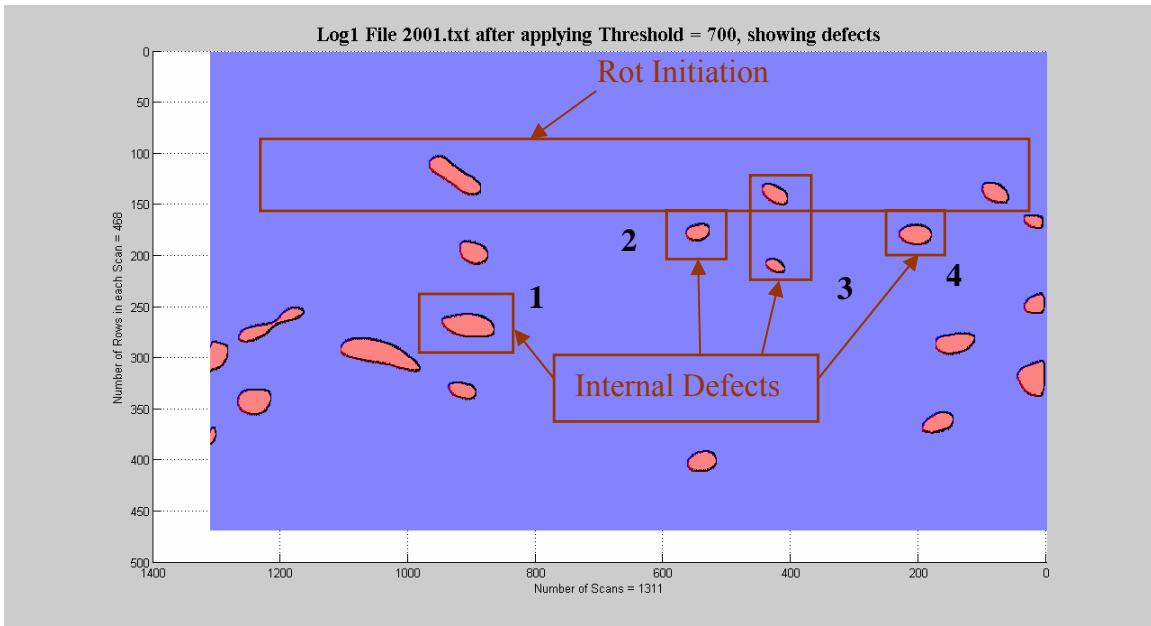


Figure 3-29 Radar scan showing defects for threshold value = 700

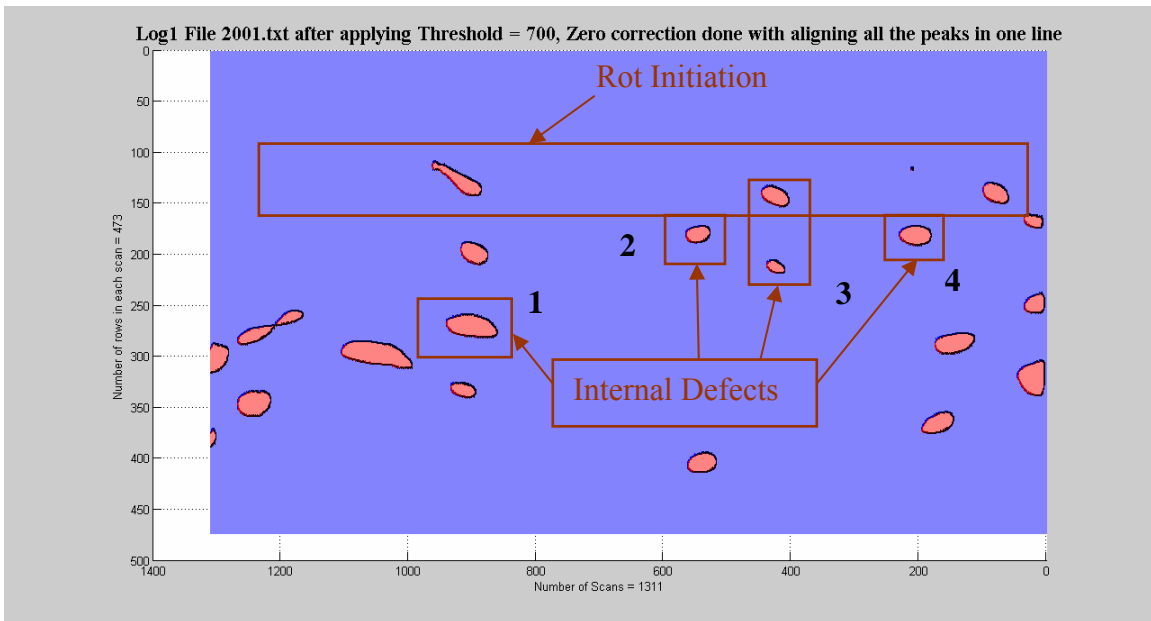


Figure 3-30 Radar scan showing defects for threshold value = 700, Zero correction done by aligning all the peaks in one line

Figure 3-31 also shows the radar scan after applying threshold of value = 700. But the bottom reflection has been removed in this scan. Reduction in the noise can be seen in this scan.

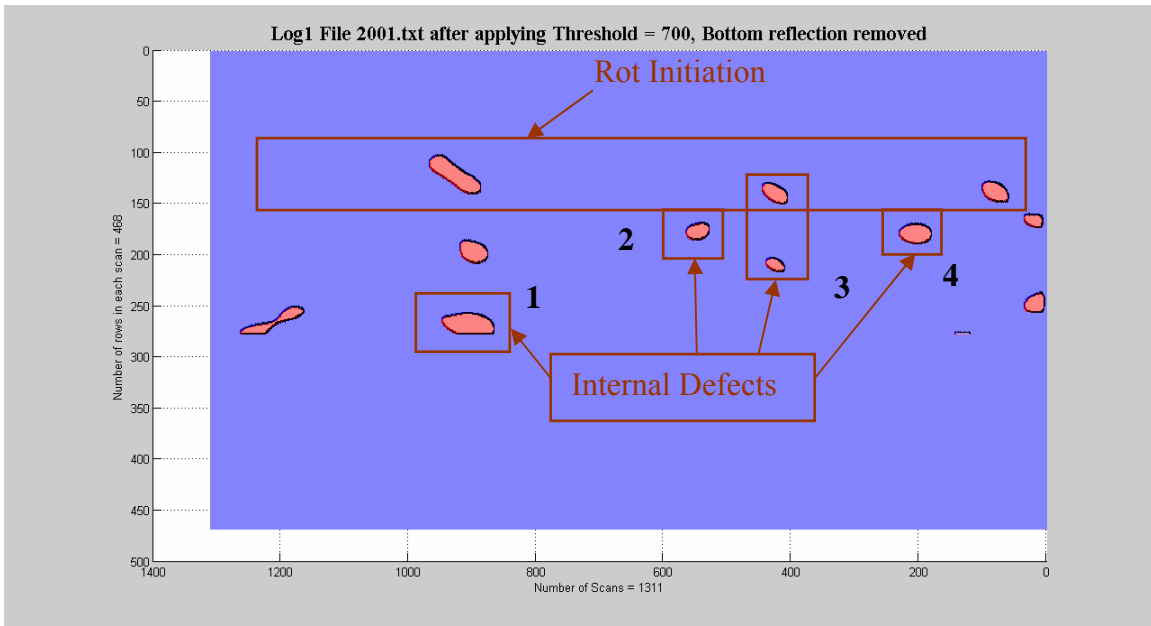


Figure 3-31 Radar scan showing defects for threshold value = 700, Bottom reflection removed

Figure 3-32 shows the radar scan after removing edge effect from previous scan Figure 3-31. Edge effect removal involved setting the values of the first and last 50 scans to zero. There is lot of noise reduction in this scan compared to Figure 3-29.

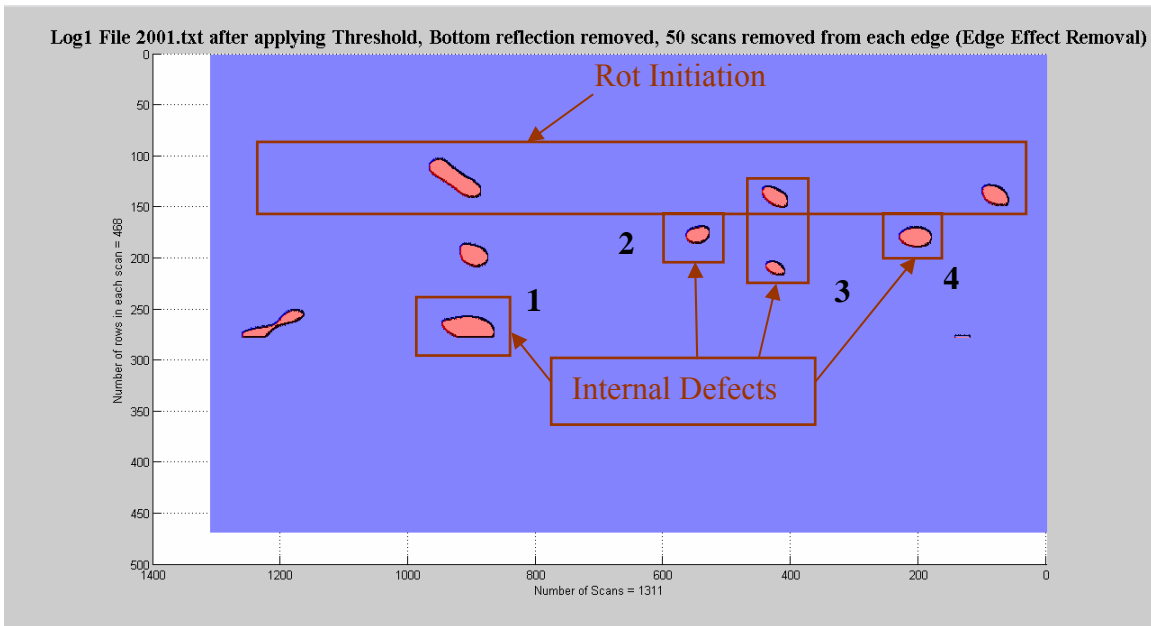


Figure 3-32 Radar scan showing defects for threshold value = 700, Bottom reflection removed, Edge effect removed

Figure 3-33 shows the defects detected in radar scan by applying automatic threshold. Defects were detected using automatic threshold since applying threshold manually is not possible in real time application. All the major defects have been identified by automatic threshold application. Figure 3-33 also involves edge effect and bottom reflection removal. Automatic threshold value calculated is 552 where as the manual threshold value is 700. Refer threshold value selection section for further details.

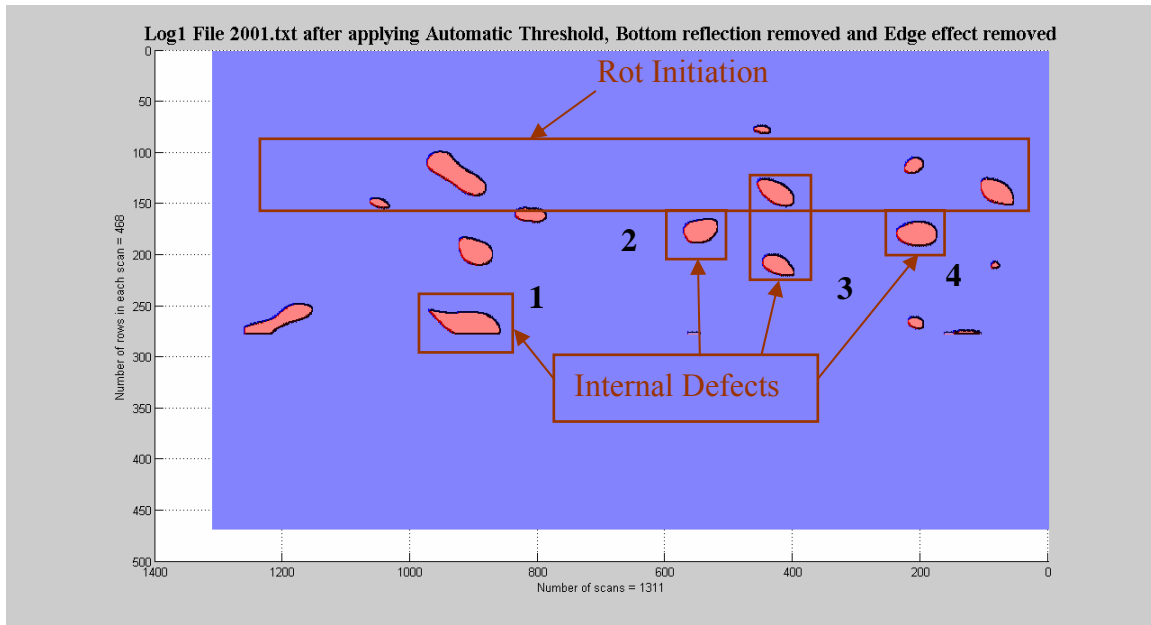


Figure 3-33 Radar scan showing defects for automatic threshold value = 552

### 3.14 Calculation of depth of the defect detected by MATLAB algorithm

The location of the defect in the Y-axis can be found by using the following formula which is now used for finding the thickness of the concrete slab. The depth of the defect in Y- axis is given by (Halabe et al. 1995),

$$\text{depth (d)} = \frac{V_o t_{\text{wood}}}{2\sqrt{\epsilon'_{\text{wood}}}} \quad (3.12)$$

Where,

d = depth of the defect in Y-axis in meters

$V_o$  = velocity of light in vacuum =  $3 \times 10^8$  m/sec

$t_{\text{wood}}$  = is the “round trip” travel time through wood in secs

$\epsilon'_{\text{wood}}$  = dielectric constant of the wood

= 10 ~ 26 for wood based on moisture content



For Log1,

Travel time or Range = 12 ns

Dielectric constant Value = 12

Substituting these in the above formula, we get the total depth traveled by signal

$$\begin{aligned}\text{Total depth (d)} &= \frac{3 \times 10^8 \times 12 \times 10^{-9}}{2\sqrt{12}} \\ &= 0.52 \text{ m}\end{aligned}$$

The 512 rows (data points) of each scan are collected in round trip travel time. So, 512 rows represent the total depth of 0.52 m. To find the depth of the defect, the ratio of the defect row with 512 has to be multiplied with the total depth. Figure 3-29 has been considered for calculating depth of the defects.

*Depth of Rot Initiation:*

Peak of the Rot initiation is spreading from 140<sup>th</sup> row to 150<sup>th</sup> row.

$$\begin{aligned}\text{Depth of Defect} &= \text{Total depth} \times \frac{\text{Defect Peak Point Number}}{512} \\ &= 0.52 \times \frac{140}{512} = 0.142 \text{ m} \\ &= 0.52 \times \frac{150}{512} = 0.152 \text{ m}\end{aligned}$$

The depth of the rot initiation detected by the MATLAB algorithm is varying from 0.142 m to 0.152 m from the surface where as the actual depth of rot initiation varies from 0.15 m to 0.16 m (Agrawal 2005).

*Depth of Defect1:*

Peak of the defect 1 lies near 260<sup>th</sup> row

$$\text{Depth of Defect} = 0.52 \times \frac{260}{512} = 0.264 \text{ m}$$

The depth of defect 1 detected by the MATLAB algorithm is 0.264 m from the surface where as the actual depth of defect 1 is around 0.25 m (Agrawal 2005).

*Depth of Defect2:*

Peak of the defect 2 lies near 175<sup>th</sup> row

$$\text{Depth of Defect} = 0.52 \times \frac{175}{512} = 0.178 \text{ m}$$

The depth of defect 2 detected by the MATLAB algorithm is 0.178 m from the surface where as the actual depth of defect 2 is around 0.18 m (Agrawal 2005).

*Depth of Defect3:*

Peak of the defect 3 lies near 215<sup>th</sup> row

$$\text{Depth of Defect} = 0.52 \times \frac{215}{512} = 0.218 \text{ m}$$

The depth of defect 3 detected by the MATLAB algorithm is 0.218 m from the surface where as the actual depth of defect 3 is around 0.23 m (Agrawal 2005)

*Depth of Defect4:*

Peak of the defect 4 lies near 187<sup>th</sup> point

$$\text{Depth of Defect} = 0.52 \times \frac{187}{512} = 0.19 \text{ m}$$

The depth of defect 4 detected by the MATLAB algorithm is 0.19 m from the surface where as the actual depth of defect 4 is around 0.23 m (Agrawal 2005). Table 3-2 shows the comparison of depths of the defects detected by MATLAB, RADAN with actual depths of the defects.

Table 3-2 Results

Sl. No.	Defect	Actual Depth	RADAN Detected	MATLAB Detected	Difference
1	Rot initiation	0.15 ~ 0.16 m	0.16 m	0.142 ~ 0.152 m	-0.008 m
2	Defect 1	0.25 m	0.26 m	0.264 m	0.014 m
3	Defect 2	0.18 m	0.18 m	0.178 m	-0.002 m
4	Defect 3	0.23 m	0.22 m	0.218 m	-0.012 m
5	Defect 4	0.23 m	0.2 m	0.19 m	-0.04 m

Distances of the defects from the edge of the scan are not calculated for this log since the actual distances of the defects from the edge of the scan are not known (Agrawal 2005).

MATLAB algorithm takes the following inputs from user to convert the defect location into X-Y coordinates in meters.

1. Average Diameter of the log
2. Scans per meter

Algorithm detects the 1<sup>st</sup> peak and the last valley position in each signal and takes the difference between them. Then the average of all these differences is calculated to find the number of rows in the scan that represents the diameter of the log. Then the ratio of the defect row to this number multiplied by the average diameter of the log gives the depth of the defect.

The ratio of defect column to scans per meter gives the distance of the defect from the edge of the scan. MATLAB algorithm puts the distance and depth of the defects location in a matrix in the form of X-Y coordinates in meters.

MATLAB algorithm can also calculate the depth of the defect by taking the values of dielectric constant and range (round trip travel time of the signal) from the user. For this, user has to know dielectric constant value. It calculates the depth as explained in Log1 depth calculation section.

**Conclusion:** The following things can be concluded from the above results.

1. The resolution of depth prediction of MATLAB algorithm is +0.014 m, -0.04 m.
2. The results obtained from MATLAB algorithm are in agreement with the RADAN software.
3. Defects location has been obtained in 2-Dimensional coordinates in meters.

## Chapter 4

### DEFECT DETECTION IN LOGS

#### 4.1 Log2 Data Processing using MATLAB

Log2 has been processed the same way as Log1. Figure 4-1 shows the surf plot of the raw data of Log2 along the Mark 4 of the log (Agrawal 2005).

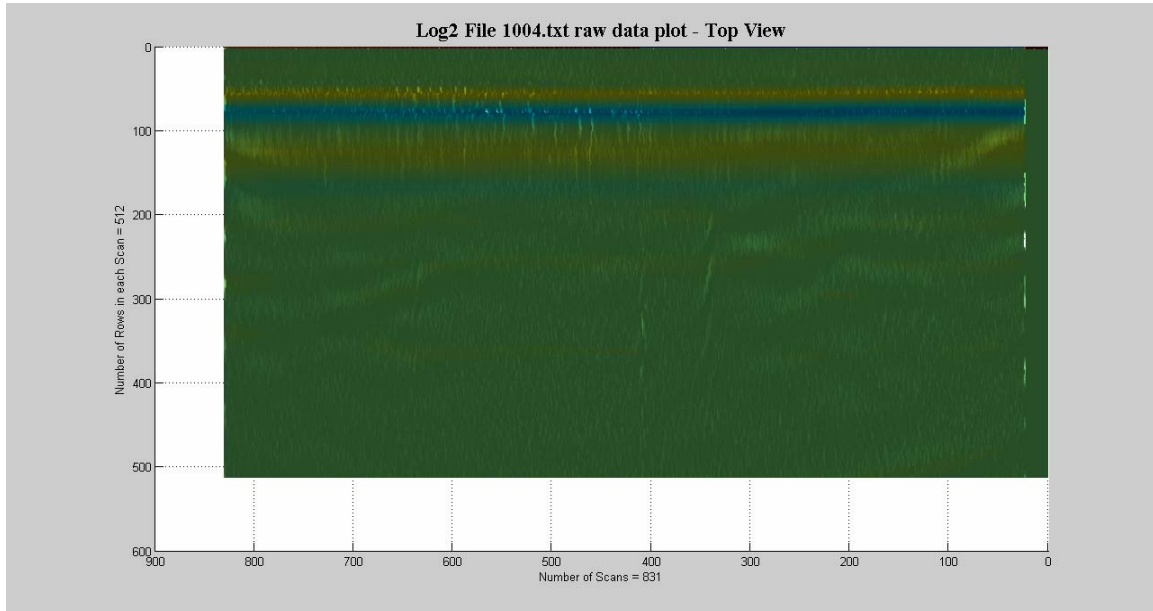


Figure 4-1 MATLAB 'Surf' plot of raw data of Log2

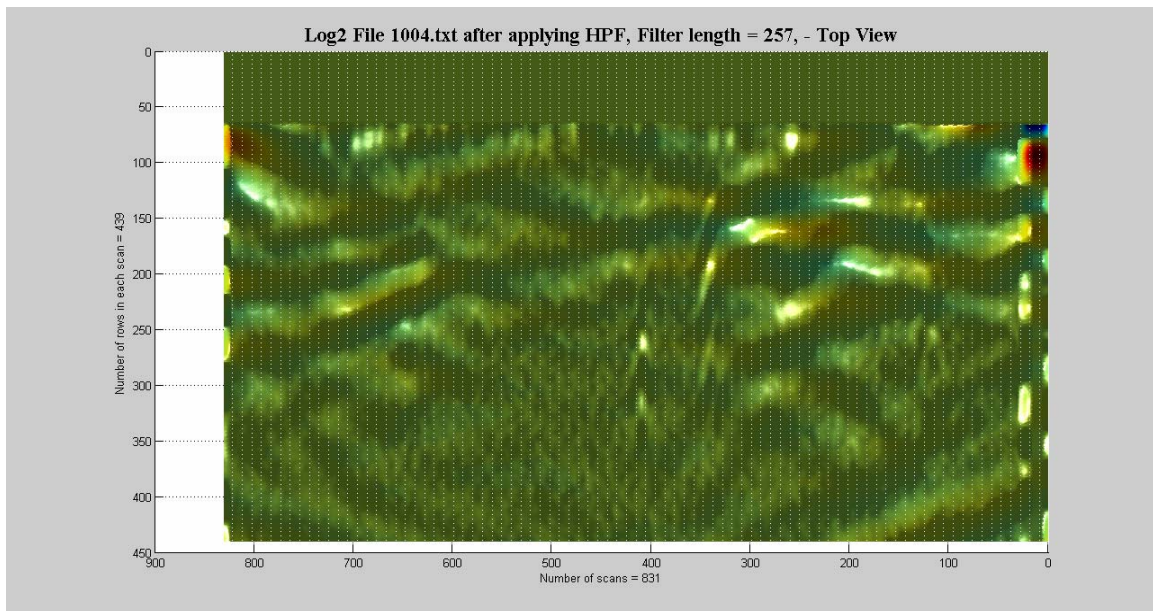


Figure 4-2 Radar scan after applying High Pass Filter

Figure 4-2 shows the radar scan after applying High Pass Filter. The defects patterns are not clear in the scan. To make the defect patterns appear better, the tall areas in the scan has to be removed. Figure 4-3 shows the radar scan after removing these tall areas. This involved removal of first 50 columns (scans) and clipping till 2<sup>nd</sup> maximum plus 35 rows of the signal. The removal of tall areas has been done just for the purpose of presentation.

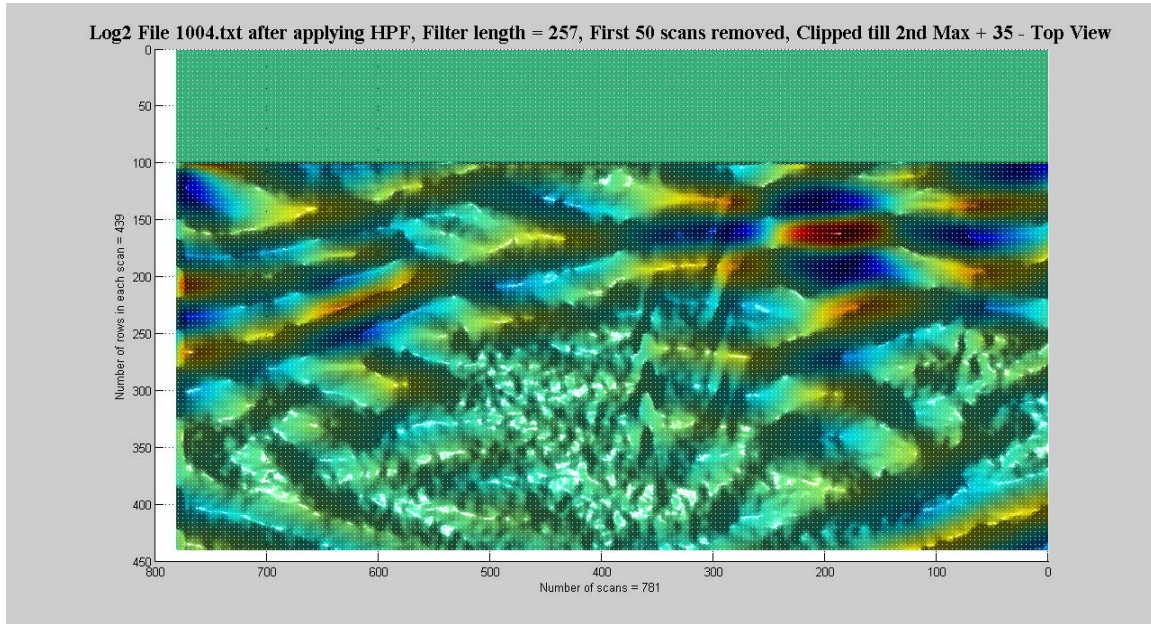


Figure 4-3 Radar scan after applying High Pass Filter, First 50 scans removed, Clipped till 2<sup>nd</sup> Maximum + 35 Rows of the signal

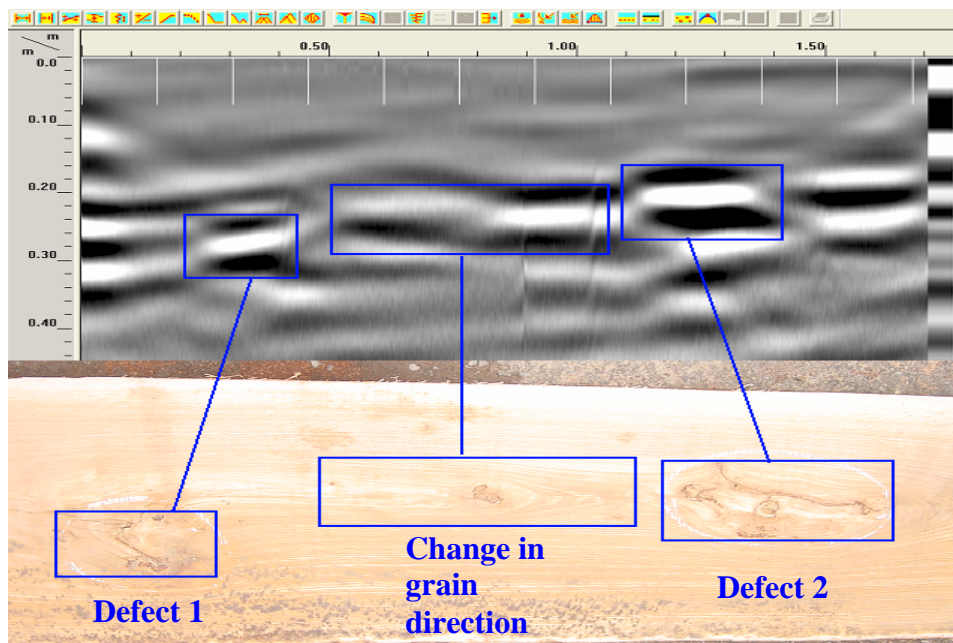


Figure 4-4 RADAN Processed GPR data of Log2 (Agrawal 2005)

Data processed using RADAN 5.0 software (Figure 4-4) has been given along with the high pass filter applied radar scan (Figure 4-3) for comparison. It can be seen from the Figures 4-3 and 4-4 that the results of MATLAB algorithm are in agreement with the results of RADAN software. Figure 4-5 shows the radar scan after applying threshold. Threshold value used is 140. Figure 4-5 contains both defects and noise. Defects have been identified in order to compare with the RADAN output.

