

1996

Development of a pulsed eddy current instrument for corrosion detection

Sunil K. Shaligram
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Computer Engineering Commons](#)

Recommended Citation

Shaligram, Sunil K., "Development of a pulsed eddy current instrument for corrosion detection" (1996). *Retrospective Theses and Dissertations*. 257.
<https://lib.dr.iastate.edu/rtd/257>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Development of a pulsed eddy current instrument
for corrosion detection

by

Sunil Kumar Shaligram

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

MASTER OF SCIENCE

Department: Electrical and Computer Engineering
Major: Computer Engineering

Approved:

In Charge of Major Work

For the Major Department

For the Graduate College

Members of the Committee:

Iowa State University
Ames, Iowa
1996

Copyright © Sunil Kumar Shaligram, 1996. All rights reserved.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	vii
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. INSTRUMENT HARDWARE OVERVIEW	5
2.1 Hardware Components	5
2.2 Pulsed Eddy Current Probe	5
2.2.1 Probe construction, coil design and optimization	6
2.3 Scanning System	7
2.3.1 Mechanical construction	7
2.3.2 Stepper motors and their control	9
2.3.3 2-D scanning algorithm	9
2.4 Data Acquisition System and PEC Card	12
2.4.1 Daughter board control	13
CHAPTER 3. SIGNAL AND IMAGE PROCESSING	14
3.1 Data Acquisition and Timing	14
3.2 Signal Storage, Plotting and Extraction	18
3.2.1 Detection of t_{ZC} and PH	19
3.3 Time and Frequency Domain Toggle	20
3.4 Image Formation	21

3.4.1 Pixel to pixel interpolation	23
3.5 Image Conversion and Storage	26
3.6 Amplitude and Time-Gating	27
3.6.1 Amplitude gating	28
3.6.2 Time-gating	29
CHAPTER 4. SOFTWARE ELEMENTS AND INSTRUMENT	
OPERATION	31
4.1 Software Organization	32
4.2 Software Implementation Details	34
4.2.1 Motor control software	34
4.2.2 Data storage software	34
4.2.3 Data retrieval and interpretation software	35
4.2.4 Image storage and redisplay software	35
4.2.5 Calibration software	36
4.2.6 Signal display and user interaction software	36
4.3 Instrument Operation	36
CHAPTER 5. CALIBRATION AND TESTING	39
5.1 Calibration	39
5.2 Testing and Results	42
5.2.1 Laboratory testing	42
5.2.2 Field testing	45
CHAPTER 6. FUTURE WORK AND CONCLUSION	48
6.1 Future Work	48
6.1.1 Scanner and probe designs	48

6.1.2 Crack detection	49
6.1.3 Image processing	50
6.1.4 Neural networks for the inverse problem	51
6.2 Conclusions	51
BIBLIOGRAPHY	53
APPENDIX. INSTRUMENT SPECIFICATIONS	55

LIST OF TABLES

Table 2.1:	Coil parameters	8
Table 3.1:	File format of a .img file	21
Table 3.2:	Color index lookup table	23
Table 5.1:	File format of a calibration(.cal) file	41

LIST OF FIGURES

Figure 2.1:	Instrument overview	6
Figure 2.2:	PEC probe	7
Figure 2.3:	Scanner frame	9
Figure 2.4:	Serpentine scan	10
Figure 3.1:	Timing diagram	15
Figure 3.2:	Difference Signal	16
Figure 3.3:	Plot of peak height vs. zero crossing time	30
Figure 4.1:	Software organization on the PEC instrument	32
Figure 4.2:	Screen dump of the PEC instrument panel	38
Figure 5.1:	Calibration curves of the PEC instrument	40
Figure 5.2:	PEC scan of the Boeing Panel VI	43
Figure 5.3:	Ultrasonic C-scan of the Boeing Panel VI	43
Figure 5.4:	Images before and after time gating of the Boeing Panel VI specimen	46
Figure 5.5:	Crack detection using the PEC instrument	47
Figure 5.6:	Scan of left wing, lower spar cap of a Aero Commander aircraft	47

ACKNOWLEDGMENTS

I would like to thank my research supervisor Mr. John C. Moulder for his excellent guidance and constant support throughout the period of this project. This project would perhaps never have successfully culminated without his supervision. I would also like to thank my major professor Dr. Charles T. Wright, and members of my committee Dr. Satish Udpa, Dr. John Basart and Dr. Prabhu for their guidance.

My parents and family are an infinite source of encouragement and inspiration in all of my endeavors, this being no exception. I am forever indebted to them. I would also like to thank the other members of the Pulsed Eddy current group, for being so friendly and cooperative. I would like to acknowledge the cooperation extended by my roommates during my graduate studies.

Finally, I would like to thank the Federal Aviation Administration for their financial support under Grant No. 93-G-018. This section is incomplete without acknowledging the help extended by the staff and faculty at the Center for Nondestructive Evaluation at Iowa State University.

CHAPTER 1. INTRODUCTION

The pulsed eddy current technique has been under development at Iowa State University since 1993 under the supervision of Mr. John Moulder [1]. It has shown much promise for detecting corrosion and cracks in multi-layered aircraft structures, and has elicited strong interest from the aviation community. The Pulsed Eddy Current Group at the Center for Nondestructive Evaluation, ISU, has built an instrument which utilizes the theory behind pulsed eddy currents and transforms it into a practical field instrument.

Initial development of the theory of pulsed eddy currents was carried out by J. H. Rose and J. C. Moulder [2]. Other investigators have also studied pulsed eddy current techniques. Beissner and Fisher [3] [4] studied the application of pulsed eddy currents to the detection and characterization of small cracks in titanium based jet-engine alloys. Recently, Bowler and Harrison [5] have reported on the application of pulsed eddy-currents for detecting corrosion. This was based on an instrument developed by Harrison [6] for detecting cracks (the Staveley Eddyscan 30).

In its basic form, our instrument excites a coil by applying a voltage step for a finite duration of time. The coil is at all times in contact with the surface (neglecting the layer of epoxy over the coil used to protect it from wear and tear). Whenever the probe goes over an area of hidden corrosion, which translates to loss of metal, there

is a change in the eddy currents produced. This change, in turn produces an end change in the impedance of the coil. It is this change that we measure by monitoring the current in the coil.

One great advantage of the pulsed operation is the wide bandwidth. A pulse theoretically consists of an infinite number of frequencies. However, there are practical limitations in the number of frequencies in the signal obtained in the instrument. A finite sampling frequency (1 MHz) immediately cuts down on the available bandwidth. Noise needs to be reduced in the signal by some form of low pass filtering. These effects combine to give us a signal with a frequency range extending up to a few hundred kHz. This resulting bandwidth is advantageous since the time taken for a single measurement with this frequency range is of the order of a few hundred milliseconds. In contrast, a conventional impedance measuring instrument such a measurement can take a lot longer, sometimes on the order of minutes.

In its present state of development, the instrument is able to detect corrosion hidden beneath aluminium sheets up to 4 mm thick. Crack detection is another possibility for this new instrument. The probe scanning platform can be attached to the surface to be inspected, via suction cup attachments for mounting onto an aircraft fuselage or wing. Once mounted on the surface, the user can selectively scan areas of interest. Scan areas are typically rectangular. The scan proceeds in a serpentine fashion from its homing position. It takes approximately fifteen minutes to complete scanning an area of 12 x 9 inches at a resolution of 3 mm. The time duration of a scan depends on the resolution desired by the user. Once a scan is complete, an 11-level pseudo-color image of the hidden corrosion is displayed on the screen. The number of colors can be increased further, but too many colors may confuse the user. We have

found by experience that 11 colors represents a fair compromise. The operation of the instrument in the imaging mode, facilitates assessment of the extent and severity of hidden corrosion. It also helps us compare images to area maps obtained with other NDE modalities such as thermal wave imaging or ultrasonic C-scans. The software that controls the instrument is written in *VisualC++TM*, and is accessed through a *WindowsTM* interface.

The pulsed eddy current (PEC) instrument is small and rugged. A 16-bit, 1MHz Analogic Data Acquisition board acquires data from the probe. We have built a card of our own to amplify and process the signal obtained from the coil of the probe, before being fed into the data acquisition board. An AT6400 motor controller card from Compumotor controls the scanning system. The whole system is small enough to comfortably fit on a 4 ft x 2 ft table. The system can easily be taken apart and reassembled within a few minutes and the mechanical design is rugged enough to withstand the wear and tear of transportation and handling. We have inspected the lower main spar cap of an Aero Commander Airplane, for corrosion.

The research and development work on the pulsed eddy current technique has been ongoing since 1993 under the direction of Mr. John C. Moulder at the Center for NDE, Iowa State University. Initial theoretical work was carried out by Dr. J. H. Rose, Dr. E. Uzal and Moulder. The methods currently in use are an outgrowth of frequency domain corrosion measurements. Mr. Mark Kubovich built the first prototype of the PEC instrument, which was a hand-held version. He also wrote all the software for this first prototype. The present instrument is based on this prototype. Mr. William W. Ward built the new version of the electronic circuit card. Mr. Jay A. Bieber worked on the probe fixture and has developed the measurement

techniques for the instrument. He has also been responsible for testing and evaluating the performance of the instrument during its development. Mr. Cheng-Chi Tai has worked on developing the PEC theory and using the instrument to measure the thickness of layers. The equations relating to the frequency domain methods discussed in Chapter 3 were formulated by Mr. Tai. The contributions of the author of this document have been in developing the new software necessary to make the instrument capable of scanning and imaging areas of corrosion. The author has also developed the signal processing routines. The algorithms and software discussed in chapters 3, 4, 5 and the scanning algorithm discussed in Chapter 2 were developed by the author. Chapter 2 and the data acquisition software of Chapter 3 were developed by one or more of the persons mentioned above and are included in this thesis to give the reader a basic understanding of the system. Most of the experimental data presented in Chapter 5 were obtained by Mr. Bieber.

The thesis is organized into six chapters. Hardware components of the instrument including the PEC card, A/D card, the design and construction of the PEC probe, the scanner and its accessories are discussed in Chapter 2. The data acquisition, signal and image display, storage and manipulation form the contents of Chapter 3. Signal processing aspects are also discussed in this chapter. The software control and development techniques are the subjects of Chapter 4. This chapter also provides a description of the operation procedure. Chapter 5 concerns itself with the testing and calibration of the pulsed eddy current instrument. Laboratory and field testing results are presented. Finally, Chapter 6 deals with future work (which will mainly be in crack detection), and the conclusion.

CHAPTER 2. INSTRUMENT HARDWARE OVERVIEW

2.1 Hardware Components

The different hardware items that constitute the PEC instrument are as follows:

- PEC probe
- Scanning System
- Data Acquisition Board and PEC Card

Before describing each of the above the reader is referred to Figure 2.1 which describes the interaction between the various components described above.

2.2 Pulsed Eddy Current Probe

The probe is undoubtedly one of the most important components in the whole system. The process of corrosion detection is initiated here. Eddy current probes [7] can be operated in various modes such as absolute, reflection or the differential mode. The probe has been configured to operate in the absolute mode for all of our experiments with the scanner, although the instrument is capable of operating in the reflection mode. An important requirement is that the probe should be in contact with the scanning surface at all times.

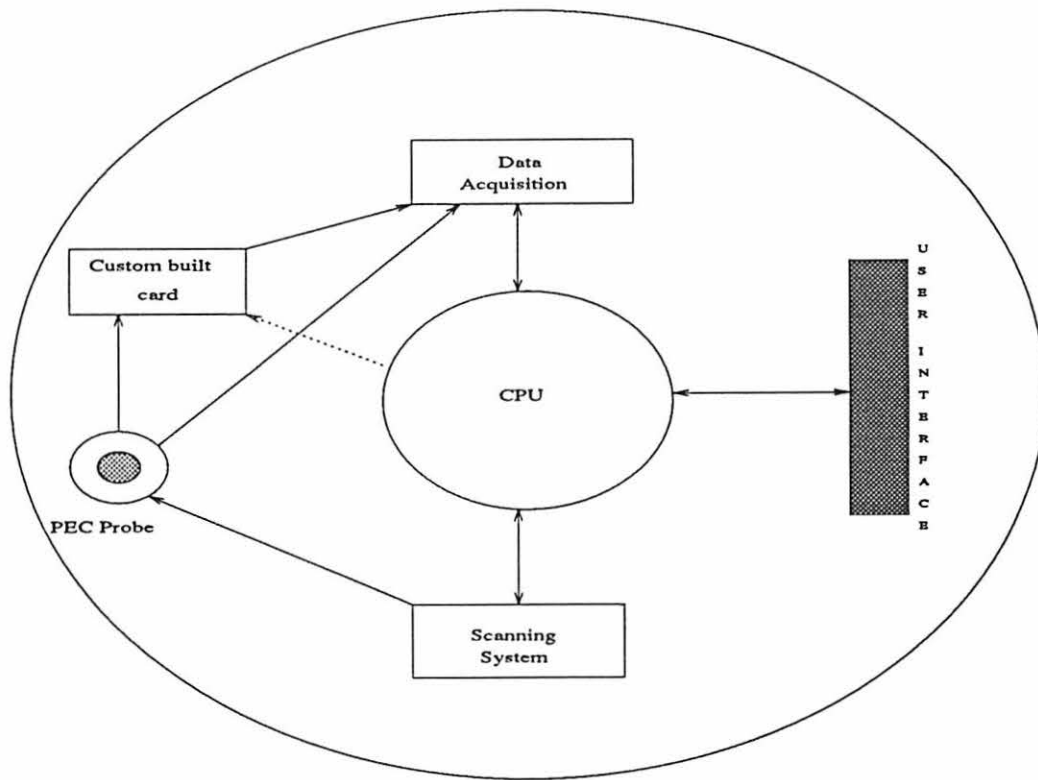


Figure 2.1: Instrument overview

2.2.1 Probe construction, coil design and optimization

The eddy current probe on the PEC instrument is unique in its construction. The probe can go over rivets. The probe can also move easily over uneven surfaces. The actual coil is mounted on a plastic spindle in an aluminum casing. The upper part of the casing is connected to another outer enclosure by a spring. Thus, the coil can physically move up and down when going over uneven surfaces. This whole arrangement has a gimbal which gives it the flexibility of following complex curvatures of the surface being scanned. This complete setup is mounted onto the probe holder of the scanner by a rack and pinion arrangement. The leads of the coil are soldered to a connector at the top of the probe body. The probe is shown in Figure 2.2.

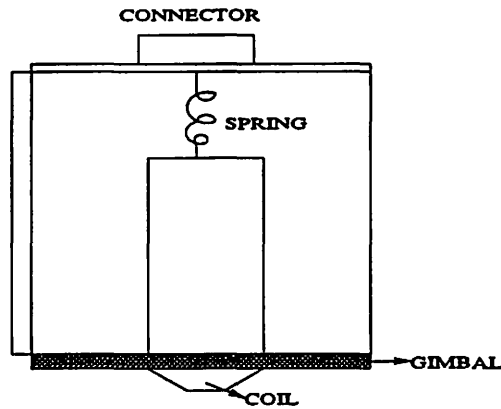


Figure 2.2: PEC probe

The pulsed eddy current probe can be optimized to give a signal of maximum peak amplitude using the design equations as in [1], by correctly selecting the various parameters of the coil. The coil which we have been using has the specifications shown in Table 2.1. The probe and coils for the instrument were designed by Mr. Jay Bieber.

2.3 Scanning System

The scanning system consists of three parts, the mechanical frame holding the probe and the motors, the stepper motors and their control, and the scanning algorithm.

2.3.1 Mechanical construction

The mechanical design and fabrication of the scanner frame were accomplished by the Ultrasonics group at the Center for NDE, Iowa State University. The hollow aluminium mechanical frame is rectangular (22 inches x 11 inches), with an inlet for compressed air attached to it. Four suction cups are fixed onto this frame. The

Table 2.1: Coil parameters

Parameter	Value
Outer Diameter	12.0 mm
Inner Diameter	5.30 mm
No. of Turns	638
Inductance	1.8 mH
Resistance	49.9 Ω
Height of Coil	3.0 mm
Wire Gauge	39

metallic connection between the frame and the suction cups is also hollow, and has a tube coming out of it, one end of which is open to the atmosphere. The mechanism operates by aspiration. This frame is shown in Figure 2.3.

The dimensions of the probe holder are 5 inches x 3.5 inches. Owing to the rectangular nature of the outer frame and the dimensions of the probe holder, the total scannable area reduces to around 13.38 inches x 4.72 inches. An offset of about 5 mm needs to be incorporated to account for the errors in the distance calculations while mapping from screen to physical coordinates. If such an offset is not taken, the probe might run into the edges of the frame, thus considerably damaging the outer rim.