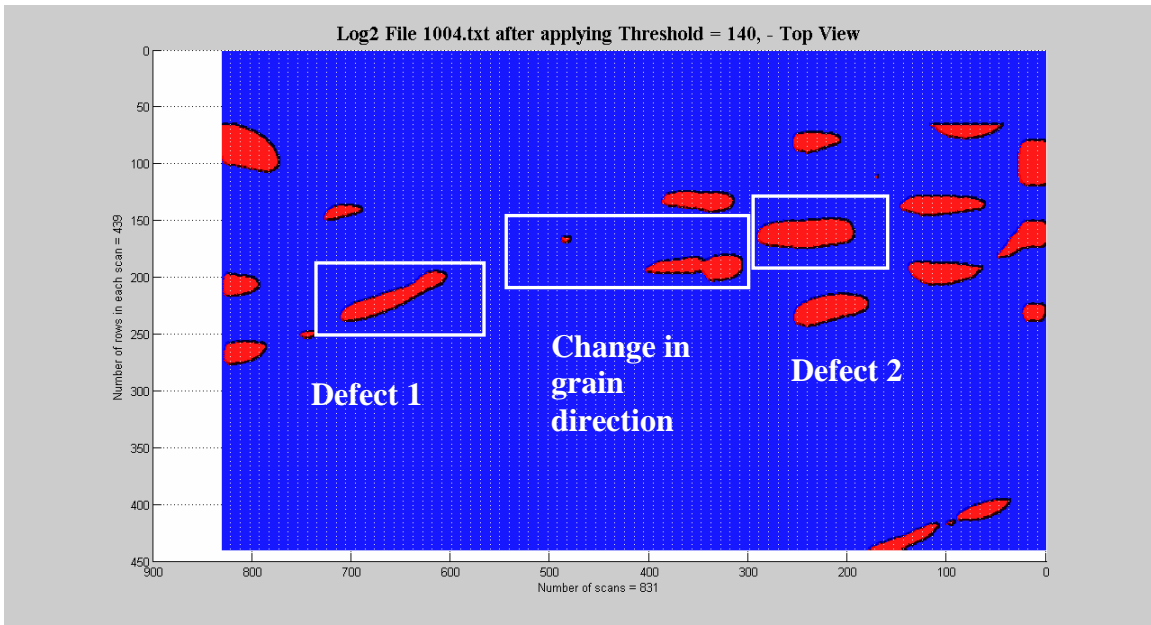


Figure 4-5 Radar scan showing defects for threshold value = 140

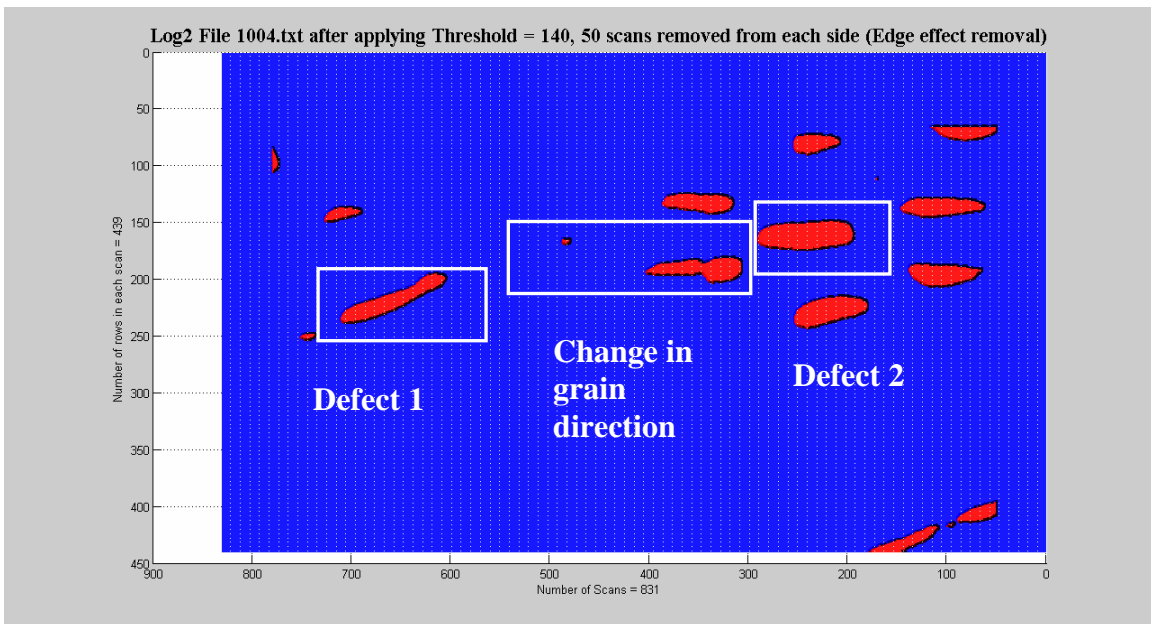


Figure 4-6 Radar scan showing defects for threshold value = 140, Edge effect removed

Figure 4-6 shows radar scan after removing edge effect. Some reduction in noise can be observed in this scan. Figure 4-7 shows the defects detected in radar scan by applying automatic threshold. Figure 4-7 also involves edge effect removal and check for bottom reflection removal. All the major defects have been identified by automatic threshold application. Automatic threshold value calculated is 186 where as the manual threshold value is 140.

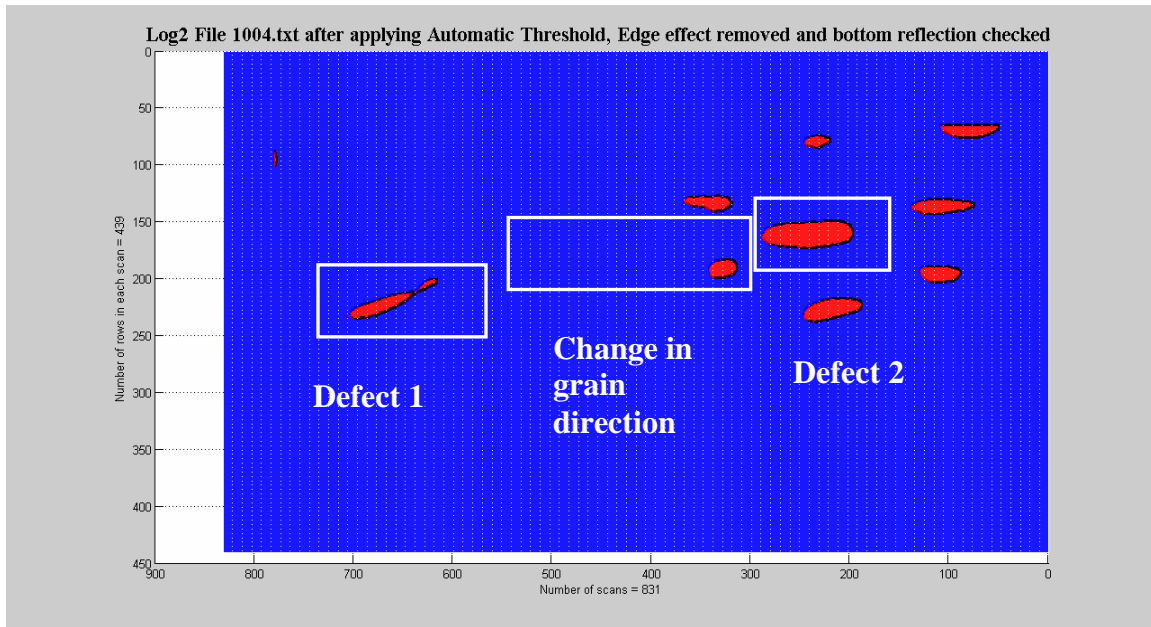


Figure 4-7 Radar scan showing defects for automatic threshold value = 186

For Log2,

Travel time or Range = 15 ns

Dielectric constant Value = 15.5

Substituting these in equation 3.12, we get the total depth traveled by signal

$$\begin{aligned} \text{Total depth (d)} &= \frac{3 \times 10^8 \times 15 \times 10^{-9}}{2\sqrt{15.5}} \\ &= 0.572 \text{ m} \end{aligned}$$

The 512 rows (data points) of each scan are collected in round trip travel time. So, 512 rows represent the total depth of 0.572 m. To find the depth of the defect, the ratio of the defect row with 512 has to be multiplied with the total depth. Figure 4-5 has been considered for calculating depth of the defects.

*Depth of Defect 1:*

Peak of the defect 1 lies near 220<sup>th</sup> row

$$\text{Depth of Defect} = 0.572 \times \frac{220}{512} = 0.246 \text{ m}$$

The depth of defect 1 detected by the MATLAB algorithm is 0.246 m from the surface where as the actual depth of defect 1 is around 0.26 m (Agrawal 2005).

*Distance of Defect 1:*

Center of Defect 1 is around 170 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{170}{472.441} \times 1 = 0.359 \text{ m}$$

Distance of defect 1 detected by the MATLAB algorithm is 0.359 m from the point of start of scan where as the actual distance is around 0.4 m (Agrawal 2005).

*Depth of Defect 2:*

Peak of the defect 2 lies near 160<sup>th</sup> row

$$\text{Depth of Defect} = 0.572 \times \frac{160}{512} = 0.18 \text{ m}$$

The depth of defect 2 detected by the MATLAB algorithm is 0.18 m from the surface where as the actual depth of defect 2 is around 0.22 m (Agrawal 2005).

*Distance of Defect 2:*

Center of Defect 2 is around 570 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{570}{472.441} \times 1 = 1.2 \text{ m}$$

Distance of defect 2 detected by the MATLAB algorithm is 1.2 m from the point of start of scan where as the actual distance is also 1.2 m (Agrawal 2005).



Table 4-1 shows the comparison of depths and lengths of the defects detected by MATLAB, RADAN with actual depths and lengths of the defects.

Table 4-1 Results

Sl. No.	Defect	Actual	RADAN Detected	MATLAB Detected	Difference
1	Defect 1 Depth	0.26 m	0.27 m	0.246 m	-0.014 m
2	Defect 1 distance	0.4 m	0.35 m	0.359 m	-0.041 m
3	Defect 2 Depth	0.22 m	0.20 m	0.18 m	-0.04 m
4	Defect 2 distance	1.2 m	1.2 m	1.2 m	0.00 m

Figure 4-8 shows the surf plot of the raw data of Log2 along the Mark 5 of the log (Agrawal 2005).

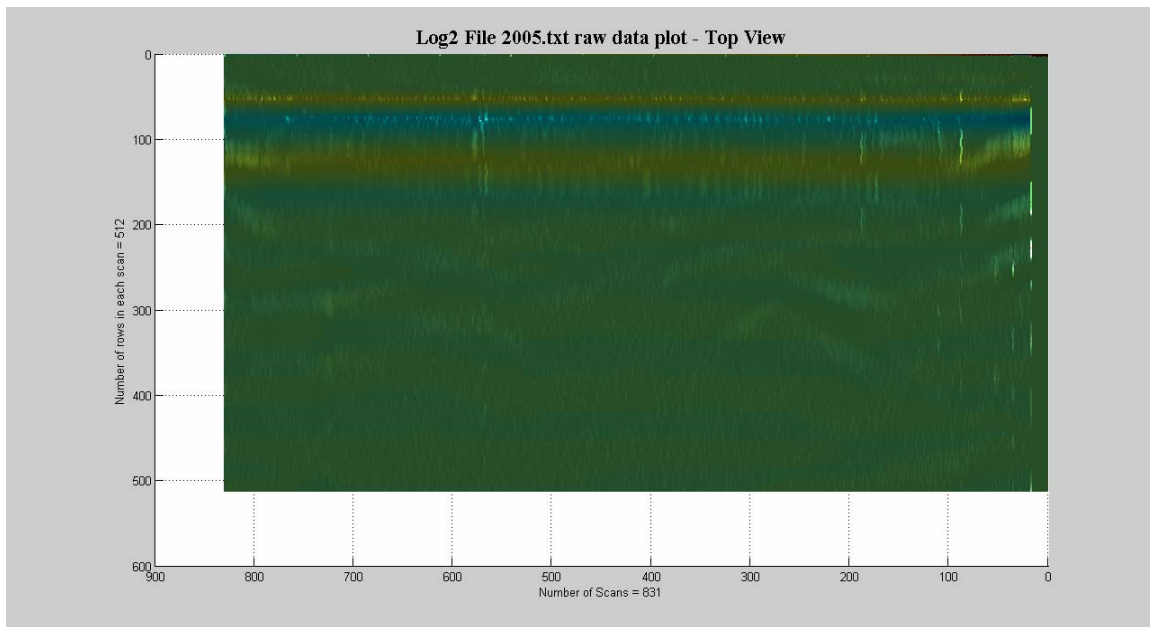


Figure 4-8 MATLAB 'Surf' plot of raw data of Log2

Figure 4-9 shows the radar scan after applying High Pass Filter. The defects patterns are not clear in the scan.

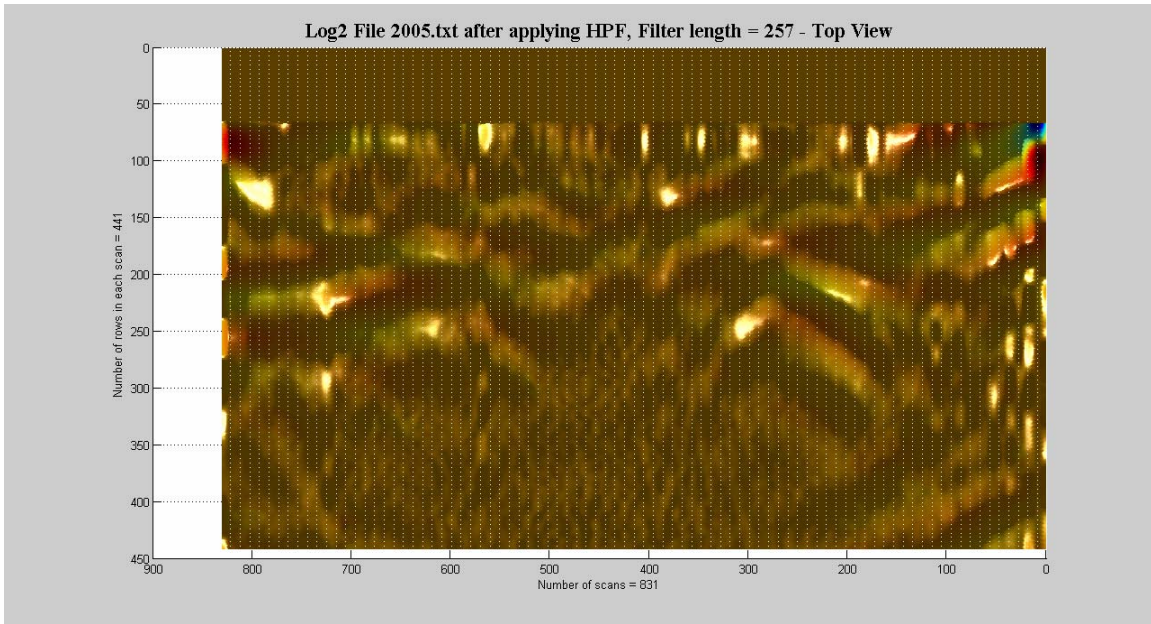


Figure 4-9 Radar scan after applying High Pass Filter

To make the defect patterns appear better, the tall areas in the scan has to be removed. Figure 4-10 shows the radar scan after removing these tall areas. This involved removal of first 25 columns (scans) and clipping till 2<sup>nd</sup> maximum plus 35 rows of the signal. The removal of tall areas has been done just for the purpose of presentation.

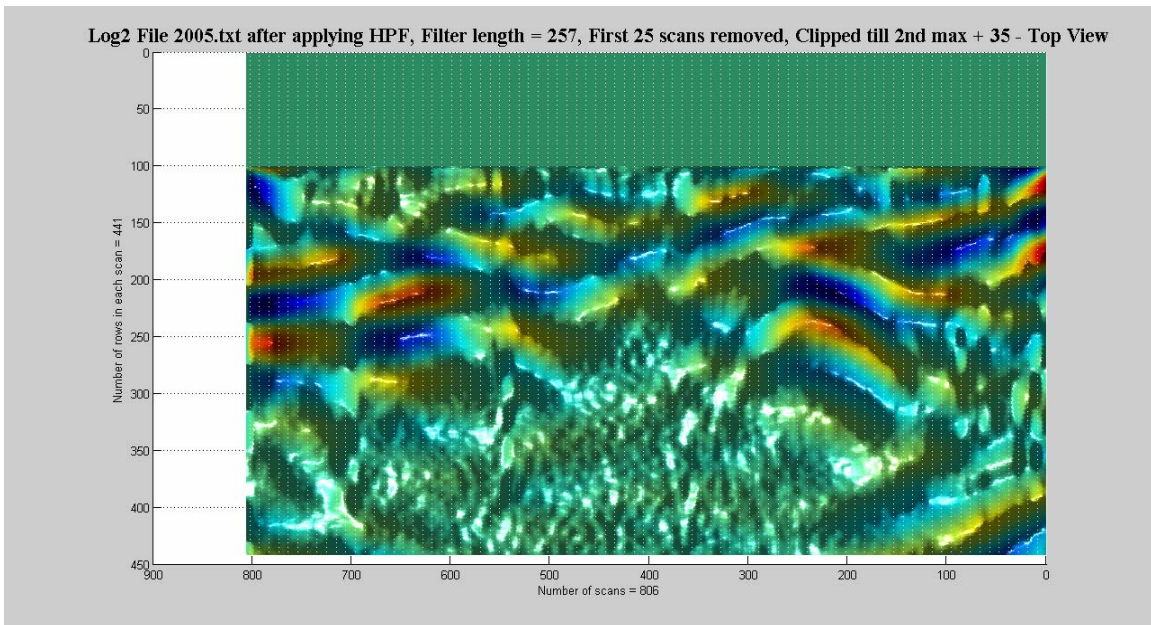


Figure 4-10 Radar scan after applying High Pass Filter, First 25 scans removed, Clipped till 2<sup>nd</sup> Maximum + 35 Rows of the signal

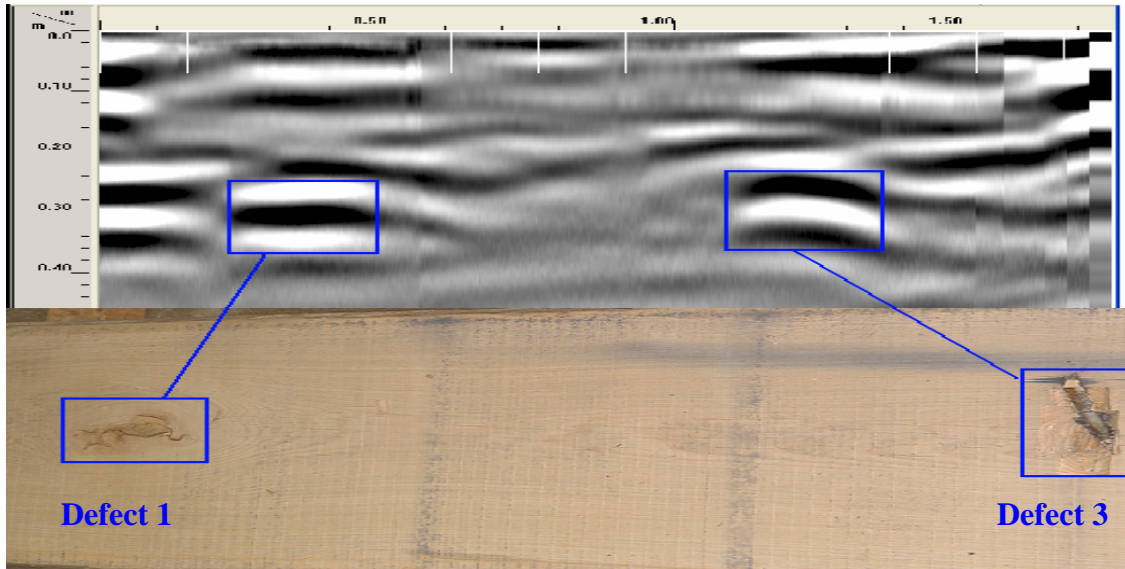


Figure 4-11 RADAN Processed GPR data of Log2 (Agrawal 2005)

Data processed using RADAN 5.0 software (Figure 4-11) has been given along with the high pass filter applied radar scan (Figure 4-10) for comparison. It can be seen from the Figures 4-10 and 4-11 that the results of MATLAB algorithm are in agreement with the results of RADAN software.

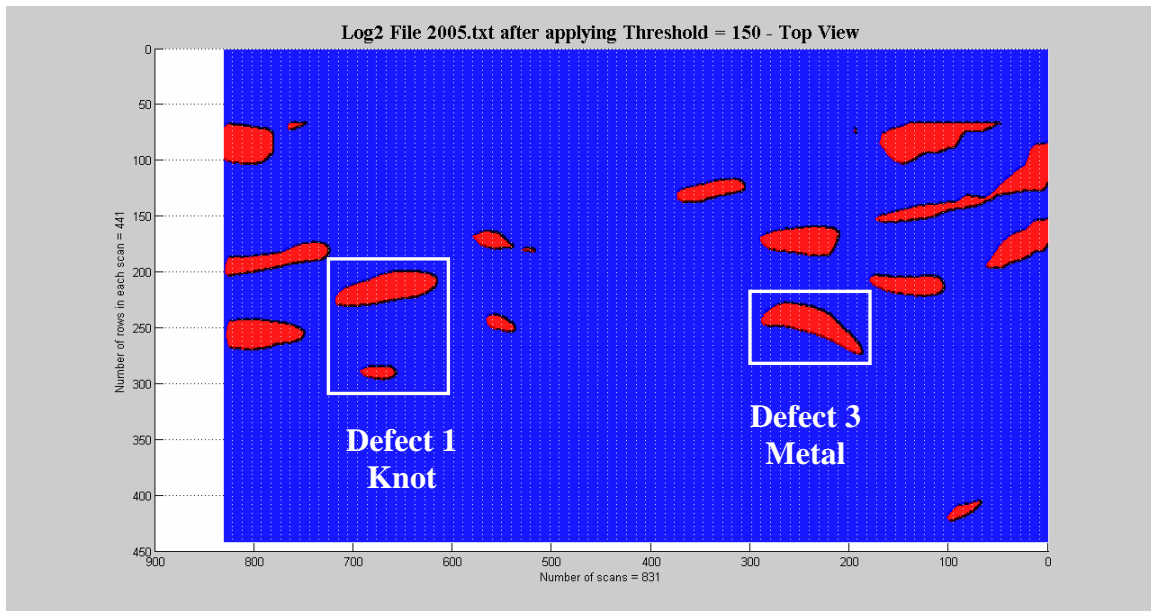


Figure 4-12 Radar scan showing defects for threshold value = 150

Figure 4-12 shows the radar scan after applying threshold. Threshold value used is 150. Figure 4-12 contains both defects and noise. Defects have been identified in order to compare with the RADAN output.

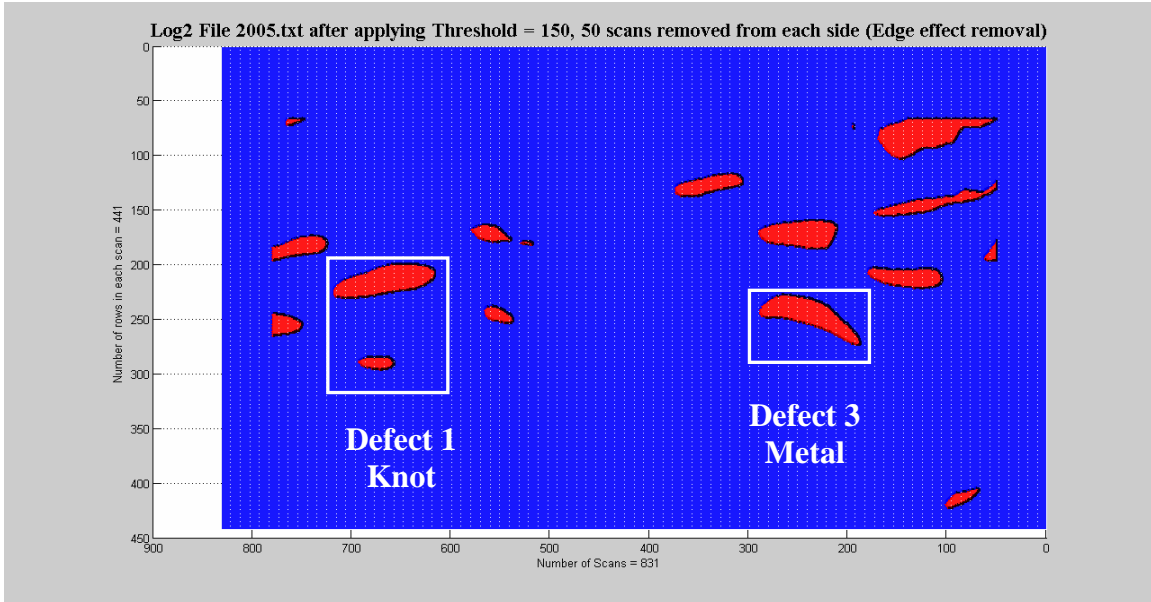


Figure 4-13 Radar scan showing defects for threshold value = 150, Edge effect removed

Figure 4-13 shows radar scan after removing edge effect. Some reduction in noise can be observed in this scan. Figure 4-14 also shows the radar scan after applying threshold of value = 150. But this scan is the one with zero correction done by aligning all the peaks in one line. This scan does not show defect 3 (metal) correctly. Aligning all the peaks in one line while doing zero correction might have caused this. So, aligning all the peaks in one line to do zero correction is not used for other logs.