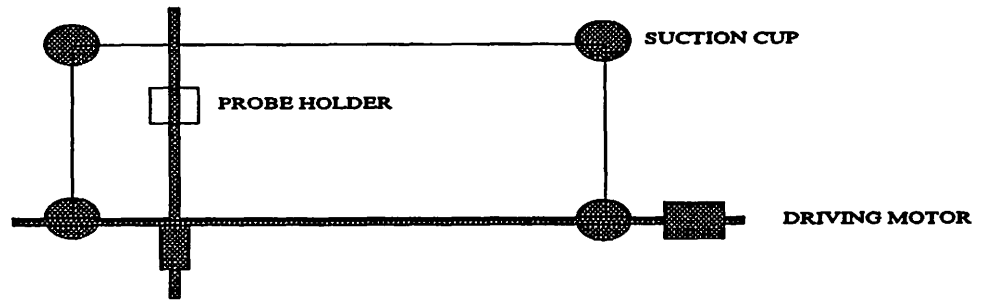


Figure 2.3: Scanner frame

2.3.2 Stepper motors and their control

The stepper motors are commercially available motors and are controlled by an AT6400 controller card. The motors are connected to driver boxes which are in turn connected to an AUX box which receives signals directly from the controller card. The stepper motors can attain a maximum speed of about 17 inches per second. The software used to control the motors uses a macro language specific to this family of controllers. Strings containing specific instructions are written to the card. The card contains a series of status registers which can be used by the control software to monitor the status of the motors at any given time.

2.3.3 2-D scanning algorithm

The scanning system is a two-dimensional scanning system. The user chooses a rectangular area to be scanned using a mouse to perform a click and drag operation in a representation of the scan area on the computer. The scan is executed in a serpentine fashion. Conventional raster scans take more time than serpentine scans. Since the instrument needs to complete the scan as fast as possible, a serpentine

scan was used. In order to build a generic algorithm we need to understand the scan plan and then design an algorithm to execute this scan plan. First of all, we need to come up with an equation which relates the number of scan lines to some parameter we know a priori. In our case, this is the number of scan lines in any of the two dimensions. We know this since this is given by the user. Once the total number of scan lines is calculated, we can allot unique numbers to each scan line. This is shown in the figure. Each scan line corresponds to a particular direction and distance depending on its scan line number. After the completion of each scan line we increment the scan line number. This new number dictates the motion of the motors in the appropriate distance and direction.

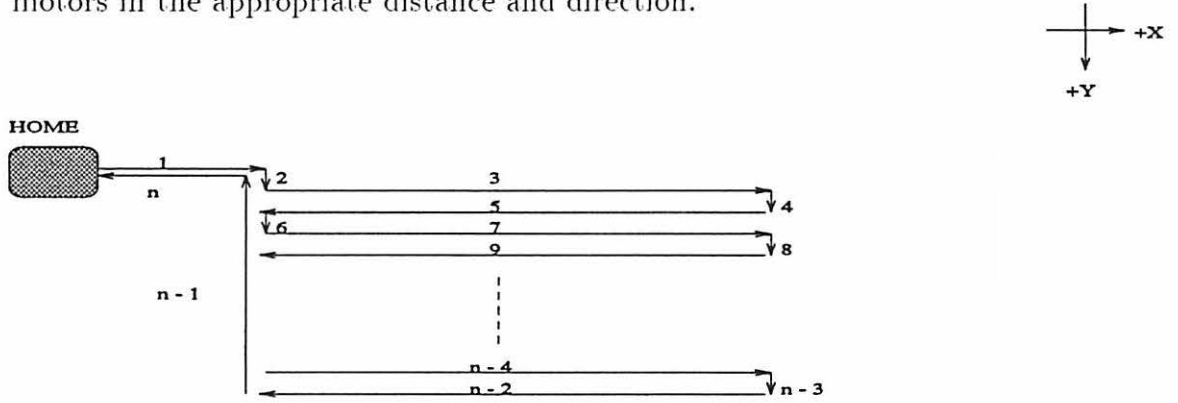


Figure 2.4: Serpentine scan

Every scan begins from the probe's home marked HOME. All scan lines are given a unique number, with the ordering as shown in Figure 2.4. The direction and distance the motors have to move is dictated by these numbers. A closer look at these scan line numbers will give us the following feature by which we distinguish each scan line: If n is the number of a scan line in the $+X$ direction, then

$$n(\text{mod}4) = 3 \quad (2.1)$$

If the scan line is in the -X direction, then

$$n(mod4) = 1 \quad (2.2)$$

Lastly, if the scan line is in the +/- Y direction,

$$n(mod2) = 0 \quad (2.3)$$

If α is the number of scan lines in the +Y direction, and β the number of scan lines in the +/- X direction, we can derive an expression for n , the total number of scan lines. For now we will consider only the area to be scanned. Hence, the path from the home to the area to be scanned and the return path will be ignored. We shall include this later as this is just a fixed number of scan lines. There are two cases to be considered, one is when α is even and the other when it is odd. When α is even, then we need to add extra scan lines since for the α th jog scanning has not yet been done in the X direction. Hence, when α is even, $\alpha = \alpha + 1$. β will always be $\alpha + 1$ for all cases of α . Now,

$$n = \alpha + \beta,$$

simplifying, we get,

$$n = 2(\alpha + (\alpha - 1)mod2) + 1.$$

Since the scan always begins from and ends at home, we need to add four more scan lines to the above equation, resulting in the following:

$$n = 2(\alpha + (\alpha - 1)mod2) + 5 \quad (2.4)$$

The value of α can be obtained by the ratio of the scan area length in the Y direction,

$Y_distance$ and the resolution r desired by the user. Resolution is basically the distance between two successive points of measurement. Hence,

$$\alpha = Y_distance/r \quad (2.5)$$

Each time a particular scan line is done, n is decremented, and a new scan line is begun depending upon the new value of n . The stepper motors are controlled through the AT6400 card manufactured by Compumotor. The motors have an AUX board and drive controllers. The software used to control the motors is written in a macro language specific to this family of controllers [8] and is embedded in the main program.

2.4 Data Acquisition System and PEC Card

The data acquisition is accomplished through a 16-bit, 1MHz, A/D board manufactured by Analogic Corporation. The code for the data acquisition is in the form of a DLL[9]. The software for this DLL was written by Mr. Mark Kubovich. Since the probe is excited by a 1kHz pulse, the time duration of interest is 0.5 millisecond. The A/D board has 1MB of memory of its own. All the samples are stored in this memory and we are presently using programmed i/o transfer to transfer the data from the local memory of the A/D card to the main memory of our PC. Further details of data processing and signal analysis are discussed in the chapter on Signal and Image Processing.

A custom built printed circuit card, called as the PEC card, has been built basically to amplify the signal obtained from the PEC probe and to ensure all triggers are synchronized properly. This card also generates the coil excitation. This card

was designed and built by Mr. William Ward.

2.4.1 Daughter board control

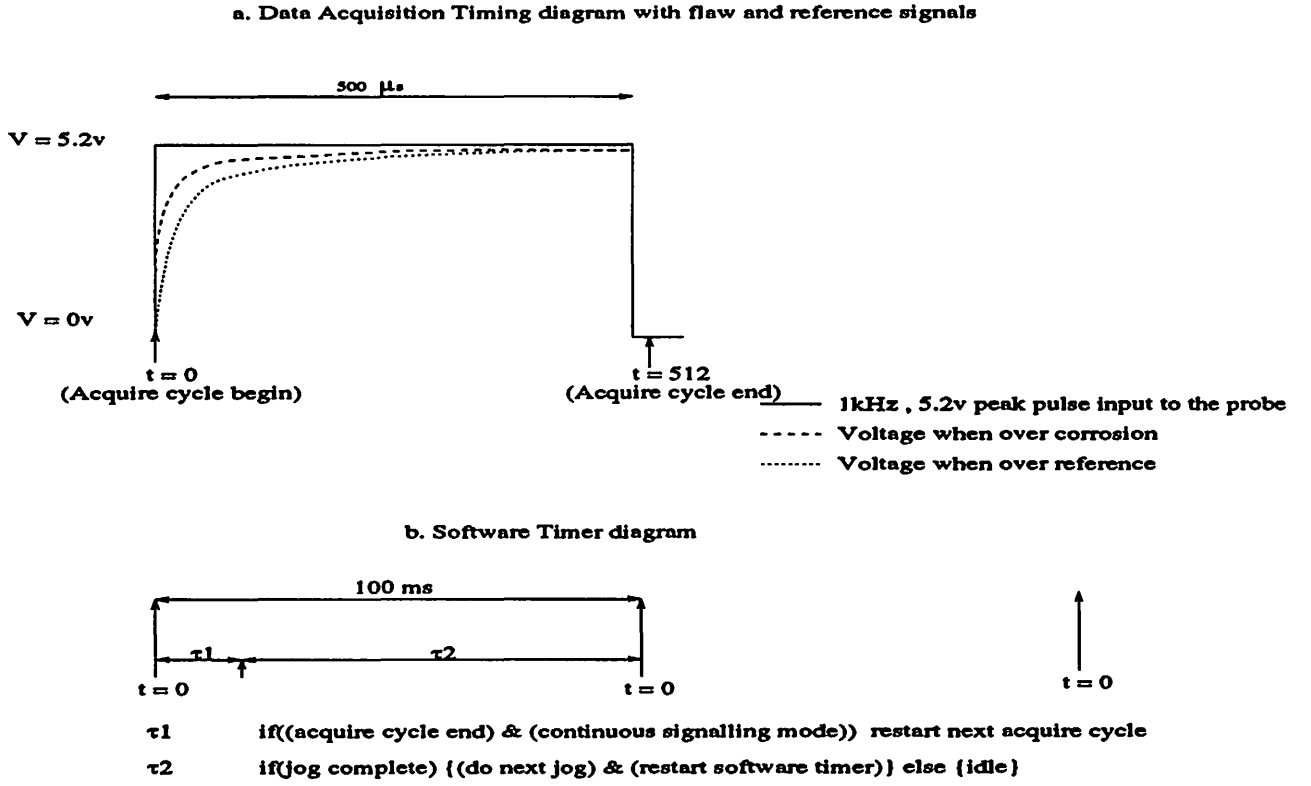
The latest version of the PEC card incorporates a daughter board controller, which enables the user to set the gain on the amplifiers through the software. This daughter board also enables users to set the voltage level and repetition rate of the excitation through the software. The board is mounted on the original PEC card.