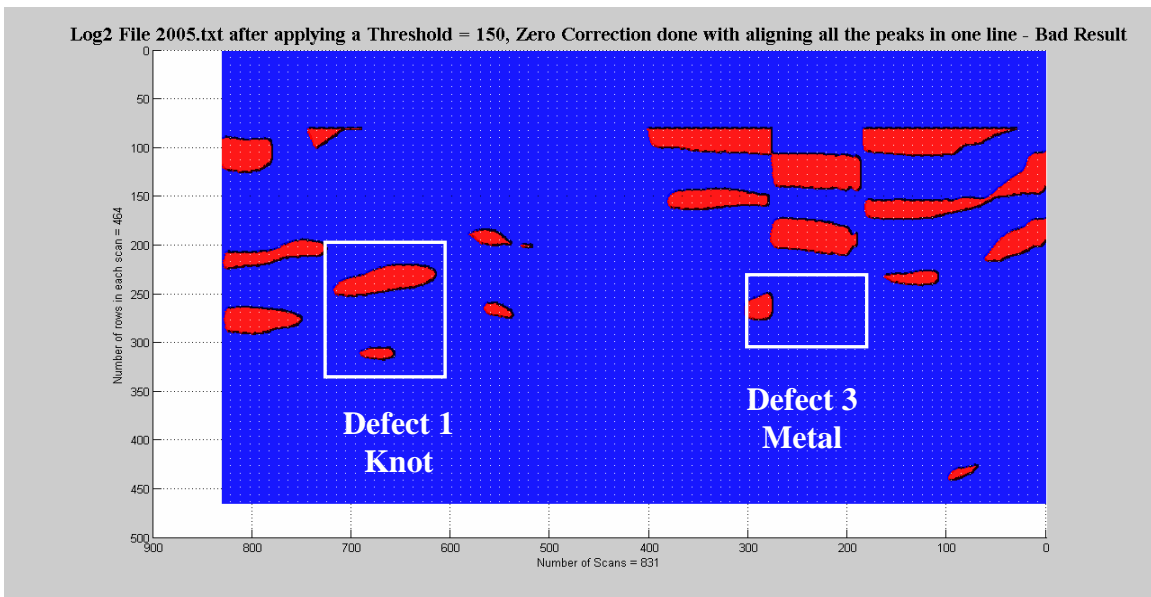


Figure 4-14 Radar scan showing defects for threshold value = 150, Zero correction done by aligning all the peaks in one line

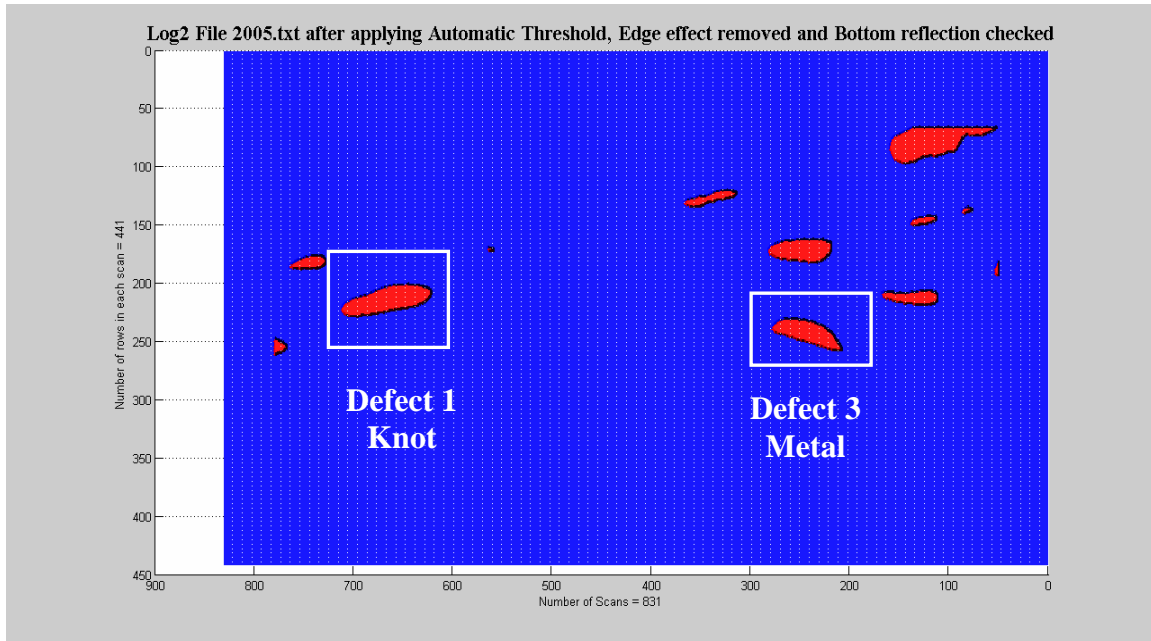


Figure 4-15 Radar scan showing defects for automatic threshold value = 197

Figure 4-15 shows the defects detected in radar scan by applying automatic threshold. Figure 4-15 also involves edge effect removal and check for bottom reflection removal. All the major defects have been identified by automatic threshold application. Automatic threshold value calculated is 197 where as the manual threshold value is 150.

*Depth of Defect3:*

Peak of the defect 3 lies near 235<sup>th</sup> row

$$\text{Depth of Defect} = 0.572 \times \frac{235}{512} = 0.263 \text{ m}$$

The depth of defect 3 detected by the MATLAB algorithm is 0.263 m from the surface where as the actual depth of defect 3 is around 0.25 m (Agrawal 2005).

*Distance of Defect 3:*

Center of Defect 3 is around 570 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{570}{472.441} \times 1 = 1.2 \text{ m}$$

Distance of defect 3 detected by the MATLAB algorithm is 1.2 m from the point of start of scan where as the actual distance is around 1.2 m (Agrawal 2005).

Table 4-2 shows the comparison of depths and lengths of the defects detected by MATLAB, RADAN with actual depths and lengths of the defects.

Table 4-2 Results

Sl. No.	Defect	Actual	RADAN Detected	MATLAB Detected	Difference
1	Defect 3 Depth	0.25 m	0.29 m	0.263 m	0.013 m
2	Defect 3 distance	1.2 m	1.2 m	1.2 m	0.00 m

#### 4.2 Log3 Data Processing using MATLAB

Log3 has been processed the same way as Log1. Figure 4-16 shows the surf plot of the raw data of Log3.

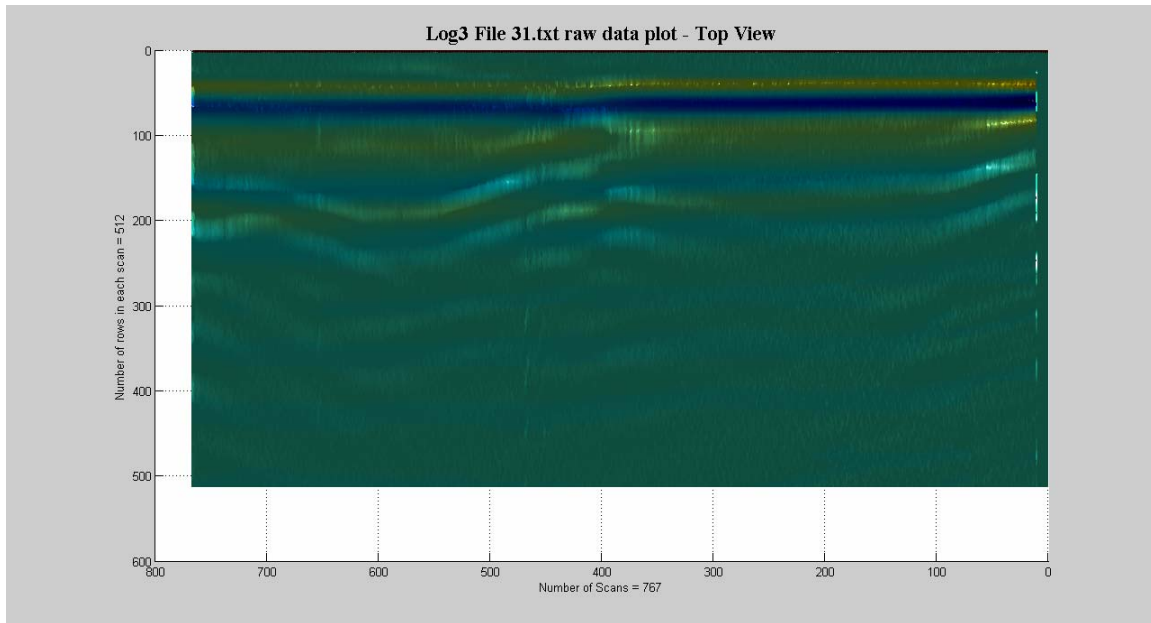


Figure 4-16 MATLAB ‘Surf’ plot of raw data of Log3

Figure 4-17 shows the radar scan after applying High Pass Filter. The defects patterns are seen clear in this scan.

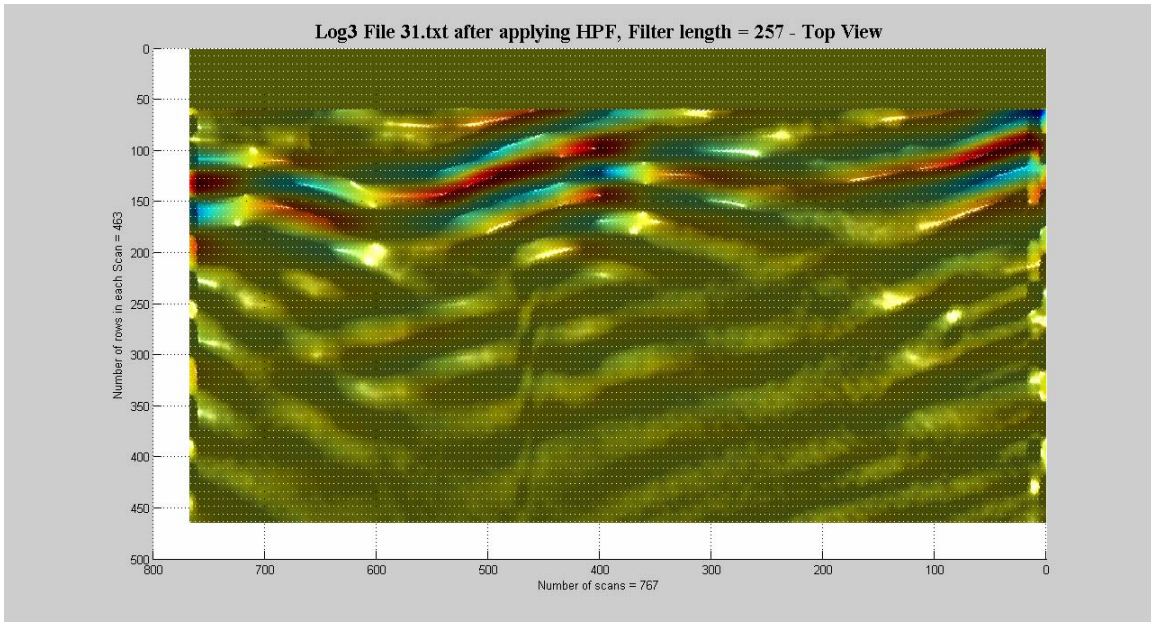


Figure 4-17 Radar scan after applying High Pass Filter

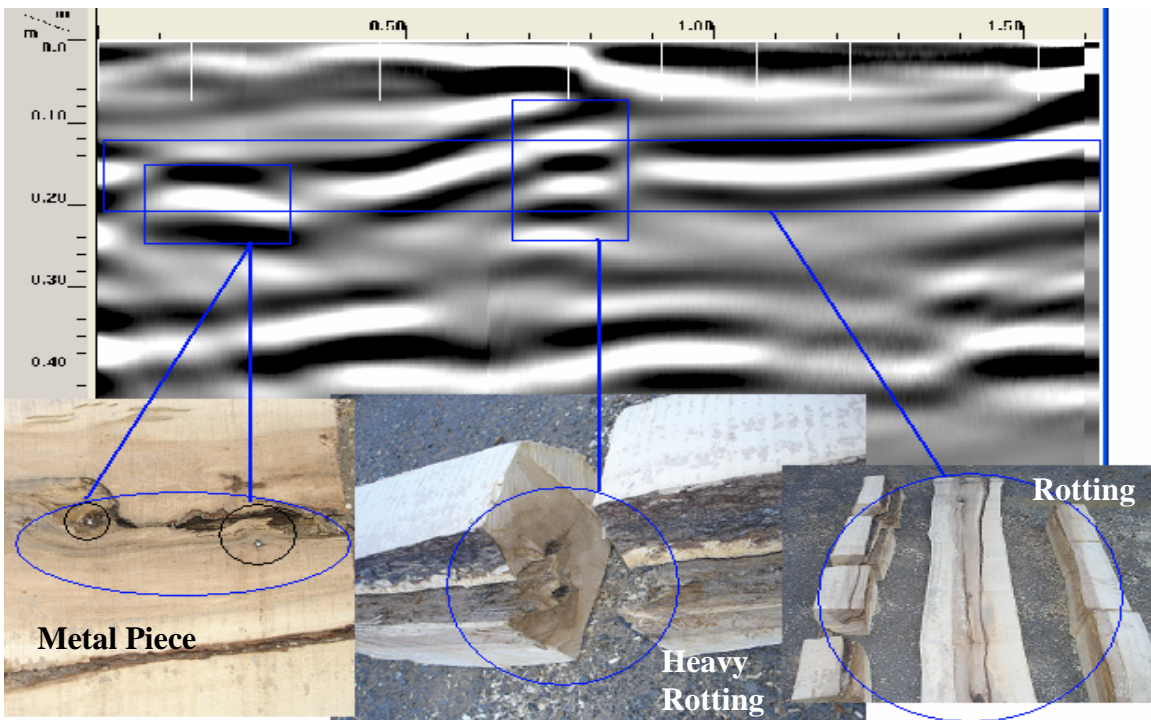


Figure 4-18 RADAN Processed GPR data of Log3 (Agrawal 2005)

Data processed using RADAN 5.0 software (Figure 4-18) has been given along with the high pass filter applied radar scan (Figure 4-17) for comparison. It can be seen from the Figures 4-17 and 4-18 that the results of MATLAB algorithm are in agreement with the results of RADAN software.

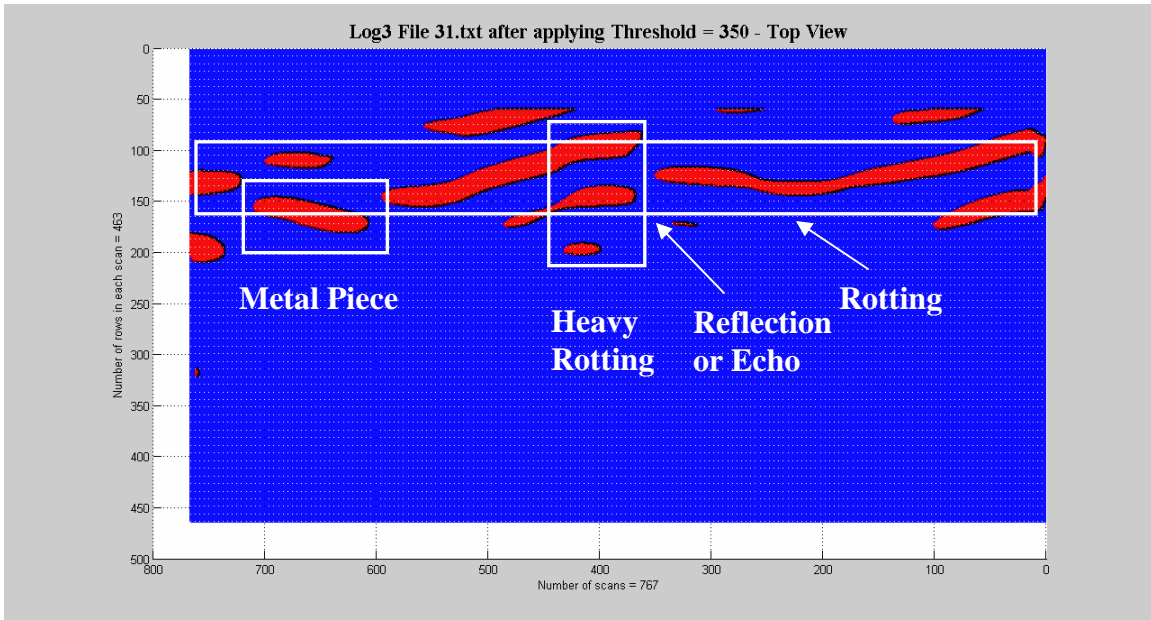


Figure 4-19 Radar scan showing defects for threshold value = 350

Figure 4-19 shows the radar scan after applying threshold. Threshold value used is 350. Figure 4-19 contains both defects and noise. Defects have been identified in order to compare with the RADAN output.

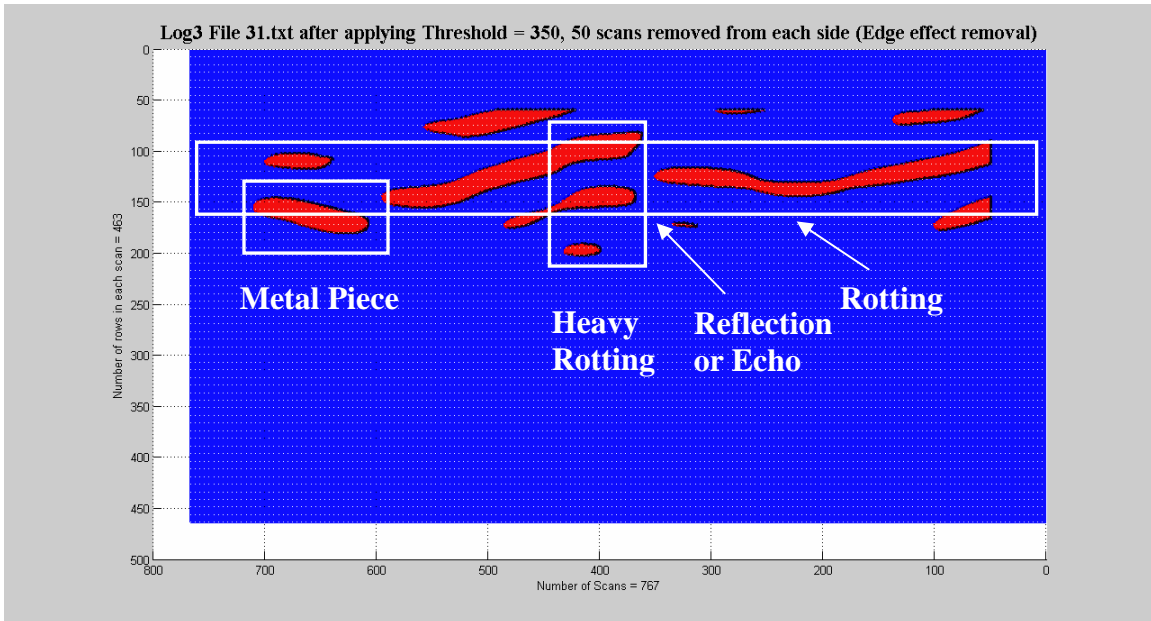


Figure 4-20 Radar scan showing defects for threshold value = 350, Edge effect removed

Figure 4-20 shows radar scan after removing edge effect. Some reduction in noise can be observed in this scan. Figure 4-21 shows the defects detected in radar scan by



applying automatic threshold. Figure 4-21 also involves edge effect removal and check for bottom reflection removal. Some of the defects were not identified by automatic threshold application in this log since automatic threshold value calculated is 893 where as the manual threshold value is 350.

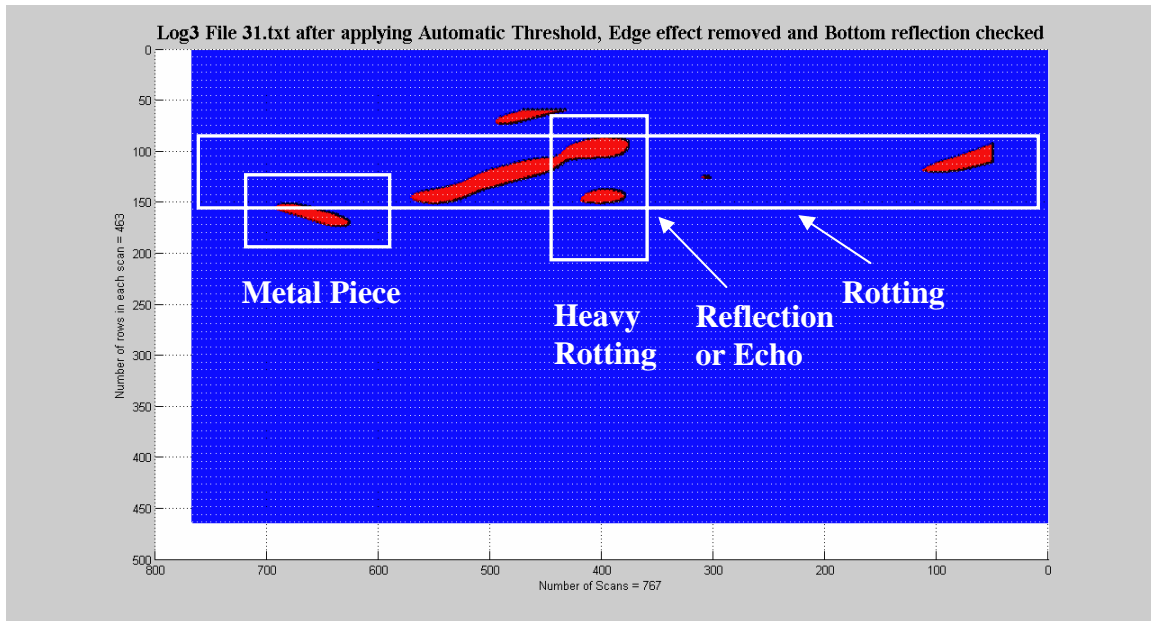


Figure 4-21 Radar scan showing defects for automatic threshold value = 893

For Log3,

Travel time or Range = 15 ns

Dielectric constant Value = 14

Substituting these in equation 3.12, we get the total depth traveled by signal

$$\begin{aligned} \text{Total depth (d)} &= \frac{3 \times 10^8 \times 15 \times 10^{-9}}{2\sqrt{14}} \\ &= 0.601 \text{ m} \end{aligned}$$

The 512 rows (data points) of each scan are collected in round trip travel time. So, 512 rows represent the total depth of 0.601 m. To find the depth of the defect, the ratio of the defect row with 512 has to be multiplied with the total depth. Figure 4-19 has been considered for calculating depth of the defects.

*Depth of Rot:*

Rot is spreading between 100<sup>th</sup> and 160<sup>th</sup> row in the scan. For 100<sup>th</sup> row,

$$\text{Depth of Defect} = 0.601 \times \frac{100}{512} = 0.12 \text{ m}$$

For 160<sup>th</sup> row,

$$\text{Depth of Defect} = 0.601 \times \frac{160}{512} = 0.19 \text{ m}$$

The depth of the rot detected by the MATLAB algorithm is spreading between 0.12 m to 0.19 m from the surface where as the RADAN detected depth of the rot is spreading between 0.13 m to 0.2 m (Agrawal 2005).

*Depth of Metal piece:*

Peak of the metal piece lies near 160<sup>th</sup> row

$$\text{Depth of Defect} = 0.601 \times \frac{160}{512} = 0.188 \text{ m}$$

The depth of metal piece detected by the MATLAB algorithm is 0.188 m from the surface where as the actual depth is around 0.18 m (Agrawal 2005).

*Distance of Metal piece:*

Center of metal piece is around 100 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{100}{472.441} \times 1 = 0.212 \text{ m}$$

Distance of metal piece detected by the MATLAB algorithm is 0.212 m from the point of start of scan where as the actual distance is around 0.2 m (Agrawal 2005).

*Depth of Heavy Rotting:*

Peak of the heavy rotting lies near 100<sup>th</sup> row

$$\text{Depth of Defect} = 0.601 \times \frac{100}{512} = 0.117 \text{ m}$$

The depth of heavy rotting detected by the MATLAB algorithm is 0.117 m from the surface where as the RADAN detected depth is around 0.13 m (Agrawal 2005).

*Distance of Heavy Rotting:*

Center of heavy rotting is around 360 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{360}{472.441} \times 1 = 0.762 \text{ m}$$

Distance of heavy rotting detected by the MATLAB algorithm is 0.762 m from the point of start of scan where as the RADAN detected distance is around 0.75 m (Agrawal 2005).

Table 4-3 shows the comparison of depths and lengths of the defects detected by MATLAB, RADAN with actual depths and lengths of the defects.

Table 4-3 Results

Sl. No.	Defect	Actual	RADAN Detected	MATLAB Detected	Difference
1	Rot Depth	-	0.13 ~ 0.2 m	0.12 ~ 0.19 m	-0.01 m
2	Metal Depth	0.18 m	0.19 m	0.188 m	0.008 m
3	Metal Distance	0.2 m	0.2 m	0.212 m	0.012 m
4	Heavy rot Depth	-	0.13 m	0.117 m	-0.013 m
5	Heavy rot Distance	-	0.75 m	0.762 m	0.012 m

### 4.3 Log5 Data Processing using MATLAB

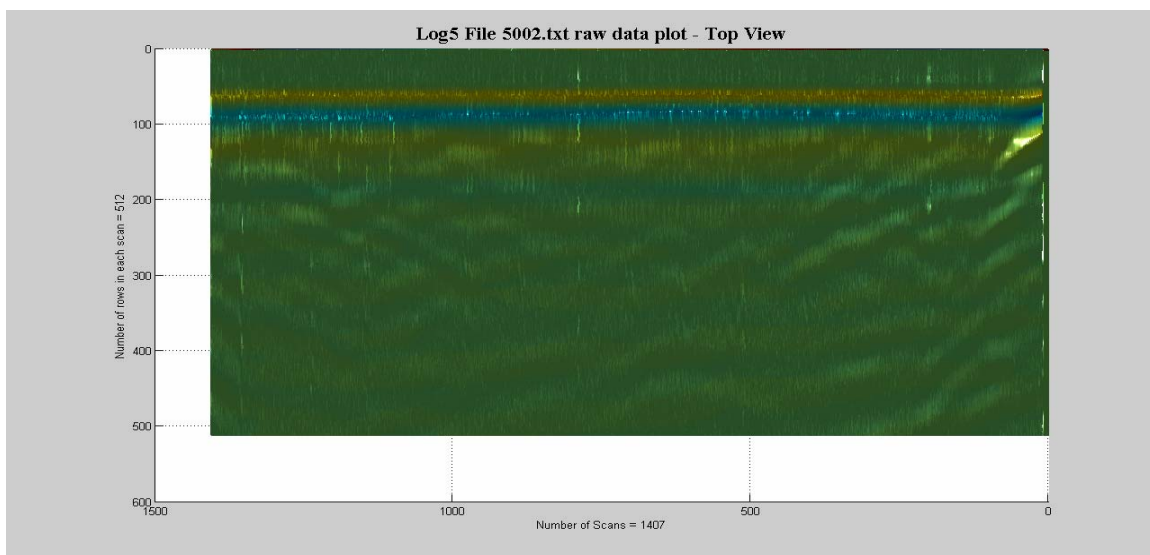


Figure 4-22 MATLAB ‘Surf’ plot of raw data of Log5

Log5 has been processed the same way as Log1. Figure 4-22 shows the surf plot of the raw data of Log5. Figure 4-23 shows the radar scan after applying High Pass Filter. The defects patterns are not seen clear in this scan.

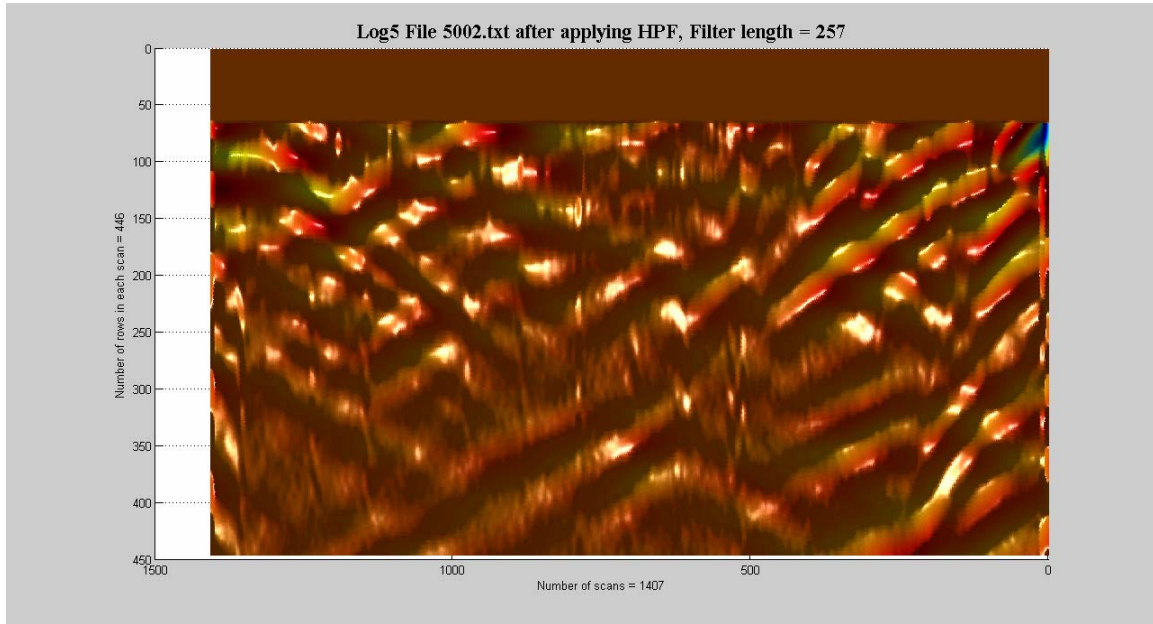


Figure 4-23 Radar scan after applying High Pass Filter

To make the defect patterns appear better, the tall areas in the scan has to be removed. Figure 4-24 shows the radar scan after removing these tall areas from the scan. This involved removal of first 60 columns from the scan.

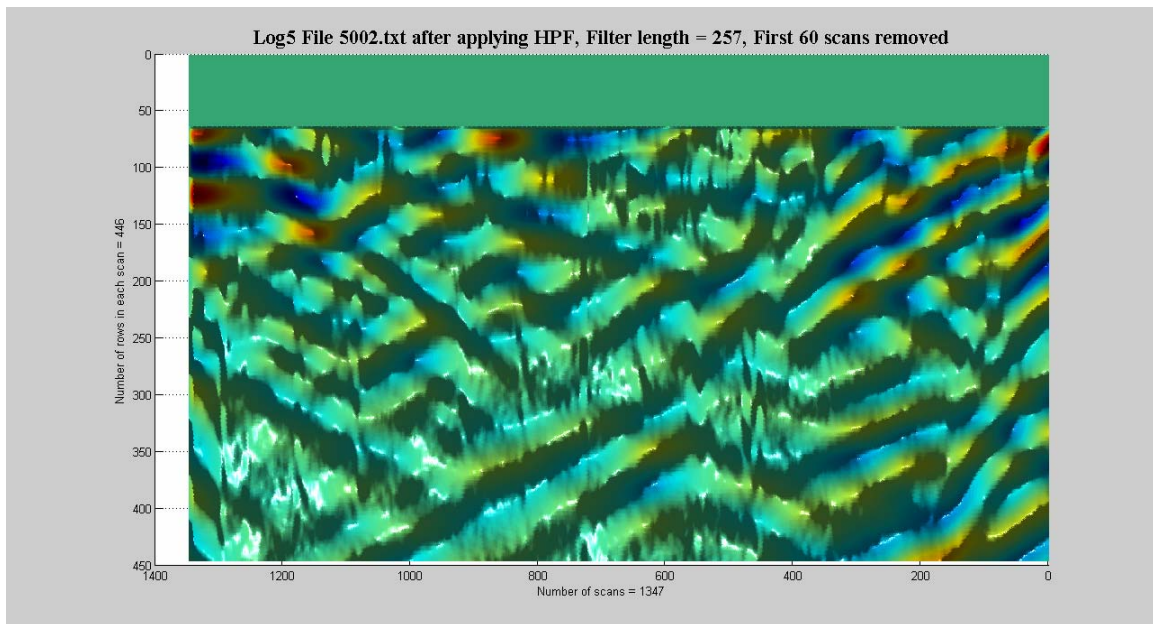


Figure 4-24 Radar scan after applying High Pass Filter & removing First 60 columns

The removal of tall areas has been done just for the purpose of presentation.

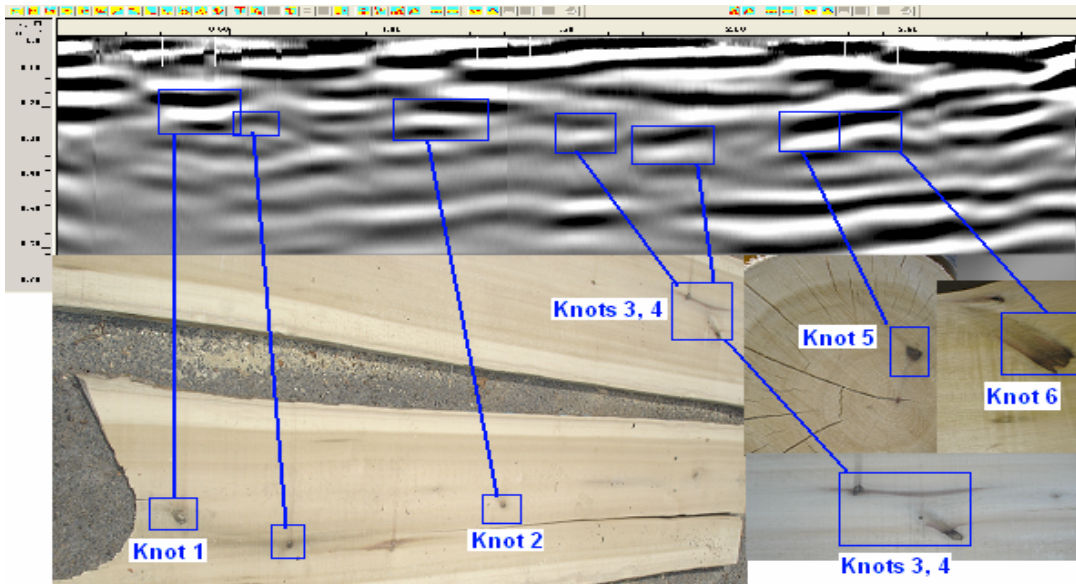


Figure 4-25 RADAN Processed GPR data of Log5 (Agrawal 2005)

Data processed using RADAN 5.0 software (Figure 4-25) has been given along with the high pass filter applied radar scan (Figure 4-24) for comparison. It can be seen from the Figures 4-24 and 4-25 that the results of MATLAB algorithm are in agreement with the results of RADAN software. Figure 4-26 shows the radar scan after applying threshold. Threshold value used is 150. Figure 4-26 contains both defects and noise. Defects have been identified in order to compare with the RADAN output.

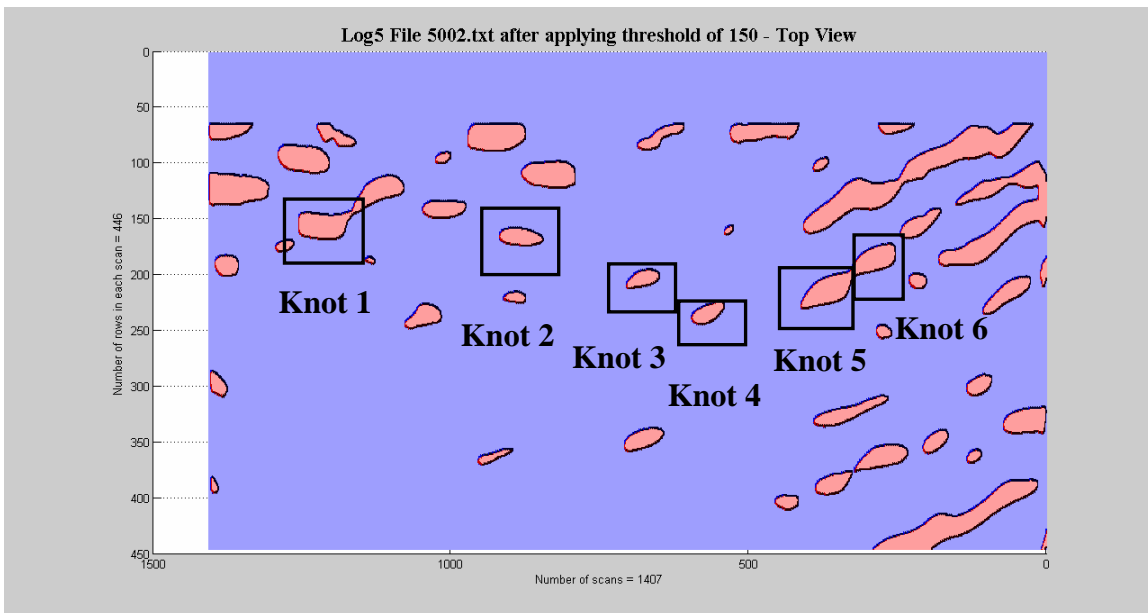


Figure 4-26 Radar scan showing defects for threshold value = 150

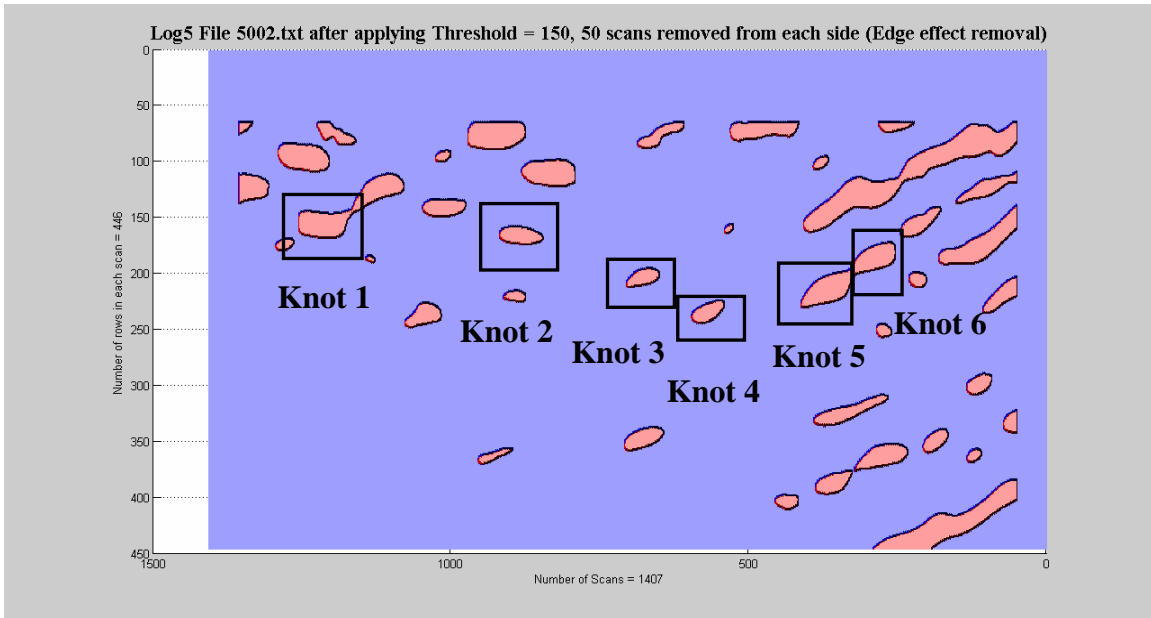


Figure 4-27 Radar scan showing defects for threshold value = 150, Edge effect removed

Figure 4-27 shows radar scan after removing edge effect. Some reduction in noise can be observed in this scan. Figure 4-28 shows the defects detected in radar scan by applying automatic threshold. Figure 4-28 also involves edge effect removal and check for bottom reflection removal. All the major defects have been identified by automatic threshold application. Automatic threshold value calculated is 285 where as the manual threshold value is 150.

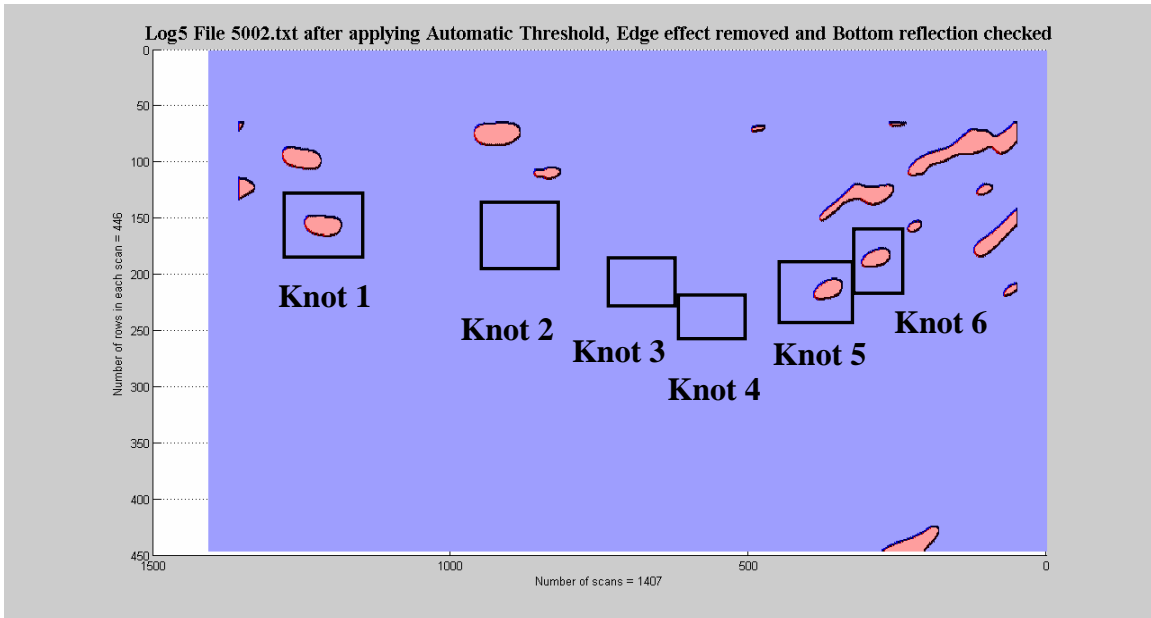


Figure 4-28 Radar scan showing defects for automatic threshold value = 285

For Log5,

Travel time or Range = 14 ns

Dielectric constant Value = 12

Substituting these in equation 3.12, we get the total depth traveled by signal

$$\begin{aligned}\text{Total depth (d)} &= \frac{3 \times 10^8 \times 14 \times 10^{-9}}{2\sqrt{12}} \\ &= 0.606 \text{ m}\end{aligned}$$

The 512 rows (data points) of each scan are collected in round trip travel time. So, 512 rows represent the total depth of 0.606 m. To find the depth of the defect, the ratio of the defect row with 512 has to be multiplied with the total depth. Figure 4-26 has been considered for calculating depth of the defects.

*Depth of Knot 1:*

Center of knot 1 lies near 165<sup>th</sup> row

$$\text{Depth of Defect} = 0.606 \times \frac{165}{512} = 0.2 \text{ m}$$

The depth of knot 1 detected by the MATLAB algorithm is 0.2 m from the surface where as the actual depth is around 0.24 m (Agrawal 2005).

*Distance of Knot 1:*

Center of knot 1 is around 180 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{180}{472.441} \times 1 = 0.38 \text{ m}$$

Distance of knot 1 detected by the MATLAB algorithm is 0.38 m from the point of start of scan where as the actual distance is around 0.4 m (Agrawal 2005).

*Depth of Knot 2:*

Center of knot 2 lies near 170<sup>th</sup> row

$$\text{Depth of Defect} = 0.606 \times \frac{170}{512} = 0.2 \text{ m}$$

The depth of knot 2 detected by the MATLAB algorithm is 0.2 m from the surface where as the actual depth is around 0.2 m (Agrawal 2005).

*Distance of Knot 2:*

Center of knot 2 is around 550 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{540}{472.441} \times 1 = 1.17 \text{ m}$$

Distance of knot 2 detected by the MATLAB algorithm is 1.17 m from the point of start of scan where as the actual distance is around 1.2 m (Agrawal 2005).

*Depth of Knot 3:*

Center of knot 3 lies near 210<sup>th</sup> row

$$\text{Depth of Defect} = 0.606 \times \frac{210}{512} = 0.25 \text{ m}$$

The depth of knot 3 detected by the MATLAB algorithm is 0.25 m from the surface where as the actual depth is around 0.27 m (Agrawal 2005).

*Distance of Knot 3:*

Center of knot 3 is around 725 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{725}{472.441} \times 1 = 1.53 \text{ m}$$

Distance of knot 3 detected by the MATLAB algorithm is 1.53 m from the point of start of scan where as the actual distance is around 1.53 m (Agrawal 2005).

*Depth of Knot 4:*

Center of knot 4 lies near 235<sup>th</sup> row

$$\text{Depth of Defect} = 0.606 \times \frac{235}{512} = 0.28 \text{ m}$$



The depth of knot 4 detected by the MATLAB algorithm is 0.28 m from the surface where as the actual depth is around 0.27 m (Agrawal 2005).

*Distance of Knot 4:*

Center of knot 4 is around 840 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{840}{472.441} \times 1 = 1.78 \text{ m}$$

Distance of knot 4 detected by the MATLAB algorithm is 1.78 m from the point of start of scan where as the actual distance is around 1.8 m (Agrawal 2005).

*Depth of Knot 5:*

Center of knot 5 lies near 220<sup>th</sup> row

$$\text{Depth of Defect} = 0.606 \times \frac{220}{512} = 0.26 \text{ m}$$

The depth of knot 5 detected by the MATLAB algorithm is 0.26 m from the surface where as the actual depth is around 0.23 m (Agrawal 2005).

*Distance of Knot 5:*

Center of knot 5 is around 1,030 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{1,030}{472.441} \times 1 = 2.18 \text{ m}$$

Distance of knot 5 detected by the MATLAB algorithm is 2.18 m from the point of start of scan where as the actual distance is around 2.2 m (Agrawal 2005).

*Depth of Knot 6:*

Center of knot 6 lies near 185<sup>th</sup> row

$$\text{Depth of Defect} = 0.606 \times \frac{185}{512} = 0.22 \text{ m}$$

The depth of knot 6 detected by the MATLAB algorithm is 0.22 m from the surface where as the actual depth is around 0.24 m (Agrawal 2005).

*Distance of Knot 6:*

Center of knot 6 is around 1,120 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{1,120}{472.441} \times 1 = 2.37 \text{ m}$$

Distance of knot 6 detected by the MATLAB algorithm is 2.37 m from the point of start of scan where as the actual distance is around 2.4 m (Agrawal 2005).

Table 4-4 shows the comparison of depths and lengths of the defects detected by MATLAB, RADAN with actual depths and lengths of the defects.

Table 4-4 Results

Sl. No.	Defect	Actual (m)	RADAN Detected (m)	MATLAB Detected (m)	Difference (m)
1	Knot 1 Depth	0.24	0.20	0.20	-0.04
2	Knot 1 Distance	0.40	0.40	0.38	-0.02
3	Knot 2 Depth	0.20	0.20	0.20	0.00
4	Knot 2 Distance	1.20	1.22	1.17	-0.03
5	Knot 3 Depth	0.27	0.25	0.25	-0.02
6	Knot 3 Distance	1.53	1.53	1.53	0.00
7	Knot 4 Depth	0.27	0.27	0.28	0.01
8	Knot 4 Distance	1.80	1.80	1.78	-0.02
9	Knot 5 Depth	0.23	0.26	0.26	0.03
10	Knot 5 Distance	2.20	2.20	2.18	-0.02
11	Knot 6 Depth	0.24	0.25	0.22	-0.02
12	Knot 6 Distance	2.40	2.40	2.37	-0.03
13	Metal Depth	0.26	-	-	-
14	Metal Distance	0.60	-	-	-

#### 4.4 Log6 Data Processing using MATLAB

Log6 has been processed the same way as Log1.

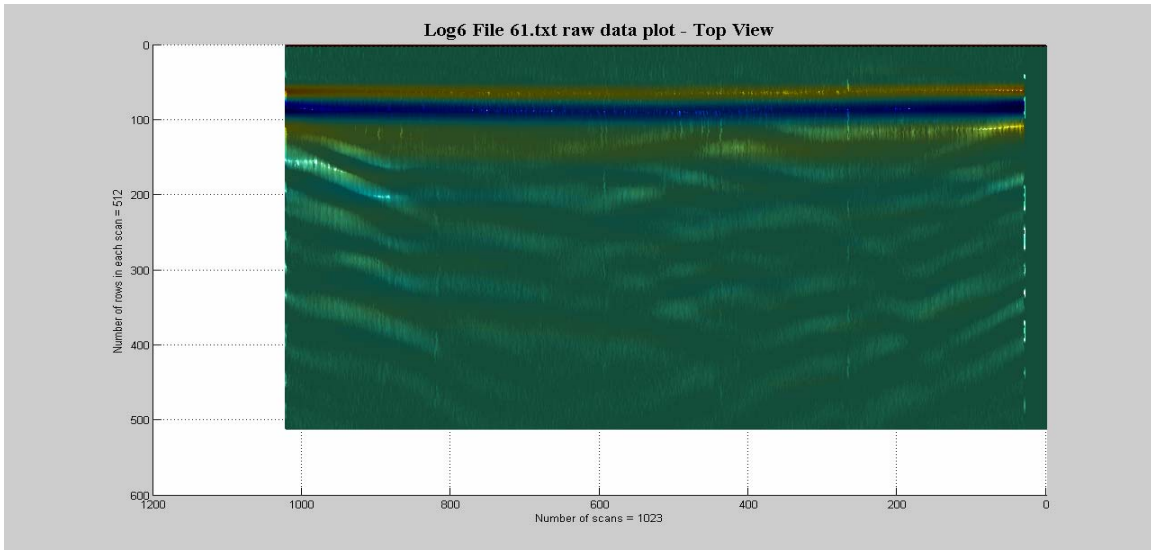


Figure 4-29 MATLAB 'Surf' plot of raw data of Log6

Figure 4-29 shows the surf plot of the raw data of Log6. Figure 4-30 shows the radar scan after applying High Pass Filter. The defects patterns are seen clear in this scan.

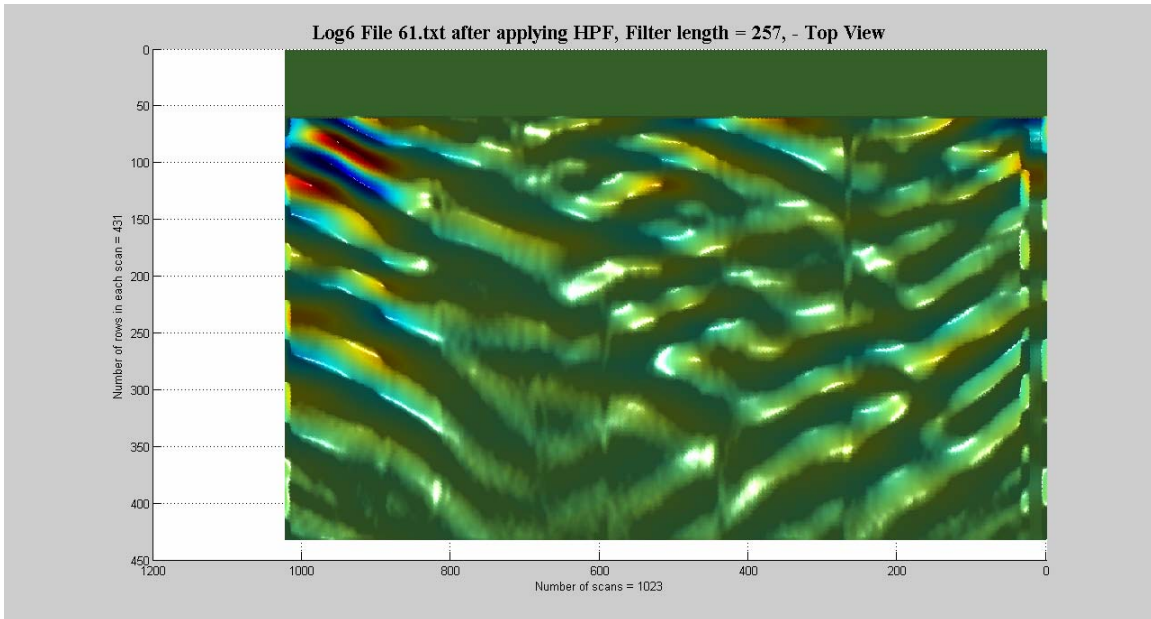


Figure 4-30 Radar scan after applying High Pass Filter

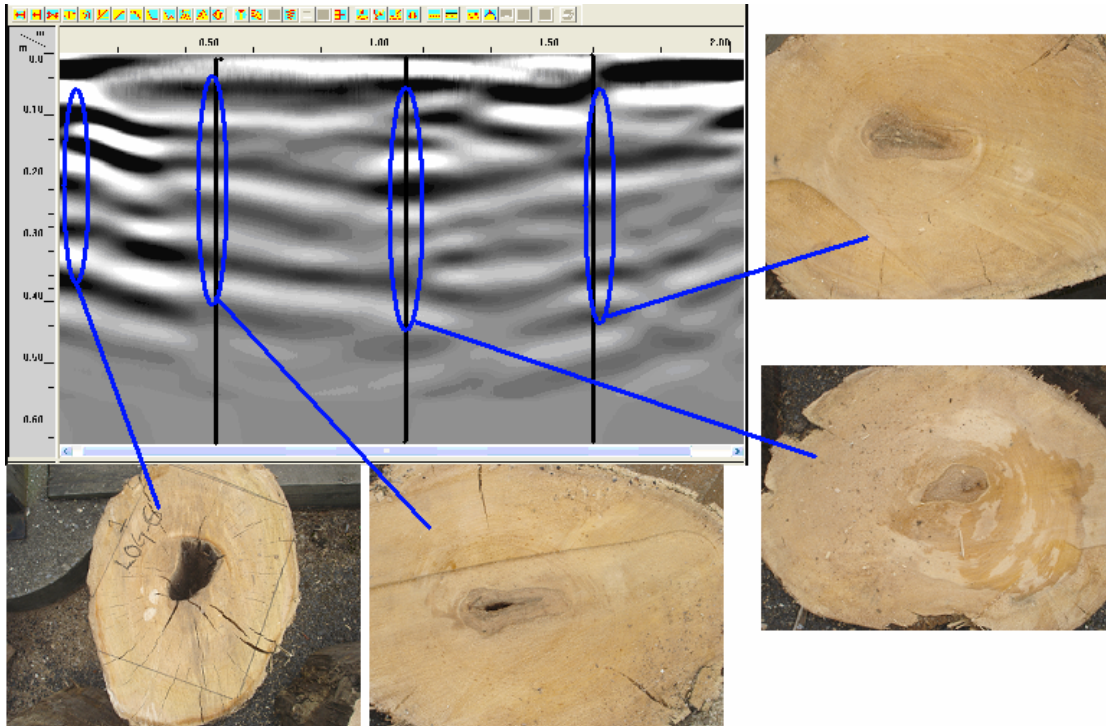


Figure 4-31 RADAN Processed GPR data of Log6 (Agrawal 2005)

Data processed using RADAN 5.0 software (Figure 4-31) has been given along with the high pass filter applied radar scan (Figure 4-30) for comparison. It can be seen from the Figures 4-30 and 4-31 that the results of MATLAB algorithm are in agreement with the results of RADAN software.