CHAPTER 3. SIGNAL AND IMAGE PROCESSING

Detection of flaws on the scanned surfaces is based on interpretation of the signals acquired from the probe. The peak amplitude of the signal is recorded and converted into a pseudo-color image. High resolution images require more data, sampled as fast as possible. Therefore the speed of data acquisition affects the resolution and hence accuracy.

3.1 Data Acquisition and Timing

The timing diagram in Figure 3.1 shows the different signals which we acquire and their differences. The probe is driven with a 1kHz, 5.2 V square wave. Data acquisition cycles begin at the upward transition of the input pulse triggered by a clock on the data acquisition board. The sampling rate of the A/D board is 1MHz, hence, every 1 microsecond a new sample is taken. The sample is stored as a floating point number in the local memory of the data acquisition card. A total of 512 samples are acquired during the first one-half cycle of the square wave, then this data is transferred to the memory on the PC by a PIO (programmed input output) operation. The A/D board can also be programmed to transfer data in a DMA (direct memory access) mode, which is much faster [9]. Difficulties were encountered using this mode of operation and therefore we used the slower PIO for data transfer.



The voltage across the 1Ω resistor connected to the coil is shown in Figure 3.1 for two different cases. When the probe is on an area of hidden corrosion, and when the probe is on a reference spot. A reference signal is obtained by placing the probe on a particular part of the structure where there is no corrosion. The important thing is that the signal of interest is a very small change in the large current - about 1 to 2 %. Thus, we subtract a reference background signal to permit us to see the actual signal.

$I_{flaw}(t)$ is the current when the probe is on a spot with hidden corrosion, and $I_{NULL}(t)$ is the current when the probe is on a reference spot, then,

$$\Delta I_{diff}(t) = I_{flaw}(t) - I_{NULL}(t) \quad (3.1)$$

is the signal of interest. The output voltage is taken across a 1Ω resistance. Hence, the voltage represents the current. The signal subtraction is done in software and is done continuously. The reference is stored and subtracted from in coming signals. This is done whether there is a flaw or not. The acquiring of the reference trace is left to the discretion of the user. The user can set the number of times the signal needs to be averaged. The more the number of averages the lesser the noise but the more time it takes to finish the process of acquiring a trace. A difference signal after 100 averages is as shown in Figure 3.2. This signal has a peak signal amplitude of about 20 mV and has a zero crossing time of 139 microseconds.

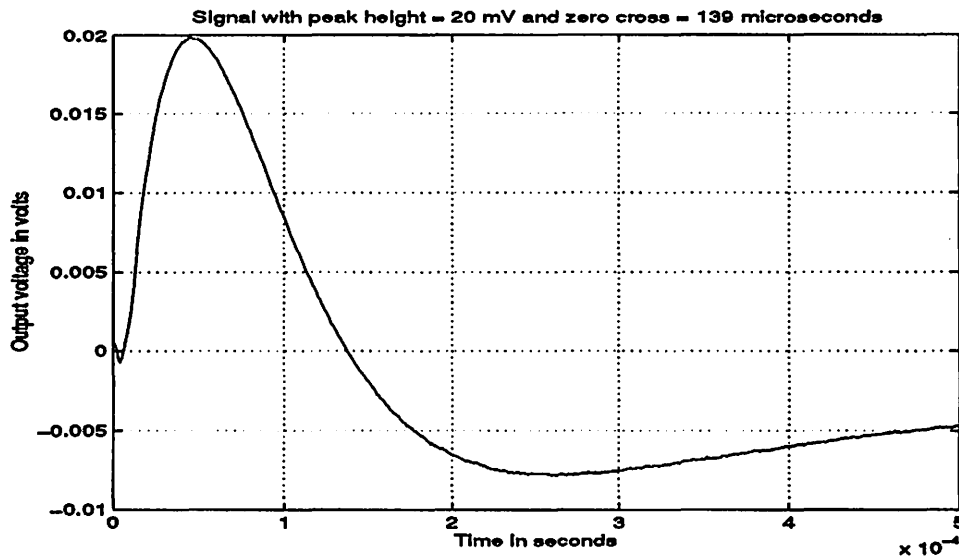


Figure 3.2: Difference Signal

After the data acquisition module has finished acquiring all the data, it sends a software interrupt to the PC. This interrupt signifies that the data to be acquired from the probe is ready for software processing. The ISR (interrupt service routine) for this interrupt displays the signal on the screen and performs signal processing routines

on this signal which will be discussed below. After all this is done, the software timer comes into play. This timer is a built-in Windows timer, which ticks every 100 milliseconds. A separate routine services the interrupt caused by the ticking of the software timer. The actions taken are shown in part b. of Figure 3.1. We simulate parallel operation of both the data acquisition and the probe motion with the help of this software timer. The routine checks if the data acquisition cycle has ended. If it has, and if the user wants signals to be obtained continuously (as in a scanning mode), the routine triggers the beginning of another acquire cycle and continues to monitor the scanning motors. Motors are monitored even if the acquire cycle has not ended. If the motors have finished their previous jog as dictated by the scanning algorithm, the routine gives a command to start the next appropriate jog, and it triggers the software timer, thus transferring control back to itself. If a scan is complete, the motors are not monitored. Parallel monitoring is possible because the A/D board and the motor controller run as two separate entities whose status can separately be checked. Thus, as the probe scans over a surface, signals from it are being captured, and stored.

It should be noted that the data acquisition system requires to be set at a signal averaging rate of 10 or less for the system to faithfully record data in the scanning mode. This is true since, the speed of the scanning motors is computed as,

$$v_s = \frac{r}{100 * 10^{-3}} \quad (3.2)$$

In the above equation r stands for the resolution of the scan in millimeters, as desired by the user. The number of averages is not taken into account in the velocity calculation. It takes about 100 milliseconds or 1/10th of a second for the system to finish acquiring, manipulating, storing and displaying data at 10 averages. If more

than 10 averages are acquired, the data acquisition will fall out of sync with the scan.

3.2 Signal Storage, Plotting and Extraction

Each signal is an array of 512 floating point numbers, 500 of which are used. The software simulates an oscilloscope and displays the signal trace on the screen. The sensitivity (1 - 500mV/division) and time base (2 - 50 microseconds/division) can be changed as desired by the user. All signals are stored in a common buffer explicitly declared in the software. At any given time this buffer can accommodate only one complete signal. The user can store and display a reference trace. In order to form a C-scan image of the whole scanned surface, we need to have at our disposal the signal at each point. We examined a few potential methods to record and store the signal at each point. The first method is obviously the brute force method wherein we store all the signals. With the present scanner, a typical scan takes about 30 minutes to complete (at a resolution of 2mm). We get a new signal every 100 milliseconds. A Simple calculation results in a requirement of about 37MB of memory to store all the signals. This is not feasible.

Another approach is to compress each of the signals by some efficient mechanism and later decompress it for future use. We could fit a curve to the signal. It so happens that the signal we obtain has a complex shape and it is difficult to find a curve to fit it exactly. Also, since precision is important this method was not pursued.

Theoretical insight into pulsed eddy current measurements revealed that there were possibly only three parameters by which the details of a $\Delta I(t)$ trace can best be represented. These were as follows:

t_{zc} , the time at which the signal crosses zero after reaching a maximum value.

We refer to this as the zero crossover time.

PH , the maximum level or height reached by the signal and lastly,

t_{PH} , the time at which PH occurs.

As of now, we extensively use the first two parameters to identify corrosion on scanned surfaces.

3.2.1 Detection of t_{ZC} and PH

The above two parameters form the basis for creating images representing corroded areas of the material scanned. When such importance is being given to these two parameters, their accurate detection and measurement is crucial. As mentioned in the previous sections, the time trace is basically an array of floating point numbers. We need the software to precisely find the values of t_{ZC} and PH in real time. We accomplish this in two stages:

STAGE I: Traverse the array and find the point with the maximum value, denoting it by a tuple, say, $(loc_{PH}, localmax)$. loc_{PH} gives the position where $localmax$ occurs. From this point of local maximum travel towards increasing time to find the first point when the signal makes a transition from a positive value to a negative value. Call this point by another tuple $(loc_{ZC}, localZC)$.

STAGE II: A close look at the shape of the trace justifies an assumption that the topmost part of the trace looks like a quadratic and the part where the signal crosses zero can be approximated by a straight line. From the preliminary values obtained from STAGE I, we take data points $SIGNAL[loc_{PH} - 25] \dots SIGNAL[loc_{PH} + 25]$ and perform a least squares quadratic fit on the points. The new maximum of the quadratic is found and this is our peak height PH . The same procedure is carried

out to find t_{zc} , except that we perform a linear least squares fit [10] to the set of data points $SIGNAL[loc_{zc} - 25]...SIGNAL[loc_{zc} + 25]$, and find where this line crosses zero. These two numbers, PH and t_{zc} are the only data stored from each point in the scan. In subsequent sections we describe how they are used to form images.

3.3 Time and Frequency Domain Toggle

It is sometimes convenient and helpful to be able to view the frequency domain variations in the signals[11]. The PEC instrument is capable of collecting data from the probe in the time domain. This signal, as has already been mentioned, is a record of the change of current in the coil. In the time domain, we have a pulse voltage V across the coil which is defined as follows:

$$V = \begin{cases} 1 & \text{for } t \geq 0, \\ 0 & \text{otherwise} \end{cases}$$

Taking the discrete Fourier transform [12] of the above equation we get:

$$V(\Omega) = \pi * \delta(\Omega) + \frac{1}{1 - e^{-j\Omega}} \quad (3.3)$$

for $-\pi < \Omega < \pi$. If $\Delta I(\Omega)$ is the discrete Fourier transform of the time domain difference trace, then we get the following equation for $\Delta I(\Omega)[1]$

$$\Delta I(\Omega) = V(\Omega)\Delta Y(\Omega) \quad (3.4)$$

$\Delta Y(\Omega)$ being the change in the admittance of the coil corresponding to the change in the current through the coil. Hence,

$$\Delta Y(\Omega) = \Delta I(\Omega)(1 - e^{-j\Omega}) \quad (3.5)$$

In the above equation the term $\delta(\Omega)$ vanishes since this exists only at $\Omega = 0$. When $\Omega = 0$, i.e. at zero frequency, there is no change in the impedance (hence admittance) of the coil, hence the term $\delta(\Omega)$ can be neglected for our purposes. The process described above is basically a transformation from the time domain current signal, $\Delta I(t)$ to the frequency domain admittance signal, $\Delta Y(\Omega)$. Previously, we had a signal representing the change in the current in the time domain, now we have a signal representing the change in the admittance of the coil in the frequency domain. We can view the $\Delta Y(\Omega)$ signal over a frequency spectrum ranging from 0 to 500 kHz.

3.4 Image Formation

After a scan has been completed, an image of the scanned surface is displayed on the screen. The image is pseudo-color coded with a palette of colors, each denoting a particular amplitude of the signal. While the scan is in progress, data is continuously being written into an image file with a .img extension. A typical image file is shown in Table 3.1.

Table 3.1: File format of a .img file

HEADER: Rectangular Scan area coordinates			
peak H	x-position	y-position	zerocross
0.012756	0.000000	0.000000	160
0.012665	0.039250	0.000000	165
0.012970	0.039250	0.000000	173
.	.	.	.

As the probe moves over the surface, its response to the 1kHz pulse is continuously acquired by the system. The peak height and the zero crossover time are extracted and stored in a file, along with the position of the probe with respect to the HOME of the scan area, as discussed in the scanning algorithm. The header

information has the screen coordinates of the area selected for the scan. This information is used when the user wants to view the image of a scan previously acquired. Any image, in order to be recognized by the PEC software, has to be in the .img format. Once an image file is written, we need to actually display its contents on the screen of the computer. We use a mapping function, f , to perform the physical to screen mapping. Thus,

$$\begin{aligned} f(x_{pos}, y_{pos}) &= (x_{screen}, y_{screen}), \\ f^{-1}(x_{screen}, y_{screen}) &= (x_{pos}, y_{pos}) \end{aligned}$$

The scan area has a predefined rectangular area of length 340 and width 120 pixels. This is hardwired into the software. After allowing suitable offsets in the scanner, the total scannable area is reduced to 34 cm x 12 cm. Thus, every square millimeter scanned on the sample corresponds to 1 pixel on the screen. The scanner accepts distance measured in inches. Thus, every pixel corresponds to 0.03937 inches. If the upper left hand corner and bottom right hand corner coordinates of the rectangle describing the scan area are (x_1, y_1) and (x_2, y_2) respectively, then, the screen coordinates (x_{screen}, y_{screen}) of a physical point (x_{pos}, y_{pos}) are,

$$x_{screen} = x_1 + \frac{x_{pos}}{0.03937} \quad (3.6)$$

Similarly,

$$y_{screen} = y_1 + \frac{y_{pos}}{0.03937} \quad (3.7)$$

Thus, the screen coordinates of the probe position are found. Continuing, we now need to allocate to a given pixel a color corresponding to the amplitude of the signal. We have an 11-level color scale, which could easily be extended to a n-level scale. The user needs to specify a minimum and maximum signal level he expects,

Table 3.2: Color index lookup table

Index	Red	Green	Blue
1	255	0	0
2	255	255	0
3	128	128	0
4	0	255	0
5	0	255	128
6	0	128	0
7	0	128	128
8	0	0	255
9	0	0	128
10	128	0	128
11	255	0	255

let us call this ΔI_{min} and ΔI_{max} . If at the point (x_{pos}, y_{pos}) a signal of peak height PH is obtained, then the color index at that point is computed as,

$$Index(x_{pos}, y_{pos}) = \left\lfloor \frac{n * PH}{\Delta I_{max}} \right\rfloor \quad (3.8)$$

The above equation will be modified while discussing amplitude gating so that no feature of the signal is excluded from the image. This modification will make use of ΔI_{min} mentioned above. Once the index corresponding to the pixel is found, a simple algorithm to convert the index to the RGB values is used. Table 3.2 shows the lookup table used to convert from the index to a RGB value. The software displays the whole image pixel by pixel until the whole scan area is covered.

3.4.1 Pixel to pixel interpolation

The quality of the image produced after a scan is largely dependent on the resolution of the scan. Resolution is basically the physical distance between two successive

signal acquisitions. Higher resolution scans yield sharper images, a desirable feature. There seems to be only one major disadvantage to higher resolution, and that is the speed. The time interval between two successive data acquisitions can be at least 100 milliseconds, which is a fixed quantity at 10 averages of the signal. Hence, by increasing the resolution, we make the distance to be traveled by the probe lesser for the same duration of time. In other words, we must decrease the speed of the scan. The resolution is the same both in the x and y directions. In order to obtain a signal every pixel, we need to have a resolution of 1 millimeter. Such a high resolution reduces the speed of the scan considerably. Typically, a scan of 34 cm x 12 cm takes almost a full hour to finish.

To improve the speed, we might try to use fewer data points. A smaller number of data points is justified, since sampling at a resolution of 1 mm, with a probe of diameter greater than 1 cm, is a case of over-sampling. The pixels in between two points of signal acquisition could be interpolated [13]. The optimum resolution is approximately the radius of the coil. By experience, we have settled on a resolution of 2 mm with interpolation. There are different types of interpolation techniques. We have implemented two types of interpolation, flat and linear interpolation.