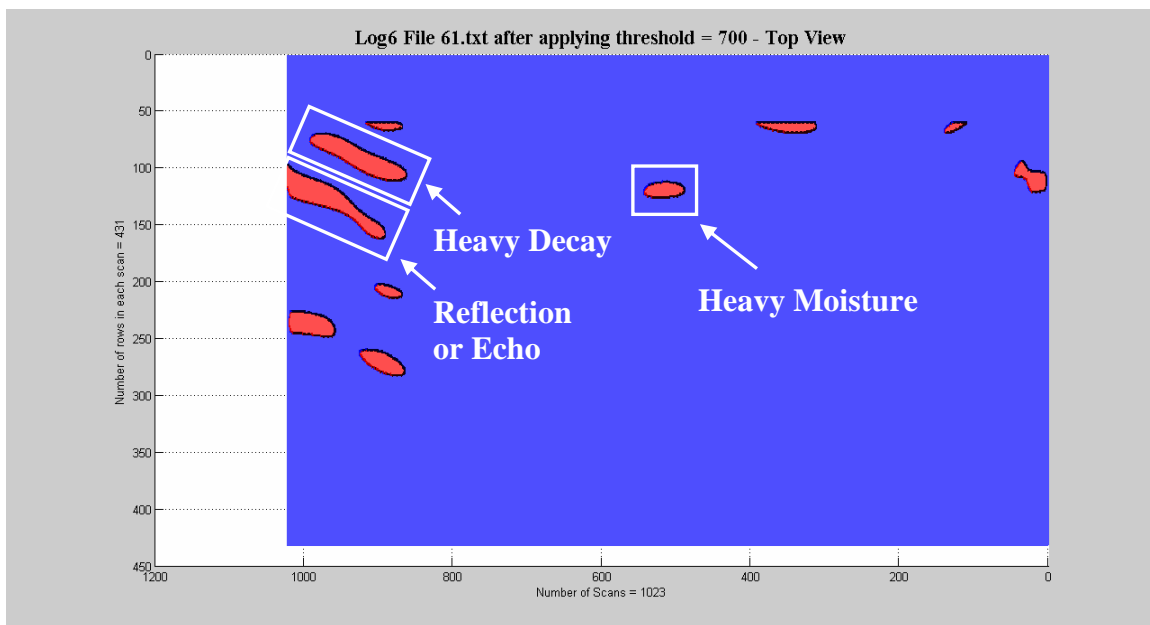


Figure 4-32 Radar scan showing defects for threshold value = 700

Figure 4-32 shows the radar scan after applying threshold. Threshold value used is 700. Figure 4-32 contains both defects and noise. Defects have been identified in order to compare with the RADAN output.

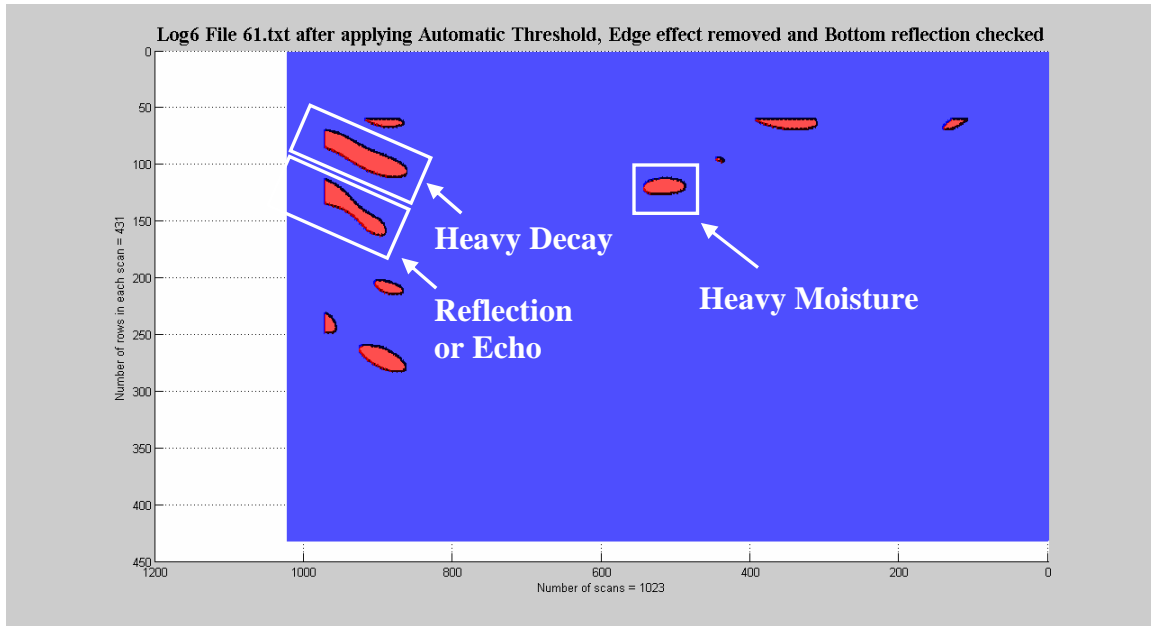


Figure 4-33 Radar scan showing defects for automatic threshold value = 685

Figure 4-33 shows the defects detected in radar scan by applying automatic threshold. Figure 4-33 also involves edge effect removal and check for bottom reflection removal. All the major defects have been identified by automatic threshold application. Automatic threshold value calculated is 685 where as the manual threshold value is 700. Radar scan showing edge effect removal for manual threshold value is not done for this log since the automatic threshold value is closer to manual threshold value.

For Log6,

Travel time or Range = 13 ns

Dielectric constant Value = 9

Substituting these in equation 3.12, we get the total depth traveled by signal

$$\begin{aligned} \text{Total depth (d)} &= \frac{3 \times 10^8 \times 13 \times 10^{-9}}{2\sqrt{9}} \\ &= 0.65 \text{ m} \end{aligned}$$

The 512 rows (data points) of each scan are collected in round trip travel time. So, 512 rows represent the total depth of 0.65 m. To find the depth of the defect, the ratio of the defect row with 512 has to be multiplied with the total depth. Figure 4-32 has been considered for calculating depth of the defects.

*Depth of Heavy decay:*

Center of heavy decay lies near 110<sup>th</sup> row

$$\text{Depth of Defect} = 0.65 \times \frac{110}{512} = 0.14 \text{ m}$$

The depth of heavy decay detected by the MATLAB algorithm is 0.14 m from the surface where as the actual depth is around 0.15 m (Agrawal 2005).

*Distance of Heavy decay:*

Heavy decay extends till 190<sup>th</sup> column from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the length of heavy decay is,

$$\text{Length of Heavy Decay} = \frac{190}{472.441} \times 1 = 0.4 \text{ m}$$

Length of heavy decay detected by the MATLAB algorithm is 0.4 m where as the actual length is around 0.4 m (Agrawal 2005).

*Depth of High Moisture Content Region:*

Center of high moisture content region lies near 125<sup>th</sup> row

$$\text{Depth of Defect} = 0.65 \times \frac{125}{512} = 0.16 \text{ m}$$

The depth of high moisture content region detected by the MATLAB algorithm is 0.16 m from the surface where as the RADAN detected depth is also 0.16 m (Agrawal 2005).

*Distance of High Moisture Content Region:*

Center of high moisture content region is around 510 columns from the edge of the scan. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the distance of the defect from the point of start of the scan is,

$$\text{Distance of Defect} = \frac{510}{472.441} \times 1 = 1.08 \text{ m}$$

Distance of high moisture content region detected by the MATLAB algorithm is 1.08 m from the point of start of scan where as the actual distance is around 1.1 m (Agrawal 2005).

Table 4-5 shows the comparison of depths and lengths of the defects detected by MATLAB, RADAN with actual depths and lengths of the defects.

Table 4-5 Results

<b>Sl. No.</b>	<b>Defect</b>	<b>Actual (m)</b>	<b>RADAN Detected (m)</b>	<b>MATLAB Detected (m)</b>	<b>Difference (m)</b>
1	Heavy Decay Depth	0.15	0.15	0.14	-0.01
2	Heavy Decay Length	0.4	0.4	0.4	0.0
3	High Moisture Depth	-	0.16	0.16	0.0
4	High Moisture Distance	1.1	1.1	1.08	-0.02

## Chapter 5

### DEFECT DETECTION IN LOG4 USING MATLAB ALGORITHM

#### 5.1 Log4 Data Processing using MATLAB

Log4 has been processed differently from Log1. Figure 5-1 shows the surf plot of the raw data of Log4.

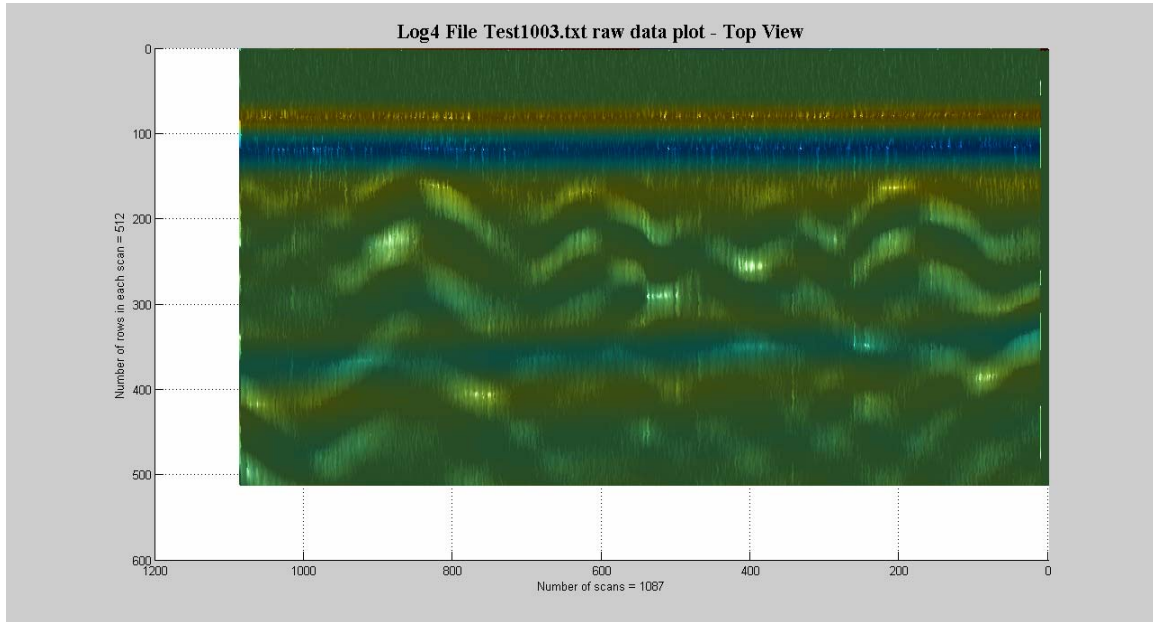


Figure 5-1 MATLAB ‘Surf’ plot of raw data of Log4

The first difference in processing is the position of signal clipping. In Log4 the metal pieces were present closer to the surface of the log (Agrawal 2005) and the reflections from these metal pieces were so strong that they merged with the reflection from air / wood interface. So, the air / wood interface reflection cannot be clipped fully as in Log1. Hence Log4 is clipped only till 1<sup>st</sup> minimum where as Log1 is clipped till 2<sup>nd</sup> maximum with respect to Figure 3-2. The other processing steps like initial noise removal, subtracting from grand average, copying and adding edges, applying linear gain, zero correction, applying Low and High Pass filter are done in the same way as Log1. Figure 5-2 shows the radar scan after applying High Pass Filter. The defects patterns are not seen very clear in this scan. To make the defect patterns appear better, the tall areas in the scan has to be removed. Figure 5-3 shows the radar scan after removing these tall areas from the scan. This involved removal of first 20 columns from the scan. The



removal of tall areas has been done just for the purpose of presentation. The interesting characteristic of this scan is the multiple reflections or echoes of the signals. Echoes have been created since the amplitude of the signals reflected from the metal are very high. The same phenomenon can be observed in Log3 and Log6 also. Defects cannot be pinpointed in this log just by applying threshold as done in other logs due these echoes.

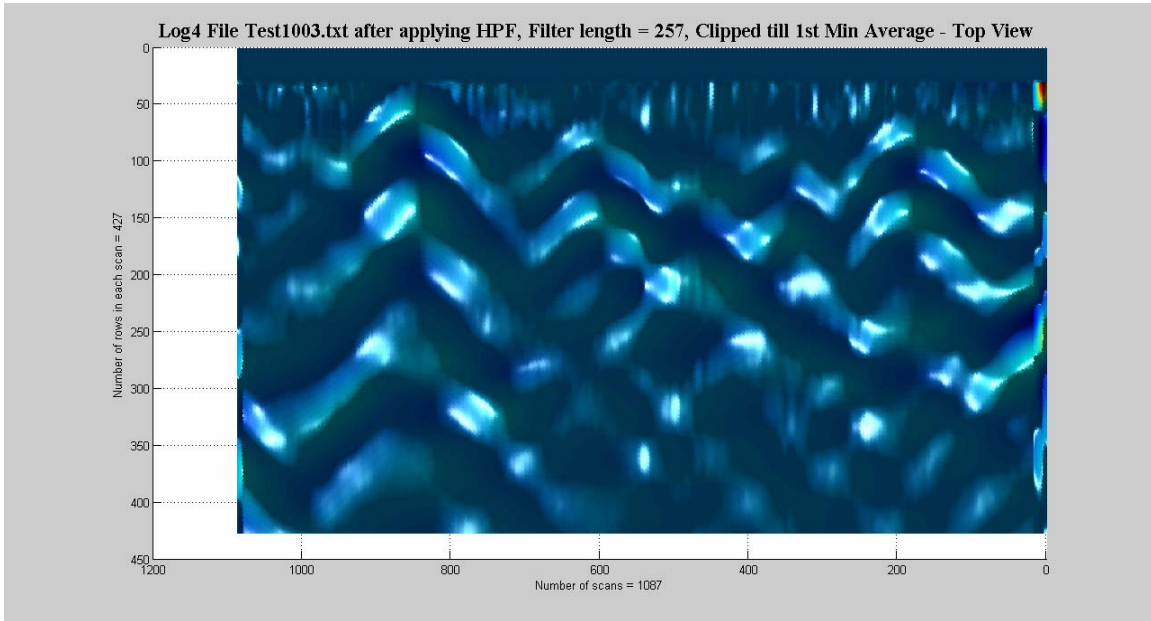


Figure 5-2 Radar scan after applying High Pass Filter

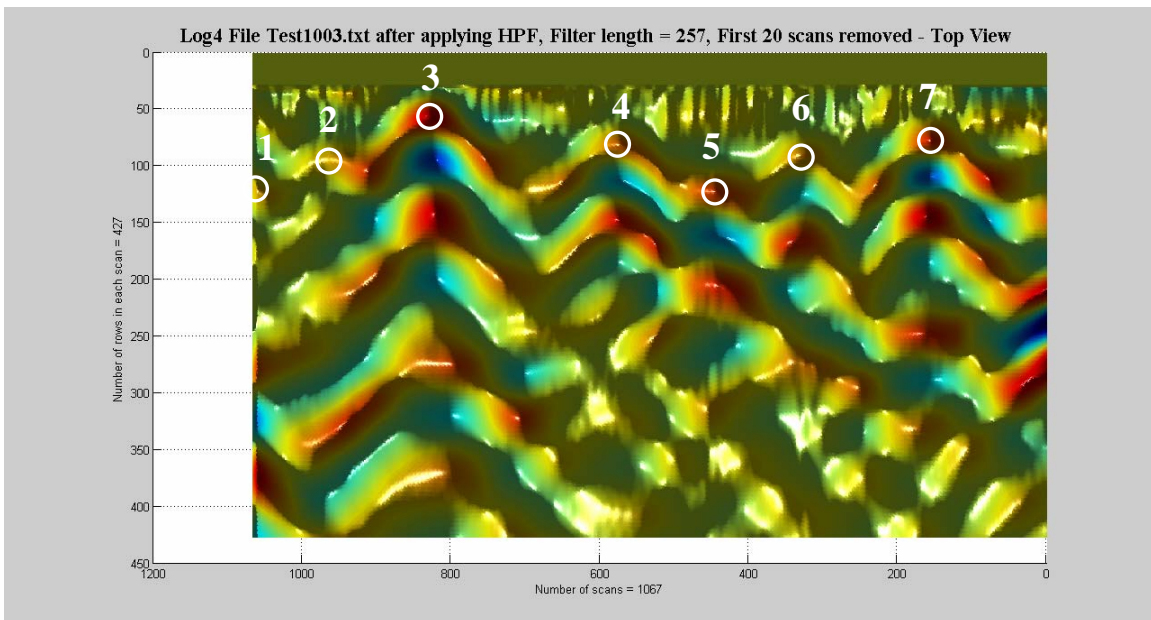


Figure 5-3 Radar scan after applying High Pass Filter & removing First 20 columns

Data processed using RADAN 5.0 software (Figure 5-4) has been given along with the high pass filter applied radar scan (Figure 5-3) for comparison. It can be seen from the Figures 5-3 and 5-4 that the results of MATLAB algorithm are in agreement with the results of RADAN software.

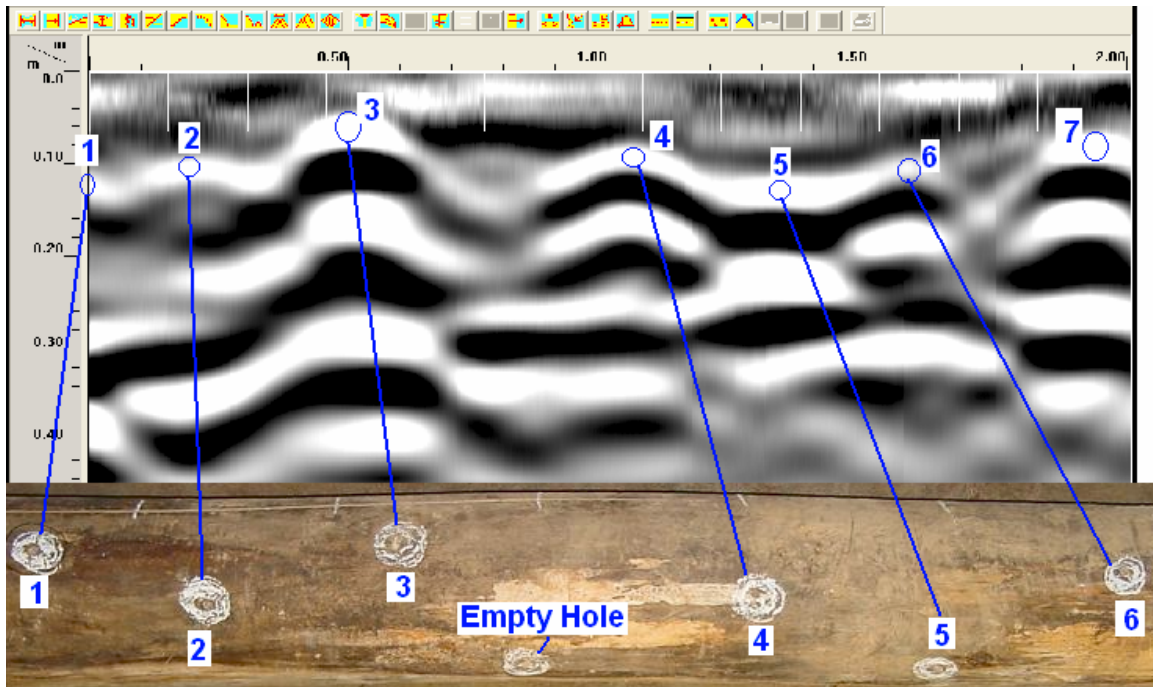


Figure 5-4 RADAN Processed GPR data of Log4 (Agrawal 2005)

For Log4,

Travel time or Range = 8 ns

Dielectric constant Value = 5.2

Substituting these in equation 3.12, we get the total depth traveled by signal

$$\begin{aligned} \text{Total depth (d)} &= \frac{3 \times 10^8 \times 8 \times 10^{-9}}{2\sqrt{5.2}} \\ &= 0.526 \text{ m} \end{aligned}$$

The 512 rows (data points) of each scan are collected in round trip travel time. So, 512 rows represent the total depth of 0.526 m. To find the depth of the defect, the ratio of the defect row with 512 has to be multiplied with the total depth. Figure 5-3 has been considered for calculating depth of the defects.

*Depth of Metal 2:*

Center of metal 2 lies near 95<sup>th</sup> row

$$\text{Depth of Defect} = 0.526 \times \frac{95}{512} = 0.10 \text{ m}$$

The depth of metal 2 detected by the MATLAB algorithm is 0.1 m from the surface where as the actual depth is around 0.1 m (Agrawal 2005).

*Depth of Metal 3:*

Center of metal 3 lies near 55<sup>th</sup> row

$$\text{Depth of Defect} = 0.526 \times \frac{55}{512} = 0.06 \text{ m}$$

The depth of metal 3 detected by the MATLAB algorithm is 0.06 m from the surface where as the actual depth is around 0.05 m (Agrawal 2005).

*Depth of Metal 4:*

Center of metal 4 lies near 80<sup>th</sup> row

$$\text{Depth of Defect} = 0.526 \times \frac{80}{512} = 0.08 \text{ m}$$

The depth of metal 4 detected by the MATLAB algorithm is 0.08 m from the surface where as the actual depth is around 0.1 m (Agrawal 2005).

*Depth of Metal 5:*

Center of metal 5 lies near 125<sup>th</sup> row

$$\text{Depth of Defect} = 0.526 \times \frac{125}{512} = 0.13 \text{ m}$$

The depth of metal 5 detected by the MATLAB algorithm is 0.13 m from the surface where as the actual depth is around 0.17 m (Agrawal 2005).

*Depth of Metal 6:*

Center of metal 6 lies near 90<sup>th</sup> row

$$\text{Depth of Defect} = 0.526 \times \frac{90}{512} = 0.09 \text{ m}$$

The depth of metal 6 detected by the MATLAB algorithm is 0.09 m from the surface where as the actual depth is around 0.1 m (Agrawal 2005).

*Depth of Metal 7:*

Center of metal 7 lies near 70<sup>th</sup> row

$$\text{Depth of Defect} = 0.526 \times \frac{70}{512} = 0.07 \text{ m}$$

The depth of metal 7 detected by the MATLAB algorithm is 0.07 m from the surface where as the actual depth is around 0.1 m (Agrawal 2005).

Table 5-1 shows the comparison of depths of the defects detected by MATLAB, RADAN with actual depths of the defects.

Table 5-1 Results

<b>Sl. No.</b>	<b>Defect</b>	<b>Actual (m)</b>	<b>RADAN Detected (m)</b>	<b>MATLAB Detected (m)</b>	<b>Difference (m)</b>
1	Metal 1	0.05	-	-	
2	Metal 2	0.10	0.11	0.10	0.0
3	Metal 3	0.05	0.06	0.06	0.01
4	Metal 4	0.10	0.09	0.08	-0.02
5	Metal 5	0.17	0.14	0.13	-0.04
6	Metal 6	0.10	0.11	0.09	-0.01
7	Metal 7	0.10	0.08	0.07	-0.03

*Spacing between Metal 2 & 3:*

Spacing between metal 2 & 3 is around 140 columns. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the spacing between metal 2 & 3 in meter is,

$$\text{Spacing between defects} = \frac{140}{472.441} \times 1 = 0.3 \text{ m}$$

Spacing between metal 2 & 3 detected by the MATLAB algorithm is 0.3 m where as the actual spacing is around 0.28 m (Agrawal 2005).

*Spacing between Metal 3 & 4:*

Spacing between metal 3 & 4 is around 250 columns. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the spacing between metal 3 & 4 in meter is,

$$\text{Spacing between defects} = \frac{250}{472.441} \times 1 = 0.53 \text{ m}$$

Spacing between metal 3 & 4 detected by the MATLAB algorithm is 0.53 m where as the actual spacing is around 0.52 m (Agrawal 2005).

*Spacing between Metal 4 & 5:*

Spacing between metal 4 & 5 is around 140 columns. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the spacing between metal 4 & 5 in meter is,

$$\text{Spacing between defects} = \frac{140}{472.441} \times 1 = 0.3 \text{ m}$$

Spacing between metal 4 & 5 detected by the MATLAB algorithm is 0.3 m where as the actual spacing is around 0.29 m (Agrawal 2005).

*Spacing between Metal 5 & 6:*

Spacing between metal 5 & 6 is around 115 columns. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the spacing between metal 5 & 6 in meter is,

$$\text{Spacing between defects} = \frac{115}{472.441} \times 1 = 0.24 \text{ m}$$

Spacing between metal 5 & 6 detected by the MATLAB algorithm is 0.24 m where as the actual spacing is around 0.26 m (Agrawal 2005).

*Spacing between Metal 6 & 7:*

Spacing between metal 6 & 7 is around 180 columns. Scans taken per meter of the log are 472.441 (Agrawal 2005). From this, the spacing between metal 6 & 7 in meter is,

$$\text{Spacing between defects} = \frac{180}{472.441} \times 1 = 0.38 \text{ m}$$

Spacing between metal 6 & 7 detected by the MATLAB algorithm is 0.38 m where as the actual spacing is around 0.4 m (Agrawal 2005).

Table 5-2 shows the comparison of spacing of the defects detected by MATLAB, RADAN with actual spacing of the defects.

Table 5-2 Results

Sl. No.	Spacing between	Actual (m)	RADAN Detected (m)	MATLAB Detected (m)	Difference (m)
1	Metal 1 & 2	0.27	-	-	-
2	Metal 2 & 3	0.28	0.30	0.30	0.02
3	Metal 3 & 4	0.52	0.54	0.53	0.01
4	Metal 4 & 5	0.29	0.30	0.30	0.01
5	Metal 5 & 6	0.26	0.24	0.24	-0.02
6	Metal 6 & 7	0.40	0.38	0.38	-0.02

## 5.2 Algorithm to remove echoes

The echoes in the scan have to be removed in order to pin point defects. An attempt has been made to write MATLAB code to remove these echoes. This code works when the remove reflections option is selected. This code removes all the echoes leaving the first reflection. To do this, first the echoes have to be separated. This is done by applying initial threshold to the scan. Figure 5-5 shows the radar scan after applying a threshold of 300. Separation of echoes can be observed in Figure 5-5.

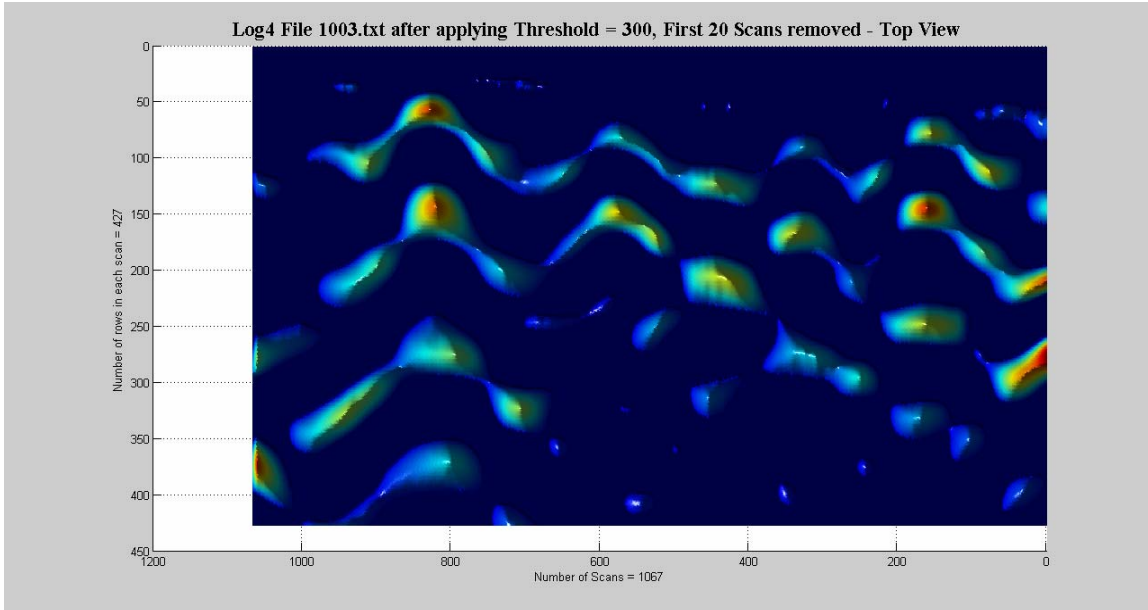


Figure 5-5 Radar scan after applying a threshold of 300 showing separation of echoes

Figure 5-6 shows radar scan after removing echoes. Some small chunks are left over from the echoes which didn't have any reflection in front of them.

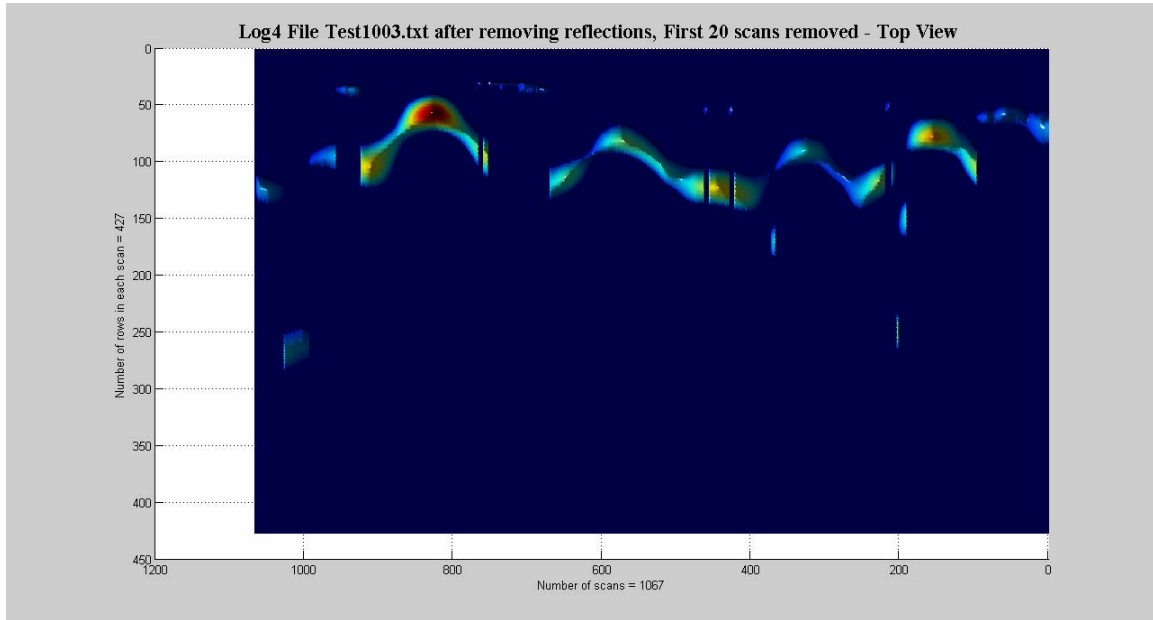


Figure 5-6 Radar scan after removing echoes showing left over small chunks

These small chunks have been further removed by MATLAB code. This code selects isolated chunks and deletes them if they are smaller than a particular threshold (here threshold means number of points in a chunk not the amplitude value). Here the threshold value is 1500. Figure 5-7 shows radar scan after removing small chunks.

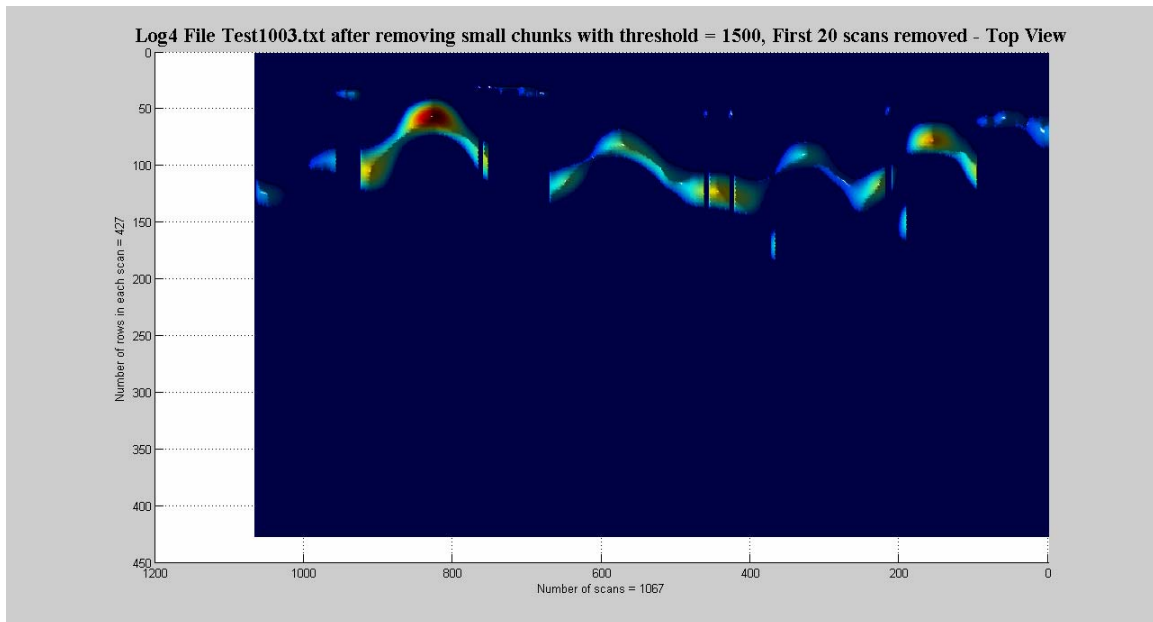


Figure 5-7 Radar scan after removing small chunks

The defects are further pin pointed by applying final threshold. Figure 5-8 shows radar scan after applying a threshold of 700.

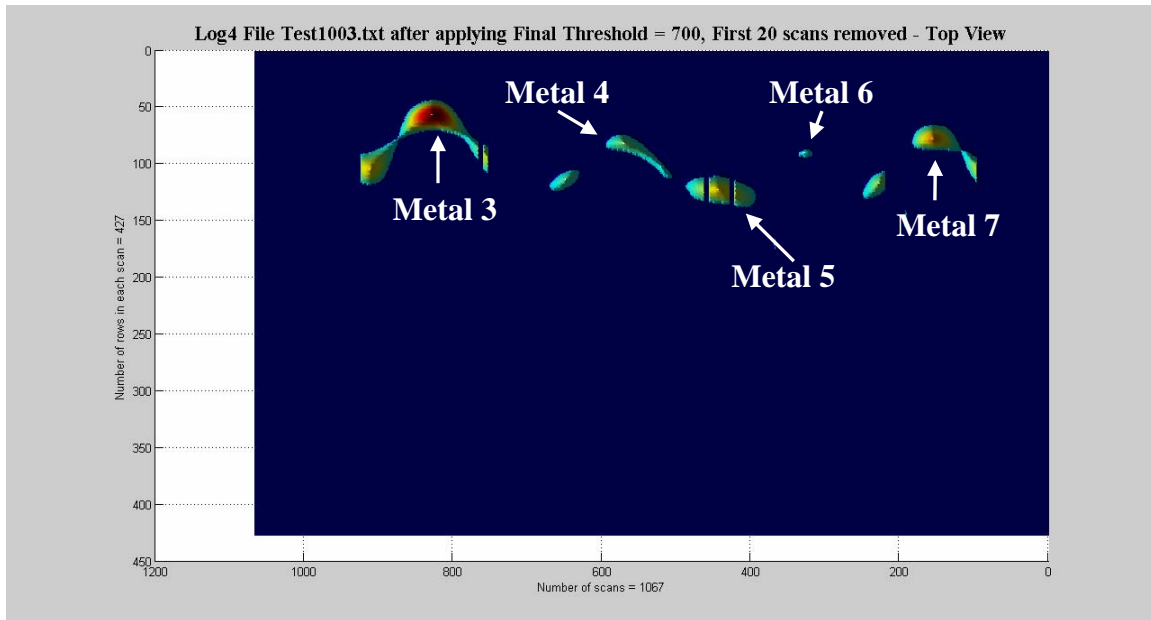


Figure 5-8 Radar scan after applying a threshold of 700

Metal defects 3,4,5,6 and 7 can be seen from Figure 5-8. Metal defect 2 is missing in this scan since it has very low amplitude value.

### 5.3 Classification of defects

Classification of defects into metals, rots, knots etc. is not possible due to the following reasons.

1. Rots and knots neither form any particular pattern nor have any particular amplitude. So, there classification is not possible
2. Metals with proper orientation form a hyperbolic shape. But there are even other defects which form patterns similar to hyperbolic shape. For example in Log1, the reflection from wood / air interface has formed a hyperbolic shape. Similarly, in Log2 in the first scan, the reflection from air/wood interface has formed a hyperbolic shape. In Log3, the main reflection from the rot forms a hyperbolic shape. In test log, in the scan along line 5 the reflection from wood / air interface has formed a hyperbolic shape. So, defects other than metals will also be detected as metals in these cases.



#### 5.4 Threshold Value Selection

Threshold values were selected by trial and error for all the logs in the beginning of this research. The range of threshold values is 140 to 700. This is a wide range and the threshold value used for one log is different from another log. Different threshold values were used for different logs since the strength of the signals reflected from different defects / logs were different. So, trial and error method of threshold value selection will not work in factory setup. Threshold value can be selected by knowing the strength of the signal.

Strength of the signal can be known in different ways. The global maximum of the scan gives some idea about the strength of the signal. But there are many chances that the global maximum will have very high value compared to the rest of the signal for example when there is a metal or edge effect. The other way is to find out the average of the entire scan. But the average of the entire scan will be closer to zero after filtering. The other alternative is to pick the maximums from each column and take an average of all these maximums and use it as a threshold value. Threshold values are calculated by this method and the defects are detected using this threshold value for different logs. These defects detected scans are shown along with the other scans of each log. Table 5-3 shows the comparison between the threshold values selected by trial and error (manual) method and taking average of maximums (automatic) method.

Table 5-3 Comparison between Threshold Values

Sl. No.	Log	Manual	Automatic
1	Log1	700	552
2	Log2	140	186
3	Log2 Line5	150	197
4	Log3	350	893
5	Log4	300, 700	NA
6	Log5	150	285
7	Log6	700	685
8	Log11_Line5	200	202
9	Test Log Line5	25	43
10	Test Log Line12	25	52

All the major defects have been detected by automatic threshold selection method. Threshold values from both the methods are closer except for Log3. This technique can be validated further by trying on good number of logs. There is one draw back with this method. The defects will be detected even in a good log. So, to overcome this, a lower limit has to be established for the threshold value below which, the defects will not be detected. Lower limit can be established based on good number of samples. The threshold value obtained from automatic method can be adjusted by adding or subtracting a constant value if required.

## Chapter 6

### RESULTS OF THE TEST CONDUCTED ON ALGORITHM

MATLAB Algorithm was tested on 10/30/06 in forestry lab. A wooden log was scanned from two different angles and the collected data was processed using MATLAB algorithm. Figure 6-1 shows the log and the antenna mounted on it for scanning.



Figure 6-1 Log and Antenna for testing MATLAB Algorithm

#### 6.1 Test Log along Mark 5

Figure 6-2 shows the surf plot of the raw data of the GPR scan through Mark 5 of the log. Some features can be observed in this plot.

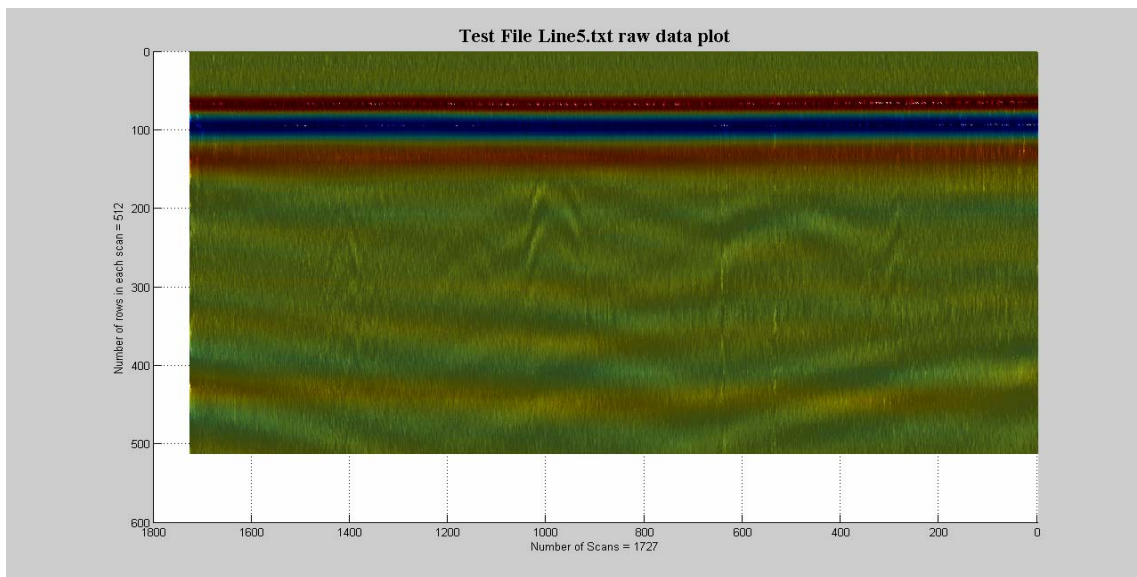


Figure 6-2 Raw data plot of GPR scan through Mark 5 of the test log

This test log has been processed the same way as Log1. Figure 6-3 shows the radar scan after applying High Pass Filter. The defects patterns are seen clear in this scan.

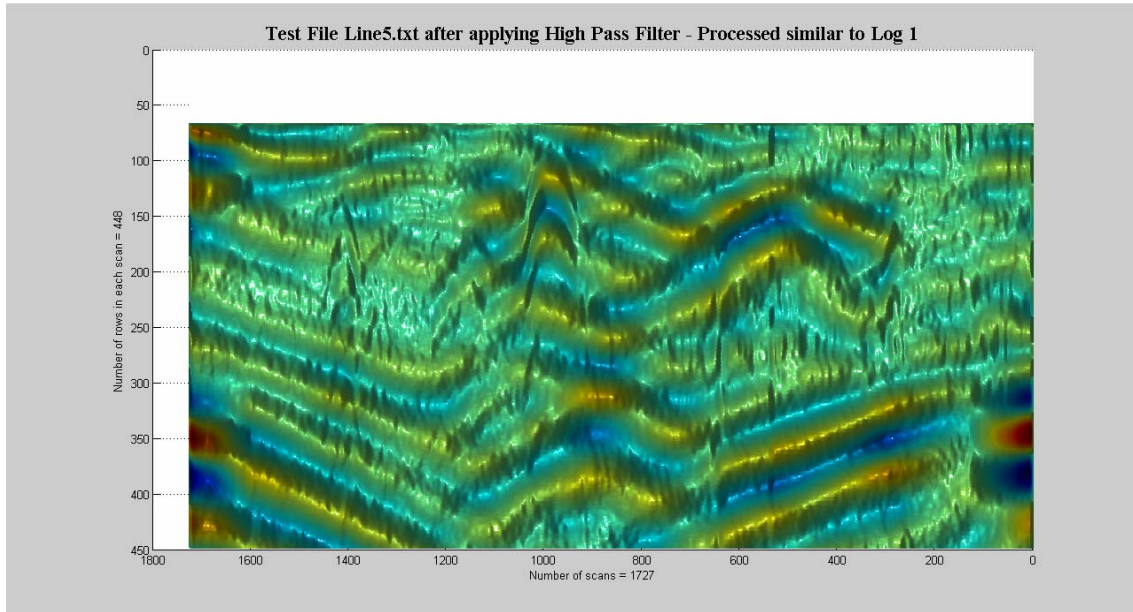


Figure 6-3 Radar scan through Mark 5 after applying High Pass Filter

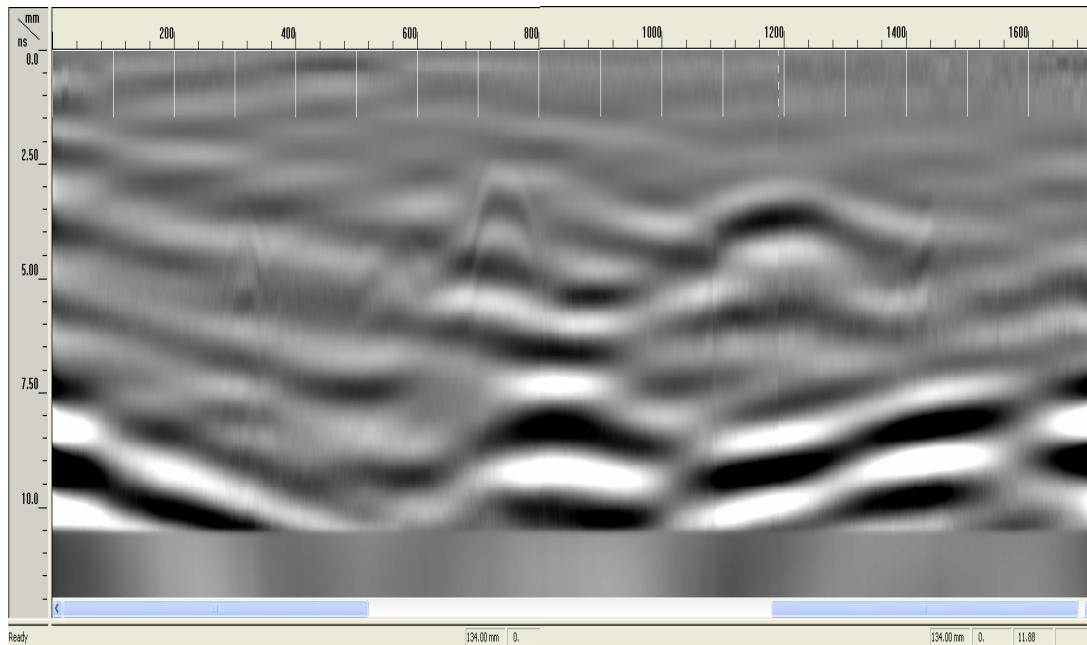


Figure 6-4 RADAN Processed GPR data of Test Log along Mark 5

Data processed using RADAN 5.0 software (Figure 6-4) has been given along with the high pass filter applied radar scan (Figure 6-3) for comparison. It can be seen from the Figures 6-3 and 6-4 that the results of MATLAB algorithm are in agreement with the results of RADAN software.

Figure 6-5 shows the radar scan after applying threshold. Threshold value used is 25. Figure 6-5 contains both defects and noise.

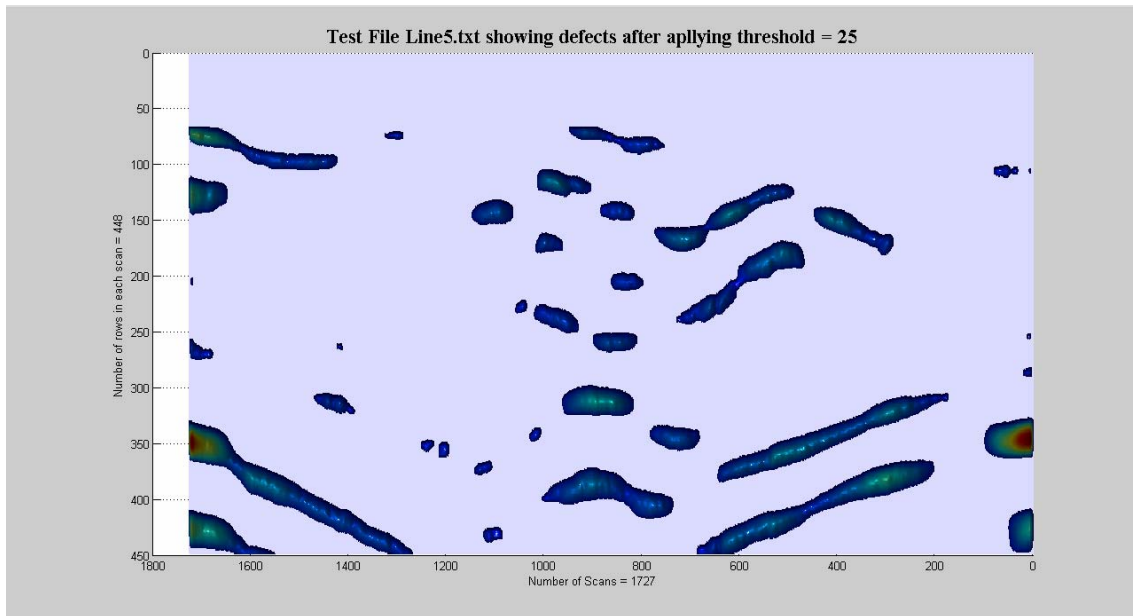


Figure 6-5 Radar scan showing defects for threshold value = 25

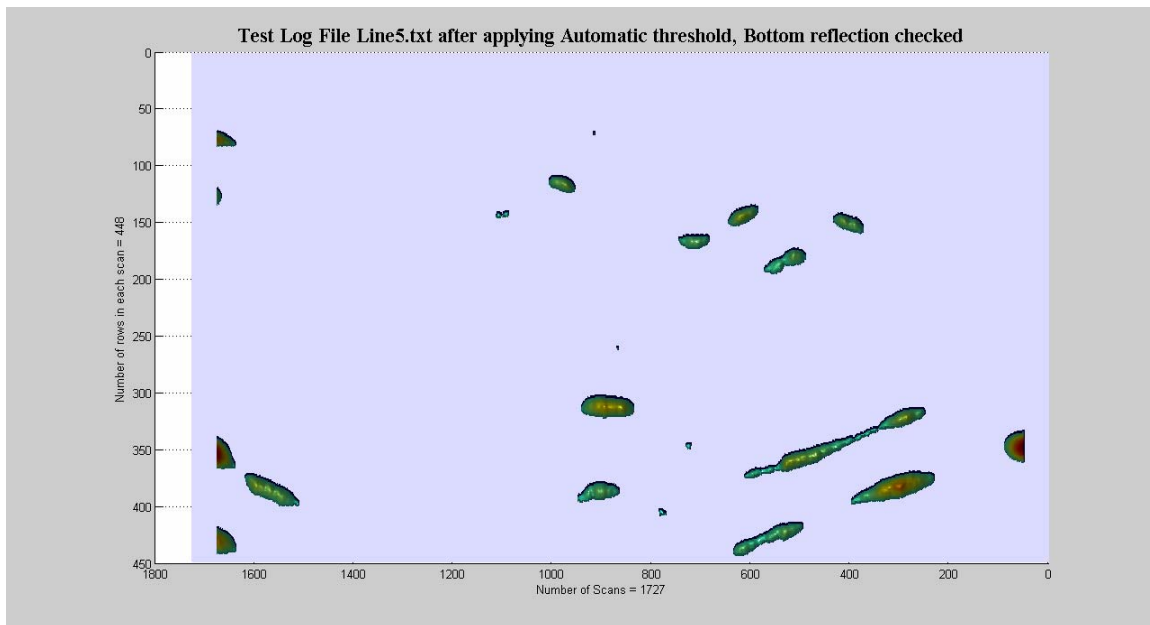


Figure 6-6 Radar scan showing defects for automatic threshold value = 43

Figure 6-6 shows the defects detected in radar scan by applying automatic threshold. Figure 6-6 also involves edge effect removal and check for bottom reflection removal. All the major defects have been identified by automatic threshold application. Automatic threshold value calculated is 43 where as the manual threshold value is 25.

## 6.2 Test Log along Mark 12

Figure 6-7 shows the surf plot of the raw data of the GPR scan through Mark 12 of the log. Some features can be observed in this plot.

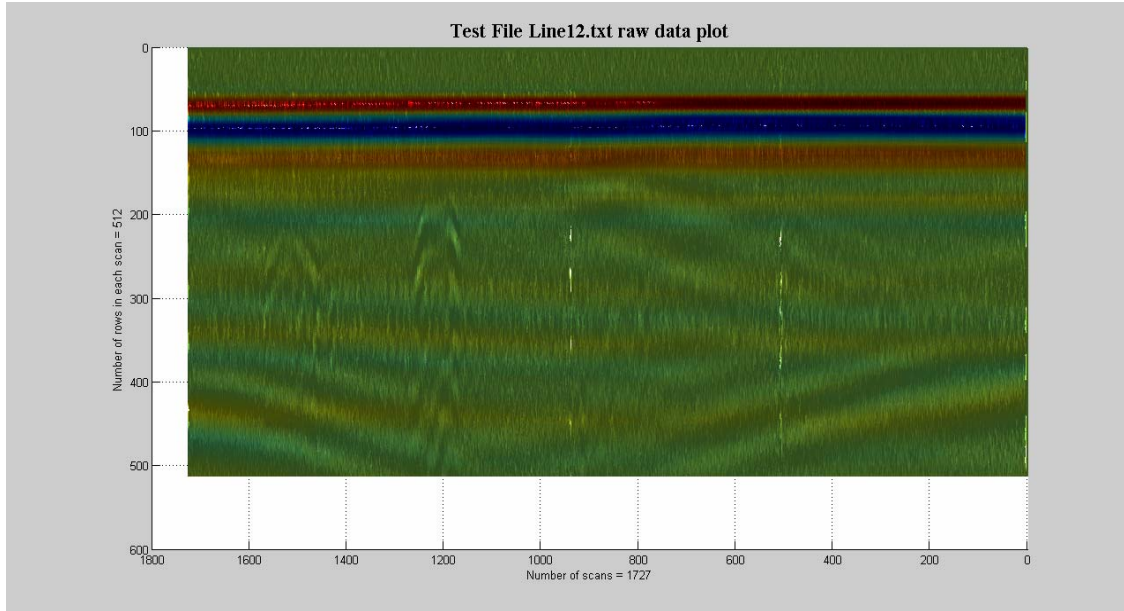


Figure 6-7 Raw data plot of GPR scan through Mark 12 of the test log

This scan has been processed the same way as Log1. Figure 6-8 shows the radar scan after applying High Pass Filter. The defects patterns are seen clear in this scan. The first 10 scans have been removed to make the defect patterns appear better.