FLAT INTERPOLATION: In this method, a fixed number of pixels around the data pixel(point of signal acquisition) are all indexed to the data pixel. The number of such pixels depends on the resolution of the scan. Let I_1 be the index at a point (x_n, y_n) , and I_2 the index at point (x_{n+n_0}, y_n) . Both the indices are computed from the actual signal. All pixels on and within the square defined by the coordinates, $(x_n - \frac{n_0}{2}, y_n - \frac{n_0}{2})$ and $(x_n + \frac{n_0}{2}, y_n + \frac{n_0}{2})$ are allotted index I_1 . The square coordinates define the upper left and bottom right corners. Similarly, all pixels on and within the

square defined by the coordinates, $(x_{n+\frac{n_0}{2}}, y_{n-\frac{n_0}{2}})$ and $(x_{n+\frac{3n_0}{2}}, y_{n+\frac{n_0}{2}})$ are allotted index I_2 . This method of interpolating the pixels exaggerates the digital aliasing, which is not a pleasing effect.

LINEAR INTERPOLATION: As the name suggests the idea behind this method is to allot pixels indices such that their indices lie on a straight line. First, let us consider interpolation only in the x direction (refer to Figure 2.4). Consider a point (x_1, y_1) with index I_1 , and another point (x_2, y_1) with index I_2 . Let the index at any arbitrary point (x_*, y_1) between these two points be I_* . Then,

$$\frac{I_* - I_1}{x_* - x_1} = \frac{I_2 - I_*}{x_2 - x_*} \quad (3.9)$$

Thus,

$$I_* = \frac{(x_* - x_1)I_2 + (x_2 - x_*)I_1}{x_2 - x_1} \quad (3.10)$$

By implementing the above equations, all the pixels in the x direction can be interpolated. This gives us the pixels on the path traced by the probe. We need to build a continuous image. To do this, we need to interpolate in the y direction also. Unfortunately the size of the image is so large that individual y positions of pixels are not stored. To overcome this problem we use a software trick. Since the image now consists only of horizontal lines, we get the index of each pixel on two successive horizontal lines, but lying on the same vertical line. Then we interpolate all pixels in between by using the above interpolation equations. The interpolation is purely based on the index and not on the peak height or the zero crossover of the signal.

3.5 Image Conversion and Storage

Images displayed on the screen of the computer can be stored as a *.bmp* file in a standard Windows acceptable format, called the DIB (device independent bitmap) format [14]. The abbreviation is self explanatory and refers to the device used by a display to represent a color. It is desirable to be able to convert the image obtained after a scan into a standard image file format. The reason is so that the file can be processed and modified by standard image processing software if so desired by the user. Typical modifications are the size of the image, its sharpness, brightness, hue, saturation, and so on. In fact most image processing packages can also convert this color image into a gray scale image.

The bitmap file has the following standard file format:

```
BITMAPFILEHEADER bmfh;  
BITMAPINFOHEADER bmih;  
RGBQUAD aColors[];  
BYTE aBitmapBits[];
```

The various structure definitions used in our applications are as follows:

struct *BITMAPFILEHEADER* members

```
bftype = "BM";  
bfsiz = (BITMAPINFOHEADER.biWidth*BITMAPINFOHEADER.biheight)*  
3 + sizeof(BITMAPINFOHEADER) + sizeof(BITMAPFILEHEADER);  
bReserved1 = 0;  
bReserved2 = 0;  
bOffBits = sizeof(BITMAPFILEHEADER)+BITMAPINFOHEADER.biSize;
```

```
struct BITMAPINFOHEADER MEMBERS
```

```
biSize = 40;
```

```
biWidth = 380;
```

```
biHeight = 170;
```

```
biPlanes = 1;
```

```
biBitCount = 24;
```

```
biCompression = BIRGB;
```

```
biSizeImage = 0;
```

```
biXPelsPerMeter = 10000;
```

```
biYPelsPerMeter = 10000;
```

```
biClrUsed = 0;
```

```
biClrImportant = 0;
```

After writing these structures byte by byte, the RGB value of all the pixels are to be written into the file. More information about these header formats can be obtained from the Microsoft Graphics File Formats help menu. At present we do not use any image compression scheme, and thus the bitmap images can be quite large.

3.6 Amplitude and Time-Gating

A gate is basically an opening which allows a signal to pass if it satisfies a particular criterion. In our case we have two criteria : the amplitude of the peak and the zero-crossing time. We shall first discuss the Amplitude Gating feature.

3.6.1 Amplitude gating

The coloring scheme we have used until now has some limitations. This is best illustrated by an example. Consider a signal with a user specified maximum level of x millivolts. Let the number of color indices be n . Now, if we have two signals of peak heights x_0 and x_1 , the color indices of both these signals will be equal, if

$$|x_0 - x_1| \leq \frac{x}{n} \quad (3.11)$$

In such a situation, there is no way we can distinguish between the two unique signals in the image. The one straightforward solution is to reduce the maximum level. This will work, but there seems to be a problem. The lower bound on the signal is unknown. It can so happen that a signal might sometimes be negative and sometimes positive (this depends on the reference).

To avoid the ambiguity, the user needs to specify both the maximum and minimum expected signal levels. This is the Amplitude Window or Gate. A similar implementation of this concept can be found in [6]. Thus Equation 3.10 needs to be modified to:

$$Index_{(x_{pos}, y_{pos})} = \left\lfloor \frac{n * (PH - \Delta I_{min})}{(\Delta I_{max} - \Delta I_{min})} \right\rfloor \quad (3.12)$$

All acquired signals will be allotted indices after splitting the user specified window into various signal levels each corresponding to unique indices. This window can take any values from $-\infty$ to $+\infty$. Thus, the same image can be redrawn with a different amplitude window to reveal a different feature of the sample, since the magnitude of the signal depends on the nature and degree of corrosion.

3.6.2 Time-gating

As has been emphasized before, the zero crossover time provides information regarding the nature and type of the signal. In particular, it helps to identify the depth of the flaw that is causing the signal. This is illustrated in Figure 3.3, which shows us the variation of the zero crossover time with the peak height of the signal. It can be seen that there are distinct time zones in which the zero crossovers occur for signals reflected from different layers. Signals obtained due to air gap between two layers of aluminium have zero crossovers that occur very early in time. Flaw signals from deeper below the surface tend to occur later in time as is shown in the figure.

This grouping of data, helps us to distinguish the source of flaw signals, and by time-gating we can extract an image showing only flaws from the top layer or flaws from the bottom layer. To be able to achieve this time discrimination, we make use of time-gating. The user specifies a time window, say t_{min} and t_{max} . All signals whose t_{zc} does not lie within this time window are given an index which corresponds to no corrosion, whereas all signals that lie within this time window are given indices prescribed by the amplitude-gating algorithm. The combination of both amplitude and time windowing can be succinctly represented by means of an equation:

$$I_{(PH,t_{zc})} = \begin{cases} \lfloor \frac{n*(PH-\Delta I_{min})}{(\Delta I_{max}-\Delta I_{min})} \rfloor & \text{if } (t_{min} < t_{zc} < t_{max}) \text{ and } (\Delta I_{max} < PH < \Delta I_{min}), \\ n & \text{if } PH \geq \Delta I_{max} \\ 0 & \text{if } (PH \leq \Delta I_{min}) \text{ or } (t_{zc} < t_{min}) \text{ or } (t_{zc} > t_{max}) \end{cases} \quad (3.13)$$

This is the final equation which represents the way in which an image is displayed from the signals acquired during the scan process. Thus, amplitude-gating identifies the amount of corrosion and time-gating identifies the location of corrosion.