Data processed using RADAN 5.0 software (Figure 6-9) has been given along with the high pass filter applied radar scan (Figure 6-8) for comparison. It can be seen from the Figures 6-8 and 6-9 that the results of MATLAB algorithm are in agreement with the results of RADAN software.

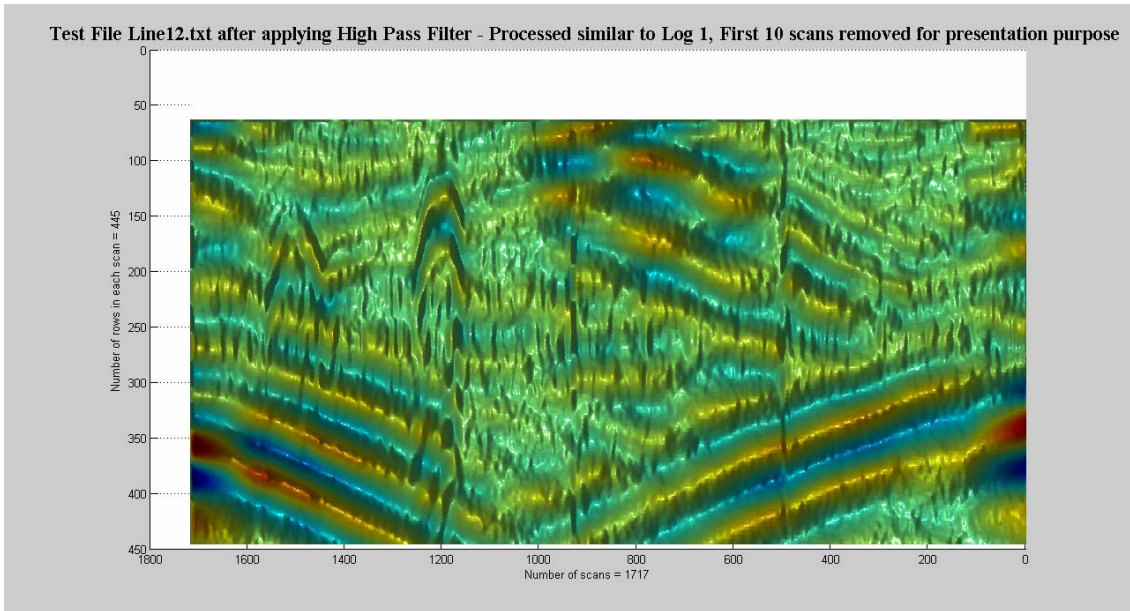


Figure 6-8 Radar scan through Mark 12 after applying High Pass Filter

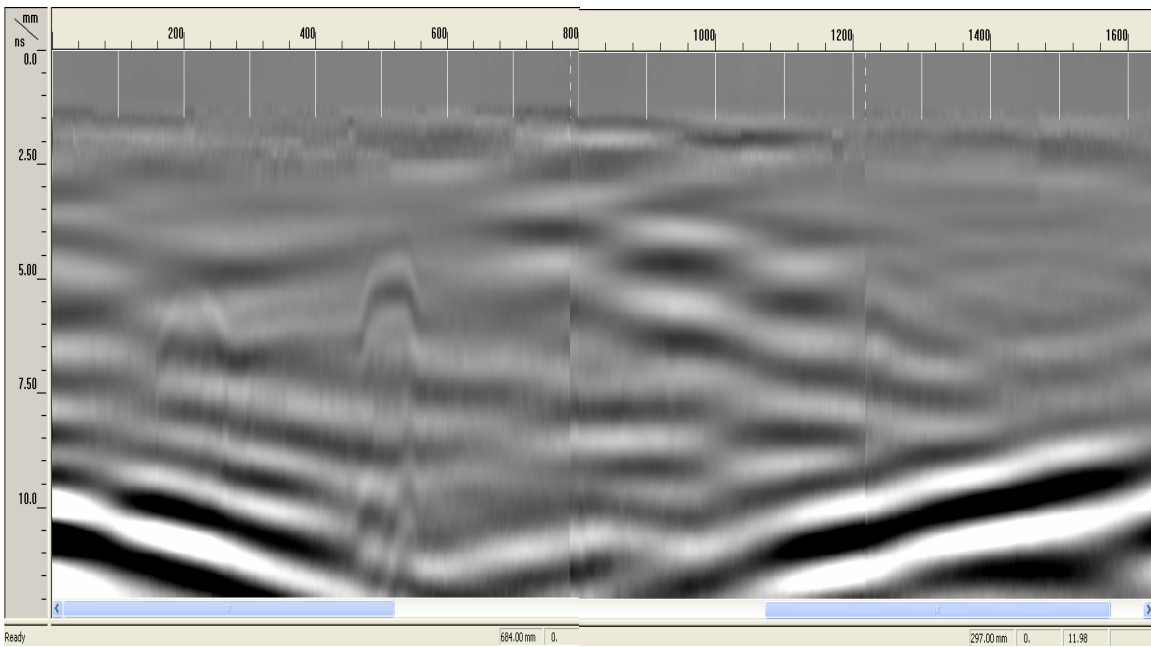


Figure 6-9 RADAN Processed GPR data of Test Log along Mark 12

Figure 6-10 shows the radar scan after applying threshold. Threshold value used is 25. Figure 6-10 contains both defects and noise.

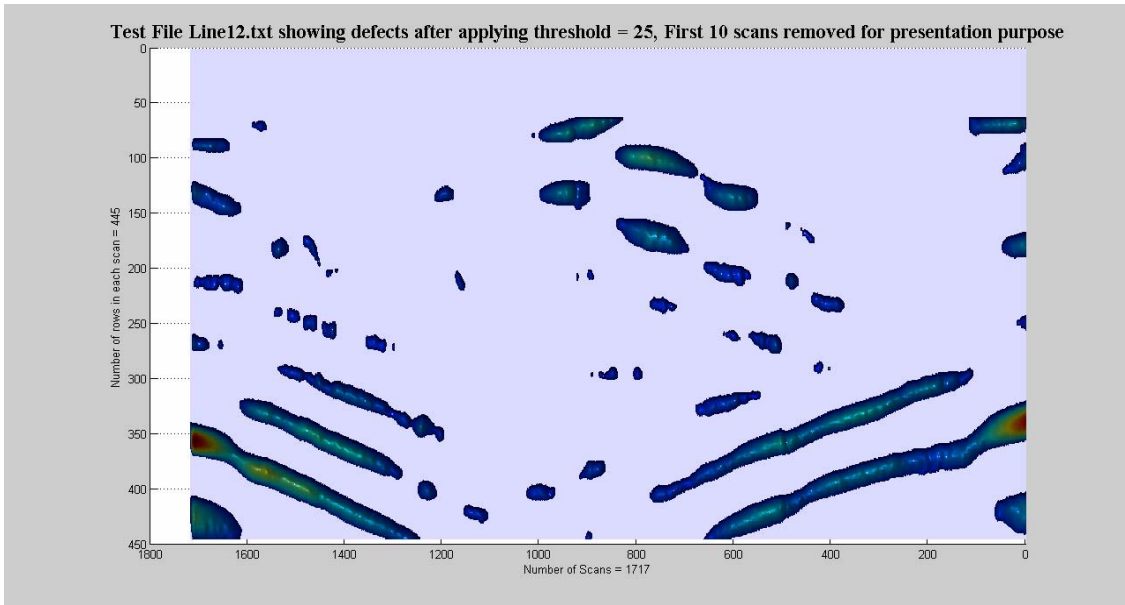


Figure 6-10 Radar scan showing defects for threshold value = 25

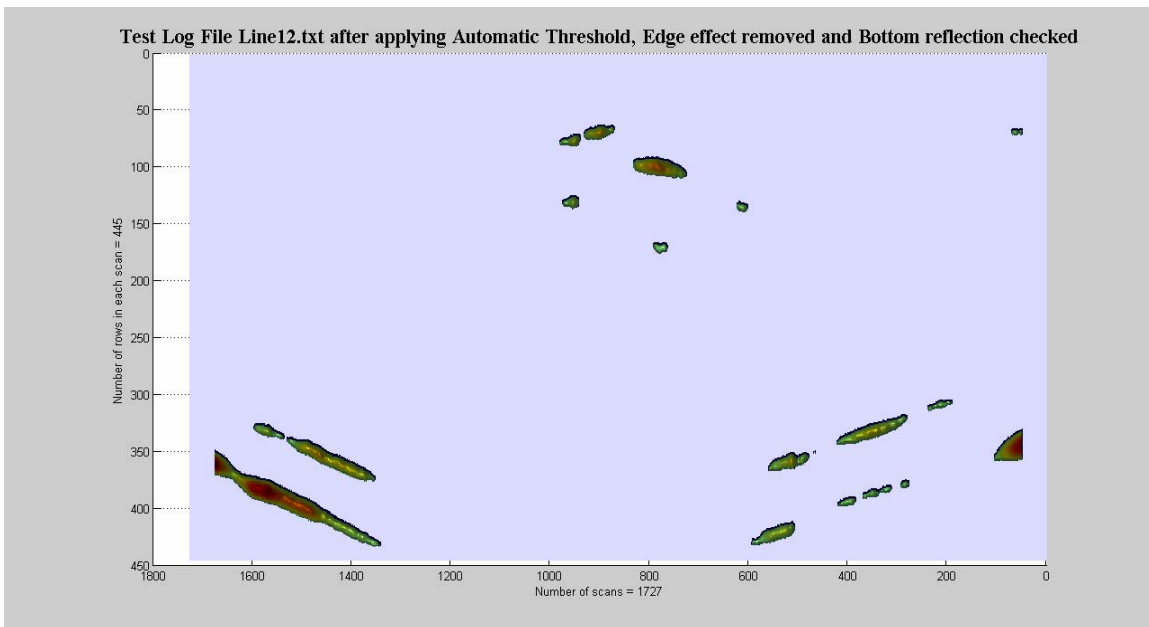


Figure 6-11 Radar scan showing defects for automatic threshold value = 52

Figure 6-11 shows the defects detected in radar scan by applying automatic threshold. Figure 6-11 also involves edge effect removal and check for bottom reflection removal. All the major defects have been identified by automatic threshold application. Automatic threshold value calculated is 52 where as the manual threshold value is 25.



### 6.3 Log11 along Line 5

Figure 6-12 shows raw data plot of Log11 along Line 5. Some patterns can be seen in this scan. Output of MATLAB algorithm for this log has been compared with actual defect locations in the log instead of RADAN output.

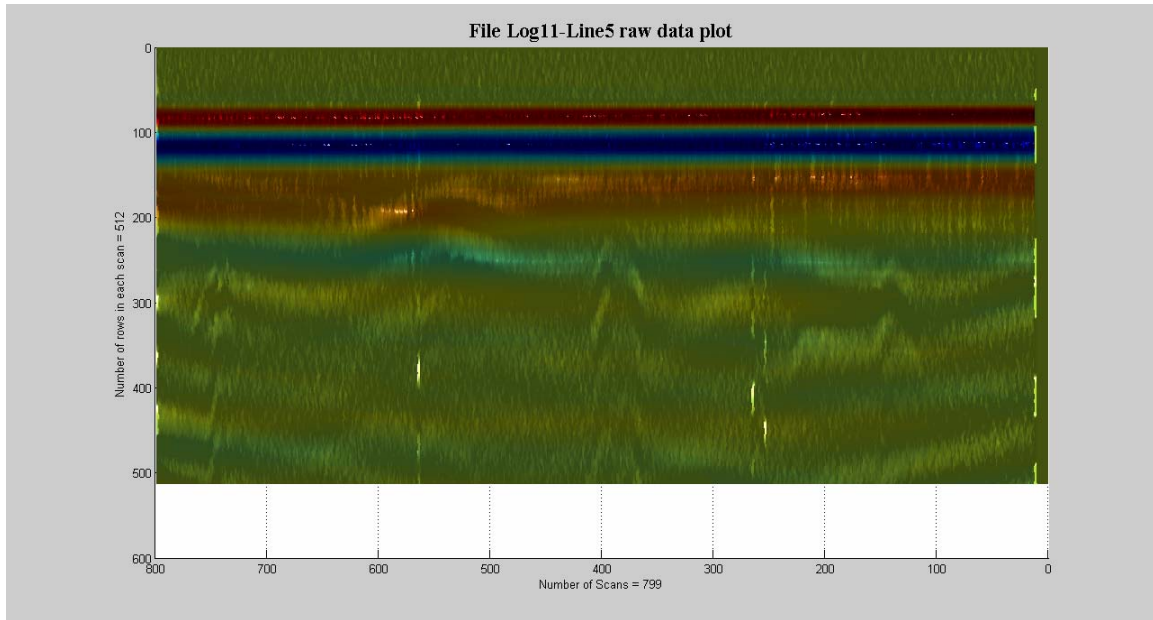


Figure 6-12: Raw data plot of Log11 along Line5

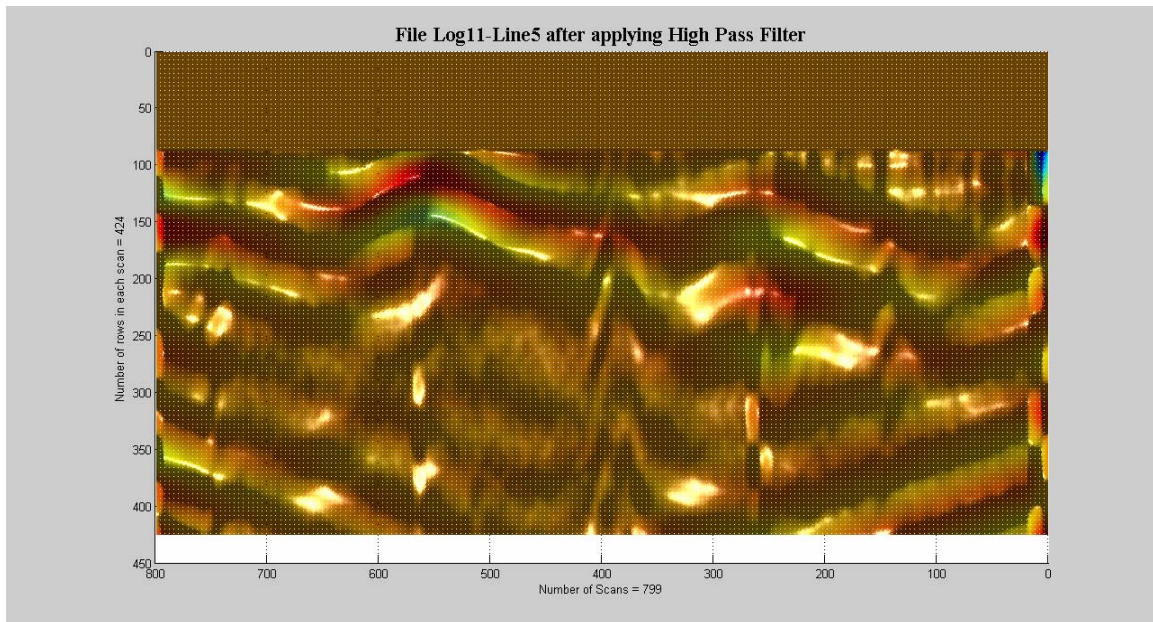


Figure 6-13: Radar scan of Log11 along Line5 after applying HPF

Figure 6-13 shows the processed data of Log11 along line 5. This log has been processed similar to Log1 in the thesis. Figure 6-13 is not very clear because of tall areas in the scan. And also, it has some edge effect. Figure 6-14 shows the scan after removing the tall areas and edge effect. The defect patterns are seen clear in this scan. The reflection from Metal 1 is clear where as the reflection from Metal 2 is not. The 2<sup>nd</sup> reflection from Metal 2 is stronger than 1<sup>st</sup> reflection.

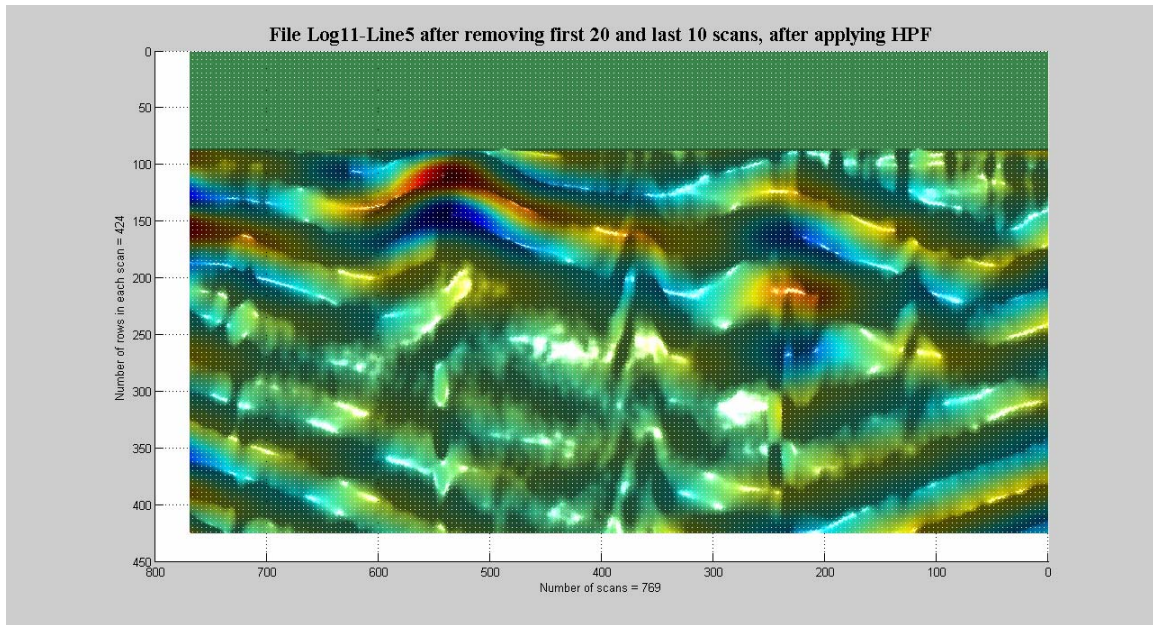


Figure 6-14: Log11 along Line5 after removing first 20 and last 10 scans

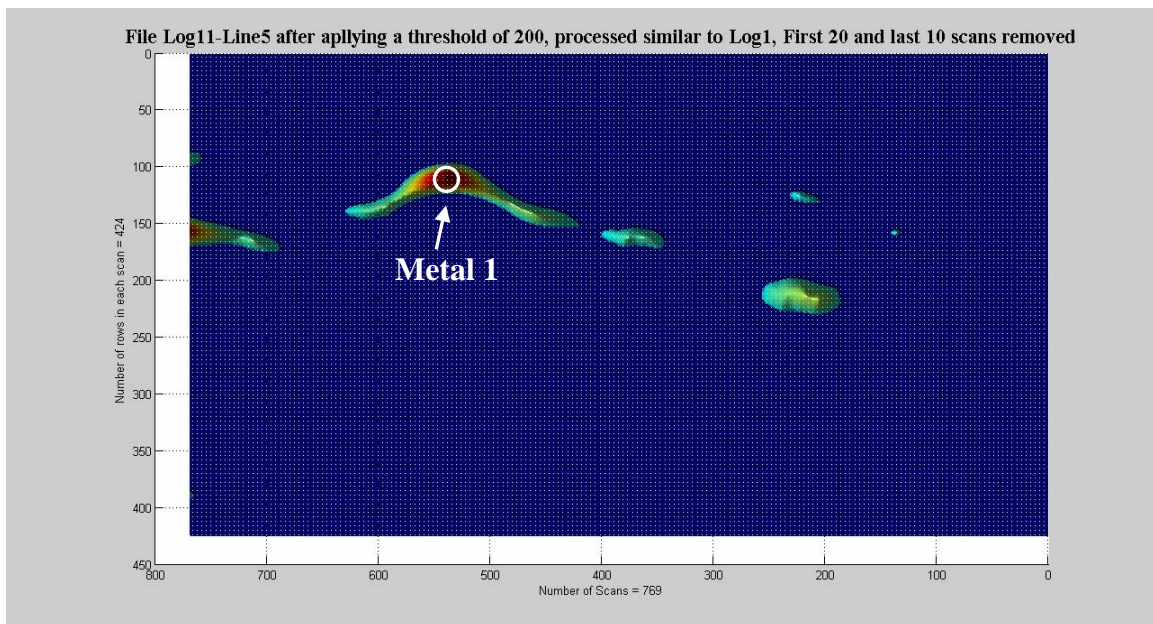


Figure 6-15: Radar scan of Log11 along Line5 after applying a threshold of 200

Figure 6-15 shows the radar scan after applying threshold. Threshold value used is 200. Figure 6-15 contains both defects and noise.

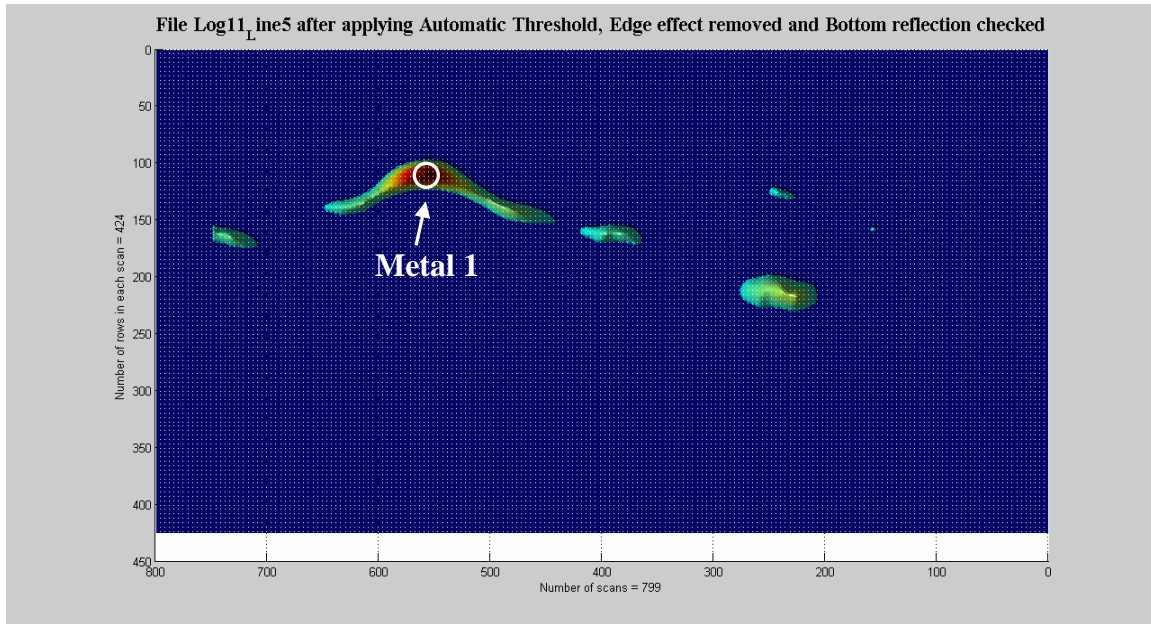


Figure 6-16: Radar scan showing defects for automatic threshold value = 202

Figure 6-16 shows the defects detected in radar scan by applying automatic threshold. Figure 6-16 also involves edge effect removal and check for bottom reflection removal. All the defects have been identified by automatic threshold application. Automatic threshold value calculated is 202 where as the manual threshold value is 200.

### Calculation of depth and distance of Metal 1

Diameter of the log = 13.85 inches, Scans per meter = 500, Defect row = 110,  
Defect Column = 230, Number of rows representing log diameter in the scan = 400

$$\text{Depth of defect} = \frac{\text{Defect Row}}{\text{No. of rows representing log diameter in the scan}} \times \text{Dia of log}$$

$$\text{Depth of Metal 1} = \frac{110}{400} \times (13.85 \text{ inches} \times 0.0254 \text{ m/inch}) = 0.0967 \text{ m}$$

$$\text{Distance of Metal from the left edge of the scan} = \frac{\text{Defect Column}}{\text{No. of scans/m}} = \frac{230}{500} = 0.46 \text{ m}$$

The actual depth of the metal in the log was 0.115 m and the actual distance of the metal from the left edge of the scan was 0.46 m which are in line with predicted values.

## Chapter 7

### CONCLUSIONS AND FUTURE WORK

#### 7.1 Conclusions

A MATLAB algorithm was developed to process the GPR data similar to RADAN software. The output from MATLAB algorithm is in agreement with the output of RADAN algorithm. The MATLAB algorithm was developed to do the following tasks to process GPR Data.

1. Linear Gain
2. Zero Correction
3. Clipping the signal
4. Signal Filtering
  - Low Pass Filter
  - High Pass Filter
5. Threshold

Two options have been given to the user to process the GPR data. The first option is to view only internal defects which eliminates more noise and the second option is to view both internal and surface defects after processing. Noise reduction has been further achieved by developing code to remove bottom reflection and edge effect. The maximum noise reduction was seen in Log1. Calculation of automatic threshold has overcome the limitation of applying threshold manually by trial and error method.

The resolution of defect prediction from MATLAB and RADAN software are as shown in Table 7-1.

Table 7-1 Resolution of Defect Prediction

Software	Depth	Length
RADAN	+/- 0.04 m	-0.05, +0.02
MATLAB	-0.04, +0.03	-0.041, +0.02

From Table 7-1, it can be seen that MATLAB algorithm is comparable to the RADAN software in terms of prediction accuracy. The depth resolution value is equal to

approximately half the wavelength of the radar signal, which comes to 0.048 m for 900 MHz radar signal in a log with dielectric constant of 12 (Agrawal 2005). Defect smaller than 0.048 m may not be detected by radar.

Defects location has been obtained in 2-Dimensional coordinates in meters. Defect location in X-coordinate (distance from the edge of the scan or log) is obtained using scans per meter information. Defect location in Y-coordinate can be obtained by two methods. One method is from the value of dielectric constant if known and the other is by knowing the position of bottom of the log in the scan.

Validation of defects detected by MATLAB algorithm has been done with six logs scanned in the beginning of this research. Further validation of the MATLAB algorithm was done by testing it on two additional logs. All the results obtained from MATLAB algorithm were in agreement with RADAN output and also with the actual location of the defects.

Table 7-2 shows the summary of the defects found from different logs. These defects were detected by MATLAB algorithm in line with RADAN software.

Table 7-2 Summary of defects found

<b>Log</b>	<b>Types of defects found</b>
1	Knots, Rot Initiation
2	Knots, Metal
3	Rot, Metals
4	Metals
5	Knots
6	Rot

A trial code has been developed to remove unwanted echoes (multiple reflections) in the radar scan. Log4 had echoes from metal defects. The trial code eliminated these echoes by separating them and deleting them leaving the first reflection in the scan.

## 7.2 Future Work

The following are the areas that need to be addressed in order to improve MATLAB algorithm and automatic defect detection using GPR.