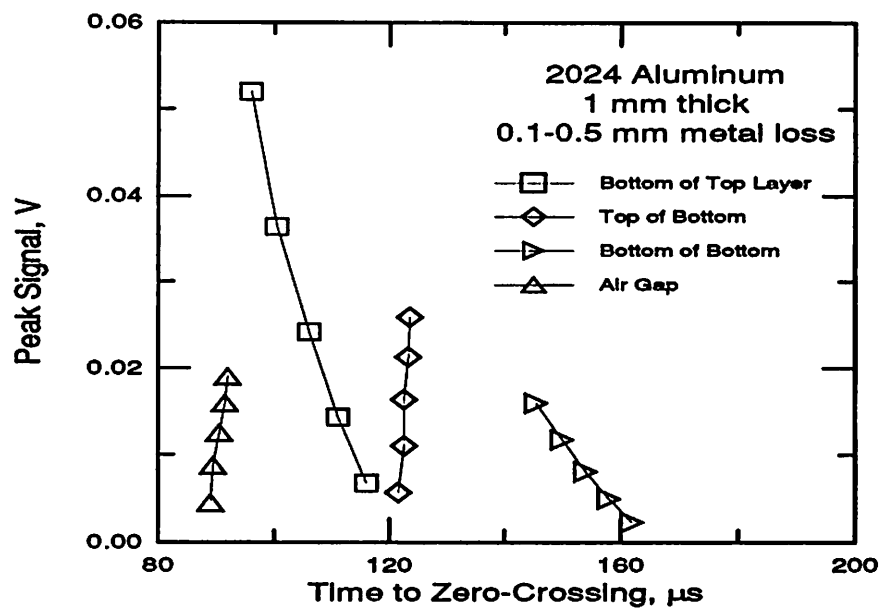


Figure 3.3: Plot of peak height vs. zero crossing time

CHAPTER 4. SOFTWARE ELEMENTS AND INSTRUMENT OPERATION

The software for the PEC instrument is written in *VisualC++TM*, version 1.0. This is basically a development system for *MS-WindowsTM*. Ours is a *WindowsTM* application and uses the Microsoft Foundation Class Library [15]. Before discussing the software organization of the instrument, some familiarity with the Class library and the development environment is needed.

The library's classes broadly fall into Root Class, Application Architecture Class, Visual Object Class, General Purpose Class and Object linking and embedding classes. Depending upon the needs of the application, we can choose what classes to include. The Microsoft Foundation Class (MFC) Library is built on top of the C-language application programming interface for Windows, and is object oriented with reduced programming effort. MFC has classes that represent standard objects like windows, dialog boxes, etc. Member functions of these objects help us design our own interface elements and we can manipulate them as need be.

The other key elements used during the process of application building are the Class Wizard and the Application Studio [14]. The Class Wizard helps us create message handling functions, and to associate member functions with events associated with the application. The application studio helps us build the framework of

the organization of menu items, dialog boxes, etc. We can construct our own bitmaps and icons to add some color and glamour to the application. The debugger is user-friendly and has the ability to set multiple breakpoints, with individual variable inspection, and modification. With this brief introduction to the software development environment let us look at the internal software organization on the PEC instrument.

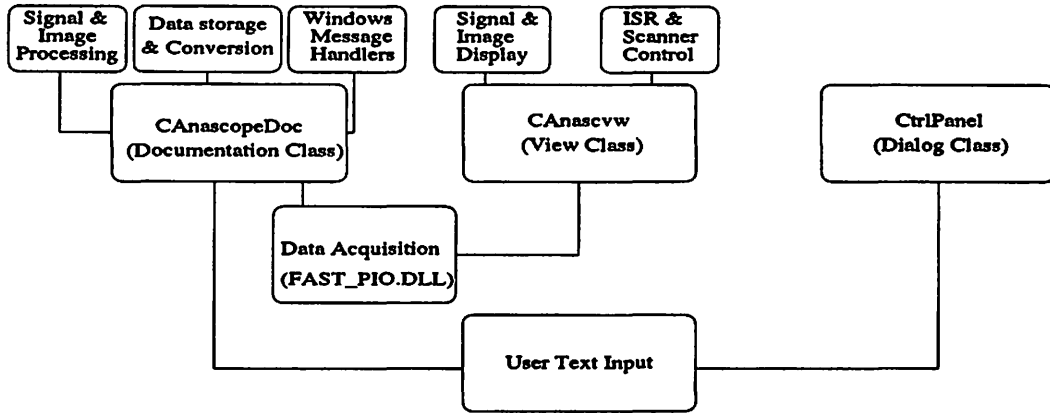


Figure 4.1: Software organization on the PEC instrument

4.1 Software Organization

Figure 4.1 shows the main classes used in the application, and their functions. The three main classes used are named CAnascopeDoc, CAnascvw and CCtrlPanel. CAnascope is basically a documentation class, CAnascvw is a View Class and CCtrlPanel is a Dialog Class. The Anascope user interface consists of what we call a control panel. All user inputs such as filenames, gating magnitudes, scan resolution, etc. are handled by the control panel.

The documentation class deals with the various data which we obtain from the instruments as well as from the user. Data are obtained from the instruments in the

form of a signal. The documentation class consists of routines like *FourierTransform()* which converts the time domain trace into a frequency domain trace. Writing data onto a file is also handled by the documentation class. The various files which need to be written to are the data files and the image files. The data files contain the trace data and the image files contain the image data (Refer to Table 3.1). As far as the image processing part is concerned, the documentation class has a member function *DisplayImage(CDC*)* which interpolates the pixels and displays the image on the screen.

The view class has two main functions, one to control the signal and image display routines and the other to handle the interrupts. Signals are displayed on a screen which simulates an oscilloscope screen. The graphical user interface while choosing a scan area is accomplished in this main class. The view class passes on a pointer to a class of device context objects (CDC *) onto the documentation class to display the image on the screen. This is convenient, since the documentation class has the necessary data and there is an algorithm to implement to display the image. The view class handles two interrupts, the first one being the data acquisition card interrupt, and the other being the timer interrupt. The data acquisition card software (FASTPIO.DLL) interrupts the windows application after data acquisition. The interrupt service routine (ISR) plots the data in the buffers on the screen. This ISR is called *OnAveragesDone()*. The timer ISR is called *OnTimer*. Simultaneous control of the data acquisition and the scanning motors is done through this routine.

4.2 Software Implementation Details

4.2.1 Motor control software

As mentioned in Chapter 2, all control commands are sent to the motor controller in sets of strings. A routine called *SendAT6400Block()* sends the string contents character by character till the end of command is reached. Motor control is based on the software timer which has a repetitive rate of 100 milliseconds. At the start of every timer cycle, the status of the motors is monitored. If the motors are still in motion, then the timer is restarted, else the scan line number is incremented and a new command string is sent to the controller. A new timer cycle is now started. The probe can be moved to any desired point within the frame of the scanner. This is accomplished by writing an event handler code to the event of the middle mouse button being pressed. If the mouse points within the scan area and the middle mouse button is pressed, then the probe moves to the appropriate physical location. A red dot of dimension 1 pixel appears on the screen, making the user aware of his current position. The program also continuously monitors the actions desired by the user. If the user wishes to abort a scan, the system completes the present jog and moves back home. The user can acquire the reference trace any time during the scan, since the data acquisition system will always be on. Signals are displayed in real time on the screen even when the probe is in motion.

4.2.2 Data storage software

Signals need to be processed after they are acquired. The data acquisition software sends an interrupt to the main program signifying the end of the data acqui-

sition. The interrupt service routine now processes the signal and finds the zero crossing time and the peak height using the schemes described in Chapter 3. After extracting this data from the signal, the location of the probe is obtained from the motor controller. All this information is written in a data file and the whole process of acquiring the data is restarted. The header information of the .img file described in Chapter 3 is written when the user presses the begin scan button.

4.2.3 Data retrieval and interpretation software

After the scan has ended the image display module takes over control from the scanning module. This module first gets the time and the amplitude gate desired by the user. It then reads each line of data from the .img file and passes the peak height and the zero crossing time through the gating algorithms described in Chapter 3. A color index is allotted to each set of data points. These points are displayed in the locations corresponding to the location of the probe when this signal was acquired. The image is made continuous by coloring the surrounding pixels by the algorithms described in Chapter 3.

4.2.4 Image storage and redisplay software

Images can be stored in the *WindowsTM* bitmap format as discussed earlier. Images currently displayed on the screen can only be stored. The program gets the RGB value of each pixel in the scan area and stores them in a buffer. After all pixel values are stored in this buffer, its contents are written into a .bmp file. Since, storing the RGB values of all pixels can take a lot of memory space, we dynamically allocate memory and write into the .bmp file in parts. The image is redisplayed in the same

way as it is displayed after a scan is over. The only difference being that the old dimensions of the scan stored in the header file are now used.

4.2.5 Calibration software

This module takes the user through the process of calibration. This software is explained in greater with the equation and graphs while discussing calibration in Chapter 5.

4.2.6 Signal display and user interaction software

The signal acquired is displayed on the screen by simulating an oscilloscope display. This capability existed in the first prototype built by Mr. Mark Kubovich. In the present version the capability of viewing the signal in the frequency domain was built. We can view both the real and the imaginary traces of the quantity in the frequency domain.

The user interacts with the instrument by selecting a scan area. The software allows the user to select an area only in a specific region of the screen, the scan area. Other forms of the user interaction include range selections, probe mode selection and image mode selection and the input of the scan parameters. Standard *VisualC++TM* functions monitor user actions and text inputs.