1. MATLAB/RADAN processed radar data contains noise along with the defects. This noise comes from different reasons like reflection from air/wood interface, wood/air interface, high moisture content, echoes etc., Reflection from wood/air interface can be eliminated if that reflection is strong and consistent in its position.
2. Developed MATLAB code can handle only 2-Dimensional data. To saw a log eliminating the defect, volume of the defect in 3-D coordinates has to be known. To know the volume of the defect, 3-dimensional data collection (Many scans and 3-D model building) and processing is required.
3. Strong metal defect obscures adjacent wooden defect in the same log. A metal nail of diameter 0.004 m and length 0.02 m was not detected by GPR in Log5. So, GPR is not 100% reliable. It can miss small defects due to resolution and other limitations.
4. GPR scanning must be done for wooden log before it is cut by head saw. Once the log is made into cant, the log loses the freedom of getting cut optimally. Wooden cant must be cut in one of the two orientations (either vertically or horizontally). Scanning a wooden log touching the surface before it is made into a cant is not easy due to unevenness of log surface.
5. Bottom 1/4<sup>th</sup> of the log cannot be scanned in one go. So, the log must be rotated to scan the bottom of the log. This takes extra time. Also, once the log is rotated, the log orientation changes and reference point on the log for sawing will be lost in mass production.

6. Data entry takes around 10 seconds. GPR scanning takes around 30 seconds (for scanning a log of length 3 m with 500 scans per meter, it takes 15 seconds at the rate of 100 scans/sec and for scanning each log twice it takes 30 seconds). It takes around 10 seconds to rotate the log. Data processing in MATLAB takes around 35 seconds for one scan. To process two scans it will be 70 seconds. Total time comes to two minutes for each log. Two minutes time on a particular task on each log is high in a factory environment. The typical tact time in a factory setting is around one and half minute. Thus, strategies to reduce this time need to be explored.

## REFERENCES

Agrawal, S., (2005) *Nondestructive Evaluation of Wooden Logs Using Ground Penetrating Radar*, M.S. Thesis, Department of Civil and Environmental Engineering, West Virginia University, Morgantown, WV.

Canpolar, Inc. (1987) *“Preliminary Assessment of Impulse Radar to Detect Decay in Hardwood”* Joint Publication of Canadian Forestry Service and the Alberta Forest Service pursuant to the Canada-Alberta Forest Resource Development Agreement, Edmonton, Alberta, Canada.

Craig V. M. Barton, Kelvin D. Montagu (2004) *“Detection of tree roots and determination of root diameters by ground penetrating radar under optimal conditions”* *Tree Physiology* 24, 1323–1331, Heron Publishing—Victoria, Canada

Detection Sciences, Inc. (1994). *“Inspection of Wood with Impulse Radar”* Research Joint Venture Agreement FP-94-2326, USDA Forest Products Laboratory, Madison, WI

Forest and Wildlife Research Center (2005)  
<http://www.cfr.msstate.edu/fwrc/forestp.htm>

Green, D. V., Winandy, J.E., Kretschmann, D. E. (1999). *“Mechanical Properties of Wood”* Chapter 4 of Wood Handbook, U.S. Department of Agriculture, Forest Products Laboratory, Madison, WI. (Paper also available at [http://www.woodweb.com/knowledge\\_base/Wood\\_Handbook.html](http://www.woodweb.com/knowledge_base/Wood_Handbook.html)).

Gupta, N. K., Schmoldt, D. L., Isaacson, B. (1998). *“Tangential Scanning of Hardwood Logs: Developing an Industrial Computer Tomography Scanner”* Proceedings of the Eleventh International Symposium on Nondestructive Testing of Wood, Forest Products Society, Madison, WI, Sept. 9-11, 131-139.

(Paper also available at <http://www.srs4702.forprod.vt.edu/pubsub/pdf/9922.pdf>)



Halabe, U. B., Petro, S. H., Ganga Rao H. V. S., (1995) *Nondestructive Evaluation Methods for Highway Bridges Superstructures*, CFC 95-215, Constructed Facilities Center, Department of Civil and Environmental Engineering, West Virginia University, Morgantown, WV, 1995.

Halabe, U. B., Ganga Rao H. V. S., Petro, S. H., and Hota, V. R., (1996) "Assessment of Defects and Mechanical Properties of Wood Members Using Ultrasonic Frequency Analysis," *Materials Evaluation*, 54(2), 1996, 314-322.

Halabe, U. B., Chen, H. L., Bhandarkar, V., and Sami, Z., (1997) "Detection of Sub-Surface Anomalies in Concrete Bridge Decks Using Ground Penetrating Radar," *American Concrete Institute (ACI) Materials Journal*, 94(5), 1997, 396-408.

Muller, W. (2002) "Trial of Ground Penetrating Radar to Locate Defects in Timber Bridge Girders" Queensland Department of Main Roads, Brisbane, Australia.  
(Paper also available at [http://www.ipwea.org.au/papers/download/Muller\\_W.pdf](http://www.ipwea.org.au/papers/download/Muller_W.pdf))

NextTag.com (2006) <http://www.nexttag.com/MATLAB/search-html>  
<http://www.nexttag.com/Desktop/search-html>

Quan Zhu, Leslie M. Collins (2005) "Application of Feature Extraction Methods for Landmine Detection Using the Wichmann/Niitek Ground-Penetrating Radar"  
*IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 1, January 2005

Ross, R. J., Brashaw, B. K., and Pellerin, R. F., (1998) "Nondestructive Evaluation of Wood," *Forest Products Journal*, 48(1), 1998, 14-19.

Schad, K. C., Schmoltdt, D. L., and Ross, R. J., (1996) "Nondestructive Methods for Detecting Defects in Softwood Logs" Research Paper FPL-RP-546, 1996, Forest Products Laboratory, U. S. Department of Agriculture, Madison, WI.

Vinay, K. Ingle and John, G. Proakis (2004) "Digital Signal Processing using MATLAB"  
Text Book, Northeastern University, Brooks/Cole Publishing Company.

## APPENDIX

**Table A-1 Time and Frequency Terminology**

Term	Symbol	Units	Notes
Sample period	$T_s$ $T_{si}$ $T_{so}$	Seconds	The time interval between consecutive samples in a sequence, as the input to a block ( $T_{si}$ ) or the output from a block ( $T_{so}$ ).
Frame period	$T_f$ $T_{fi}$ $T_{fo}$	Seconds	The time interval between consecutive frames in a sequence, as the input to a block ( $T_{fi}$ ) or the output from a block ( $T_{fo}$ ).
Signal period	$T$	Seconds	The time elapsed during a single repetition of a periodic signal.
Sample rate, or Sample frequency	$F_s$	Hz (samples per second)	The number of samples per unit time, $F_s = 1/T_s$ .
Frequency	$f$	Hz (cycles per second)	The number of repetitions per unit time of a periodic signal or signal component, $f = 1/T$ .
Nyquist rate		Hz (cycles per second)	The minimum sample rate that avoids aliasing, usually twice the highest frequency in the signal being sampled.
Nyquist frequency	$f_{nyq}$	Hz (cycles per second)	Half the Nyquist rate.
Normalized frequency	$f_n$	Two cycles per sample	Frequency (linear) of a periodic signal normalized to half the sample rate, $f_n = \omega/\pi = 2f/F_s$ .
Angular frequency	$\Omega$	Radians per second	Frequency of a periodic signal in angular units, $\Omega = 2\pi f$ .
Digital (normalized angular) frequency	$\omega$	Radians per sample	Frequency (angular) of a periodic signal normalized to the sample rate, $\omega = \Omega/F_s = \pi f_n$ .

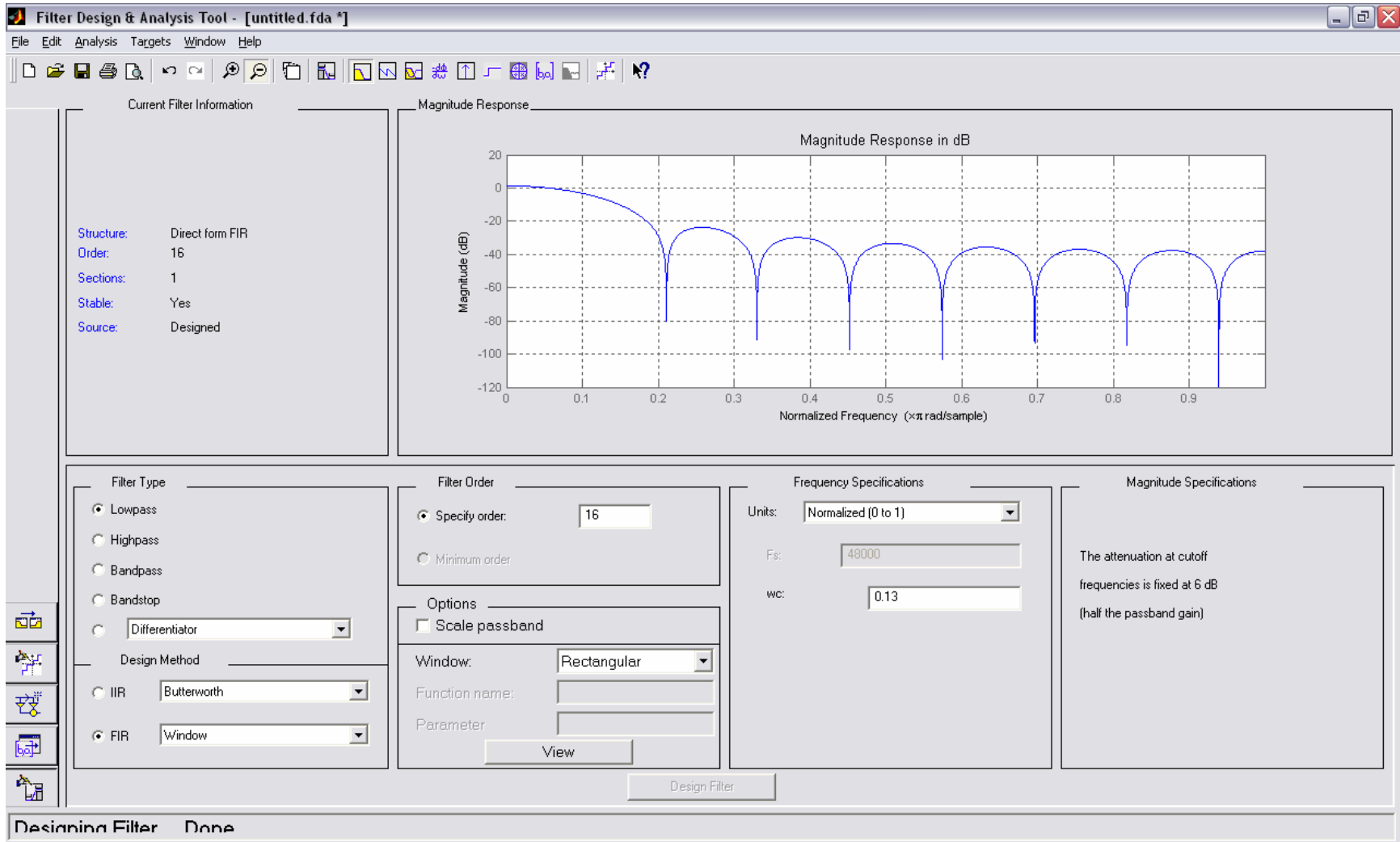


Figure A-1 Screen shot of Filter Design and Analysis Tool (FDATool) showing Low Pass Filter Design Specifications

**Defect\_Identifier**

**Algorithm to Detect Defects in Wooden Logs**

Enter name of Data file (\*.txt) :

Enter round trip travel time through wood (in nano secs) :

Enter DI-electric constant of wood :   
(if known)

Enter the number of scans per meter :

Enter the average diameter of the log (in meters) :

Enter the amplitude value for which you would like to see defects :   
(if known)

Remove Reflections

Choose Option  Internal Defects

Internal + Surface Defects

Figure A-2 Screen shot of Algorithm to Detect Defects in Wooden Logs

## MATLAB CODE

```
%%%%%%%%%%
%%GUI Starts
%%%%%%%%%%

function varargout = Defect_Identifier(varargin)
% DEFECT_IDENTIFIER M-file for Defect_Identifier.fig
% DEFECT_IDENTIFIER, by itself, creates a new DEFECT_IDENTIFIER or raises
the existing
% singleton*.
%
% H = DEFECT_IDENTIFIER returns the handle to a new DEFECT_IDENTIFIER
or the handle to
% the existing singleton*.
%
% DEFECT_IDENTIFIER('CALLBACK',hObject,eventData,handles,...) calls the
local
% function named CALLBACK in DEFECT_IDENTIFIER.M with the given input
arguments.
%
% DEFECT_IDENTIFIER('Property','Value',...) creates a new
DEFECT_IDENTIFIER or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before Defect_Identifier_OpeningFunction gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to Defect_Identifier_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Defect_Identifier

% Last Modified by GUIDE v2.5 14-Dec-2006 03:03:52

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Defect_Identifier_OpeningFcn, ...
    'gui_OutputFcn', @Defect_Identifier_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin & isstr(varargin{1})
```

```

    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Defect_Identifier is made visible.
function Defect_Identifier_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Defect_Identifier (see VARARGIN)

% Choose default_val command line output for Defect_Identifier
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Defect_Identifier wait for user response (see UIRESUME)
% uiwait(handles.figure1);

set(handles.filename,'string','.txt');
set(handles.internal,'value',1);

% --- Outputs from this function are returned to the command line.
function varargout = Defect_Identifier_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default_val command line output from handles structure
varargout{1} = handles.output;

% --- Executes during object creation, after setting all properties.
function filename_CreateFcn(hObject, eventdata, handles)
% hObject    handle to filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function filename_Callback(hObject, eventdata, handles)
% hObject    handle to filename (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of filename as text
%    str2double(get(hObject,'String')) returns contents of filename as a double

% --- Executes during object creation, after setting all properties.
function time_CreateFcn(hObject, eventdata, handles)
% hObject    handle to time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%    See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function time_Callback(hObject, eventdata, handles)
% hObject    handle to time (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of time as text
%    str2double(get(hObject,'String')) returns contents of time as a double

% --- Executes during object creation, after setting all properties.
function di_electric_CreateFcn(hObject, eventdata, handles)

```

```

% hObject handle to di_electric (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function di_electric_Callback(hObject, eventdata, handles)
% hObject handle to di_electric (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of di_electric as text
% str2double(get(hObject,'String')) returns contents of di_electric as a double

% --- Executes during object creation, after setting all properties.
function scans_mtr_CreateFcn(hObject, eventdata, handles)
% hObject handle to scans_mtr (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function scans_mtr_Callback(hObject, eventdata, handles)
% hObject handle to scans_mtr (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of scans_mtr as text
% str2double(get(hObject,'String')) returns contents of scans_mtr as a double

```



```

% --- Executes during object creation, after setting all properties.
function dia_CreateFcn(hObject, eventdata, handles)
% hObject    handle to dia (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function dia_Callback(hObject, eventdata, handles)
% hObject    handle to dia (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of dia as text
%       str2double(get(hObject,'String')) returns contents of dia as a double

```

```

% --- Executes during object creation, after setting all properties.
function amplitude_CreateFcn(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function amplitude_Callback(hObject, eventdata, handles)
% hObject    handle to amplitude (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles  structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of amplitude as text
%        str2double(get(hObject,'String')) returns contents of amplitude as a double

function mutual_exclude(off)
set(off,'Value',0)

% --- Executes on button press in internal.
function internal_Callback(hObject, eventdata, handles)
% hObject  handle to internal (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of internal
off = [handles.int_surf];
mutual_exclude(off)

% --- Executes on button press in int_surf.
function int_surf_Callback(hObject, eventdata, handles)
% hObject  handle to int_surf (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of int_surf
off = [handles.internal];
mutual_exclude(off)

% --- Executes on button press in field_clear.
function field_clear_Callback(hObject, eventdata, handles)
% hObject  handle to field_clear (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)
set(handles.filename,'string','.txt');
set(handles.time,'string','');
set(handles.di_electric,'string','');
set(handles.scans_mtr,'string','');
set(handles.dia,'string','');
set(handles.amplitude,'string','');

% --- Executes on button press in default_val.
function default_val_Callback(hObject, eventdata, handles)
% hObject  handle to default_val (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)
set(handles.filename,'string','.txt');

```

```

set(handles.time,'string','');
set(handles.di_electric,'string','');
set(handles.scans_mtr,'string','472');
set(handles.dia,'string','0.8');
set(handles.amplitude,'string','700');

% --- Executes during object creation, after setting all properties.
function comments1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to comments1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% --- Executes during object creation, after setting all properties.

function comments2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to comments2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes during object deletion, before destroying properties.
function comments1_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to comments1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object deletion, before destroying properties.
function comments2_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to comments2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in reflect.
function reflect_Callback(hObject, eventdata, handles)
% hObject    handle to reflect (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of reflect

% --- Executes on button press in run_prog.
function run_prog_Callback(hObject, eventdata, handles)
% hObject    handle to run_prog (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles  structure with handles and user data (see GUIDATA)

%% Reading input from GUI
option=get(handles.internal,'value');
filename=get(handles.filename,'string');
file_find=exist(filename,'file');
if file_find == 2
    time=str2num(get(handles.time,'string'));
    di_electric=str2num(get(handles.di_electric,'string'));
    scans_mtr=str2num(get(handles.scans_mtr,'string'));
    dia=str2num(get(handles.dia,'string'));
    amplitude=str2num(get(handles.amplitude,'string'));
    reflect=get(handles.reflect,'value');
    set(handles.comments1,'string','Working. Please wait...');
    set(handles.comments2,'string','');
    set(handles.comments1,'BackgroundColor',[0 1 1]);
    set(handles.comments2,'BackgroundColor',[0.925 0.914 0.843]);
    pause(1)
    %Loading the data file
    data=load(filename);
else
    set(handles.comments1,'string','File not found');
    set(handles.comments2,'string','');
    set(handles.comments1,'BackgroundColor',[1 0 0]);
    set(handles.comments2,'BackgroundColor',[0.925 0.914 0.843]);
    error('File not found')
end

%%%%%%%%%%
%%GUI Ends
%%%%%%%%%%

%%%%%%%%%%
%% Detection Program Starts
%%%%%%%%%%

%Recording the size of the data file

[m,n]=size(data);

%% Realigning of Scans

```

```

for j=1:n
    data1(:,n+1-j)=data(:,j);
end

%% Initial Noise Removal

for j=1:n
    data1(1:3,j)=32750;
end

%% Bringing the average of the entire scan to zero

whole_avg=sum(sum(data1(1:m,1:n)))/(m*n);
data2(1:m,1:n)=data1(1:m,1:n)-whole_avg;

%% Copying and adding scans from the edges

a = data2(:,1:138);
dat = [a data2];

%Recording the size of the data file

[m4,n4] = size(dat);

b = dat(:,n4-137:n4);
d = [dat b];

%Recording the size of the data file

[m5,n5] = size(d);

%% Applying Linear Gain

for j=1:n5
    gain=1;
    for i=1:m5
        gain=gain+ 0.00015625;
        e(i,j)=d(i,j)*gain;
    end
end
end

```

```

%% Zero Correction by taking average of all the maximums

pos1 = 0;

for j=1:n5,
    [array_max,pos]=max(e(3:200,j)); %% storing highest value position from each scan
= pos
    pos1 = pos + pos1;
end

avg = fix(pos1/n5);

f = e(avg:m5,:);

%% Zero Correction by realigning all the maximums in a line

% for j=1:n5,
%   [y,I]=max(e(4:m5,j));
%   f(1:m-I+1,j)=e(I:m,j);
% end

%Recording the size of the data file

[m1,n1]=size(f);

if option == 1

%% OPTION - INTERNAL

%% 1st Clipping of the signal

for j=1:n1,
    [array_min,pos3]=min(f(3:200,j)); %% storing lowest value position in pos3
    for i=pos3:m1,
        if f(i,j) >= 0
            g=i;
            break;
        else
            g=0;
        end
    end
end

```

```

        f(1:g,j)=zeros;
    end

%% 2nd Clipping of the signal

    ps = 0;

    for j=1:n1,
        [array_mac,pos2]=max(f(3:200,j)); %% storing highest value position from each
scan = pos2
        ps = pos2 + ps;
    end

    avrg = fix(ps/n1);

    for j=1:n1,
        f(1:avrg,j)=zeros;
    end

%% END OF OPTION - INTERNAL

else

%% OPTION - INTERNAL + SURFACE

    ps = 0;

%Clipping the Signal

    for j=1:n1,
        [array_min,pos2]=min(f(3:200,j)); %% storing lowest value position in pos2
        ps = pos2 + ps;
    end

    avrg = fix(ps/n1);

    for j=1:n1,
        f(1:avrg,j)=zeros;
    end

%% END OF OPTION - INTERNAL + SURFACE

end

% Clipping waveform going out of the log

```

```

ps6 = 0;
ary = 0;

for j=1:n1,
    [array_in,pos6]=min(f(200:m1,j));

    ary = array_in + ary;
    ps6 = pos6 + ps6 +200;
end

aryavg = (ary/n1);
avrg6 = fix(ps6/n1);

%% Checking reflection from the bottom of the log

if aryavg < -1000,
    for j=1:n1,
        f(avrg6:m1,j)=zeros;
    end
else %% ELSE - DISPLAY MESSAGE 'BOTTOM OF THE LOG IS NOT FOUND'
    set(handles.comments1,'string','Reflection from Bottom of the log is not found. ');
    set(handles.comments1,'BackgroundColor',[1 0 0]); % red color
end

%% Applying Lowpass Filter

L = load('Lowpass');
d1 = imfilter(f,L);
d2 = d1(:,10:end-9);

%% Applying High Pass Filter

g = -ones(257,1)/256;
g(129) = 1;
d3 = imfilter(d2,g');
d4 = d3(:,130:end-129);

%% Displaying the processed data

% figure;
% Surf(d4)

```



```

%Recording the size of the data file

[m2,n2] = size(d4);

%% Edge effect Removal

for i=1:m2,
    d4(i,1:50)=zeros;
    d4(i,n2-50:n2)=zeros;
end

%% Displaying the processed data

figure;
surf(d4)

%% Automatic Threshold Value Calculation

ray = 0;

for j=51:n2-50,
    [ray_max,positi]=max(d4(3:m2,j)); %% storing highest value position from each scan =
    posi
    ray = ray_max + ray;
end

rayavg = (ray/n2);

%%Checking Automatic Threshold Value

if rayavg < 100
    % DISPLAY MESSAGE AUTOMATIC THRESHOLD VALUE NOT FOUND and
    IT SHOULD NOT BE USED FOR APPLYING THRESHOLD FURTHER IN THIS
    PROGRAM
    set(handles.comments1,'string','Automatic threshold value not found.');
```

```

    set(handles.comments1,'BackgroundColor',[1 0 0]); % red color
    if isempty(amplitude) == 1
        set(handles.comments2,'string','Either no Defects were found in log or Retry with
known Threshold Value');
        set(handles.comments2,'BackgroundColor',[0 1 1]); % blue color
        error('Program Terminated'); % terminate M-file
    end
end

```

```

        %error('Automatic threshold value not found'); % terminate M-file
    end
end

```

```

%% Applying Initial Threshold if remove reflections is selected

```

```

if reflect == 1
    for j=1:n2
        for i= 1:m2
            if d4(i,j)<=300
                d4(i,j)=0;
            end
        end
    end
end

```

```

%% Displaying the processed data

```

```

% figure;
% Surf(d4)

```

```

% %% Removing Reflections

```

```

for i=1:n2
    reset = 0;
    notzero = 0;
    for j=1:m2
        if reset == 1
            d4(j,i) = 0;
            continue;
        end
        if d4(j,i) >= 1
            notzero = 1;
            continue;
        elseif notzero == 1
            reset = 1;
        end
    end
end
end

```

```

%% Removing Small Chunks

```

```

itt=1;
for i=1:n2
    flag = 0;

```

```

for j=1:m2
    if d4(j,i) > 0
        flag = 1;
        break;
    end
end
if flag == 0
    blank_list(itt) = i;
    itt = itt+1;
else blank_list(itt) = 0;
end
end

for i=1:length(blank_list)-1
    test=d4(:,blank_list(i):blank_list(i+1));
    s=length(find(test));
    if s < 1500
        d4(:,blank_list(i):blank_list(i+1)) = 0;
    end
end

clear blank_list;
clear test;

itt=1;
for i=1:m2
    flag = 0;
    for j=1:n2
        if d4(i,j) > 0
            flag = 1;
            break;
        end
    end
    if flag == 0
        blank_list(itt) = i;
        itt = itt+1;
    else blank_list(itt) = 0;
    end
end

for i=1:length(blank_list)-2
    test=d4(blank_list(i):blank_list(i+1),:);
    s=length(find(test));
    if s < 1500
        d4(blank_list(i):blank_list(i+1),:) = 0;
    end
end

```

```

    end

%% Displaying the processed data

% figure,surf(d4);

% Applying Final Threshold

    for j=1:n2
        for i= 1:m2
            if d4(i,j)<=700
                d4(i,j)=0;
            end
        end
    end

else

    %% Applying Threshold --- ** HERE IF USER INPUT AMPLITUDE VALUE IS
    AVIALABLE, IT HAS TO BE USED otherwise, AUTOMATIC THRESHOLD VALUE
    HAS TO BE USED - 'rayavg'

    threshold=[];
    if isempty(amplitude) == 1
        threshold= rayavg;
    else
        threshold = amplitude;
    end

    for j=1:n2
        for i= 1:m2
            if d4(i,j)<= threshold;%% **
                d4(i,j)=0;
            end
        end
    end
end

figure;
surf(d4)

%% Converting defect locations into X-Y Coordinates in meters

count=1;
d_row=[];

```

```

d_col=[];
for i=1:n2
    for j=1:m2
        if (d4(j,i))>10
            d_row(count)=j;
            d_col(count)=i;
            count=count+1;
        end
    end
end

%% IF DIELECTRIC CONSTANT VALUE IS KNOWN, THE FOLLOWING LINES
HAS TO WORK

d_row_scaled=[];
d_col_scaled=[];
if isempty(di_electric) == 0
    depth=(0.3*time)/(2*sqrt(di_electric));
    d_row_scaled(1:length(d_row))=d_row(1:length(d_row))*depth/512;

else %% WHEN BOTTOM OF THE LOG IS FOUND, THE FOLLOWING LINES
HAS TO WORK

    if aryavg < -1000,
        dep = avrg6 - 1;
        d_row_scaled(1:length(d_row))=d_row(1:length(d_row))*dia/dep;
    end
end

d_col_scaled(1:length(d_col))=d_col(1:length(d_col))/scans_mtr;

coord = [d_col_scaled %% X - Coordinate Matrix
         d_row_scaled]; %% Y - Coordinate Matrix

%% Final Message Display

if isempty(d_row_scaled) == 1
    set(handles.comments1,'string','Defects Detection was not Sucessful');
    set(handles.comments1,'BackgroundColor',[0 1 1]); % blue color
    set(handles.comments2,'BackgroundColor',[0.925 0.914 0.843]); %% grey/brown
    (same as background)
else
    set(handles.comments1,'string','Defects Detection Successful');
    set(handles.comments1,'BackgroundColor',[0 1 0]); %% green color

```

```
    set(handles.comments2,'BackgroundColor',[0.925 0.914 0.843]); %% grey/brown
(same as background)
    save coord_matrix.mat coord
    load('coord_matrix.mat');
end

%%%%%%%%%%%%%%
%% Detection Program Ends
%%%%%%%%%%%%%%
```