4.3 Instrument Operation

In this section we briefly discuss the steps necessary to operate the PEC instrument, safety precautions and performance. Before starting the instrument, care should be taken to check the motor and probe connections. The probe should be in

the HOME position before startup. After startup, to perform a scan the following steps have to be taken:

- **Start Signal Acquisition:** The probe mode should be set (Absolute or Reflection), the number of averages set to 10 (optimal performance). The proper time base and sensitivity should be chosen so that the whole trace is visible on the screen. The signal acquisition should be set to continuous mode.
- **Referencing:** This is by far the most important step. A good null has to be chosen, i.e., a spot with no corrosion has to be found. The probe can be moved to this spot by pointing and clicking with the middle mouse button to a point within the scan area of the screen.
- **Scan area selection:** A rectangular/square scan area should be chosen by clicking, dragging and releasing the left mouse button. This must be done before a scan can be carried out.
- **Scan begin:** Before starting a scan, a suitable filename for the .img file has to be given. This should be no more than eight characters in length. The resolution of the scan also has to be set, along with the time and amplitude window sizes. The begin scan button can now be pressed. After the scan has been completed an image is displayed in the scan area which corresponds to the most recent scan taken. The signal at any point on the image can be seen by moving the probe to that particular point. The user can at any time abort the scan by pressing the abort button. This also serves as a button to home the probe.

The same image can be redisplayed with different time and amplitude windows by clicking the “Display Image File” option in the File menu. The same menu item

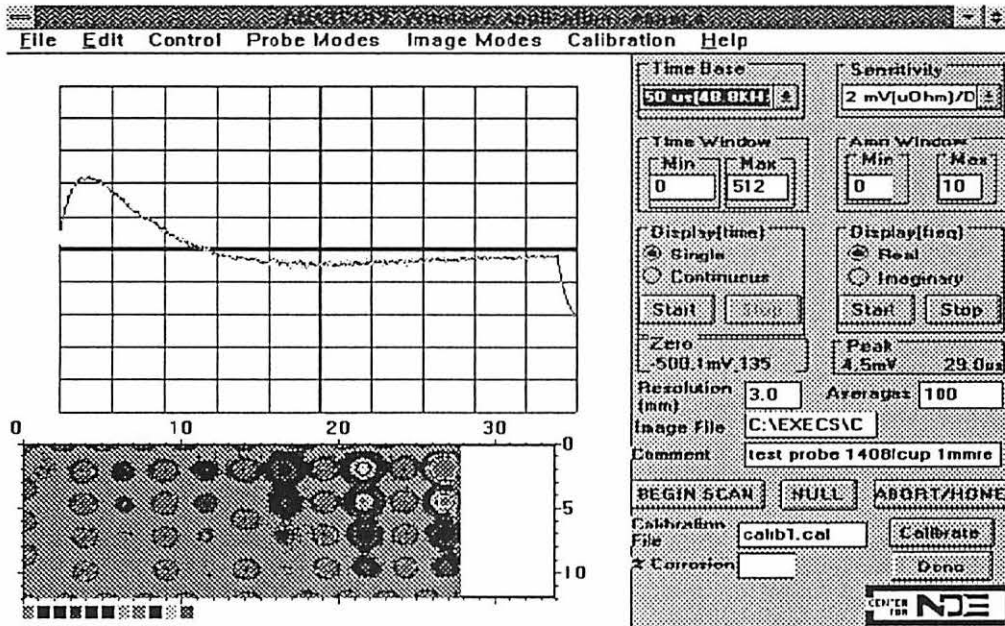


Figure 4.2: Screen dump of the PEC instrument panel

can be used to display any other .img file. Images can be saved in a standard bitmap format by choosing the Save As Bitmap format from the File menu. The time domain and frequency domain traces can also be saved onto data files. If, for some reason, the user wishes to abort the scan while in progress, it can be done by pressing the Abort Scan button. In the frequency domain, both the real and imaginary part of $\Delta Y(\omega)$ can be plotted. A screen dump of the instrument panel is shown in the Figure 4.2.

CHAPTER 5. CALIBRATION AND TESTING

Now that the principles and algorithms used in the instrument have been discussed, we shall turn to some practical results. The PEC instrument has been rigorously tested and has proven itself to be rugged. In this chapter we shall see results from laboratory tests as well as tests on an actual aircraft (field testing). We shall also have a detailed look at the usefulness of the time and amplitude gating, and how it effectively removes signals due to fasteners, rivets and air gaps.

5.1 Calibration

One of the advantages of the PEC instrument is its ability to quantify the amount of metal lost to corrosion. Since we understand the theory behind signal formation [1], the instrument is able to infer the percentage of corrosion present under the scanned surface. To be able to do this accurately, the instrument needs to be calibrated. Calibration is an important tool for any field technician. All that the end user cares about is the actual amount of corrosion, and depending on this, a decision about the need for repair or replacement will be taken. The motivation behind calibration is to make such an operation easy.

Figure 3.3 gives us the dependence of the zero crossover time with the peak height. The points plotted are for different amounts of corrosion (expressed as a

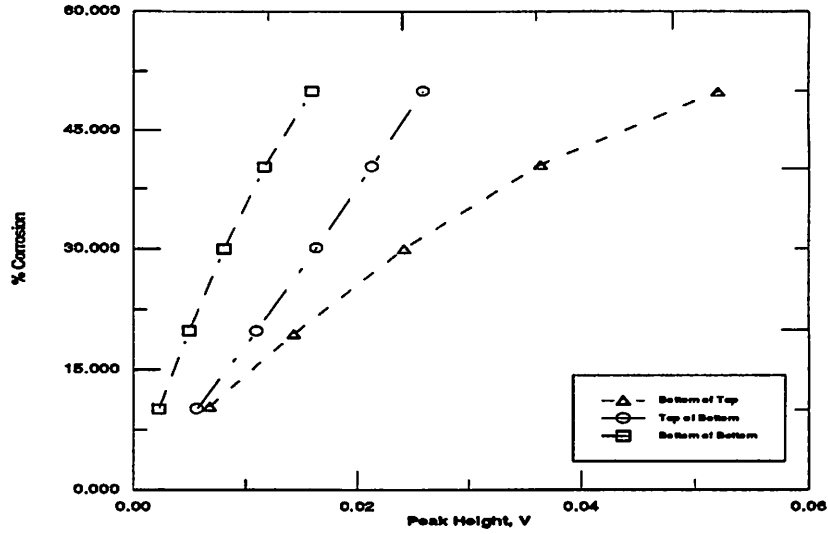


Figure 5.1: Calibration curves of the PEC instrument

percentage of the thickness of the layer). If we plot the percentage of corrosion as a function of the peak height, we get a set of calibration curves of the PEC instrument. The basic equation therefore is,

$$\%Corrosion = f(pH) \quad (5.1)$$

We get a family of calibration curves depending on the location of the corrosion or flaw. Typical calibration curves are as shown in Figure 5.1. These curves are least-squares fit of data from Figure 3.3.

Once the calibration curves for a particular specimen and probe are determined, we can quantify the signals we obtain from a scan. A calibration file (with an extension of .cal) is used to store all the calibration data. The file is structured and has a format as shown in Table 5.1.

The utility in having a calibration file, is that theoretically calculated calibration curves can also be written into this file, using any standard text editor, since it is

Table 5.1: File format of a calibration(.cal) file

Location	% Corrosion	Mean peak Height	Mean zerocross
Bottom of Top			
Top of Bottom			
Bottom of Bottom			

a simple ascii file. The user takes an initial scan on a calibration sample and an image in a non-calibration mode is generated. The probe is then moved to individual flaws on this sample and the user presses a calibrate button, to record the location, percentage corrosion, peak height and zero crossover at that point. The user needs to specify the first two parameters mentioned above. At a particular amount and location of corrosion, all calibrated values are averaged and a mean computed. When the user moves to a new spot, the average values are all stored in the file with the structure as shown.

Subsequently, if any image is to be displayed in a calibration mode, the user needs to specify the calibration file to be used. The data from the calibration file is read and calibration curves are fitted to the data (2nd degree polynomials) [10]. Once the curves are determined, the exact amount of corrosion corresponding to any signal obtained while scanning can be found from the signal amplitude. We presently are using only 5 colors to display calibrated images. Each color corresponds to a range of 5% corrosion, from 0 – 25%, i.e., the first color represents 0 – 5%, the second color represents 5 – 10%, and so forth.

5.2 Testing and Results

The PEC instrument has been tested on different kinds of samples in the laboratory. Specimens of actual aircraft skins and milled sheets of aluminium have been scanned to test the viability of the instrument. Some of the lab test results will be presented, and later the image of a scan on an actual working aircraft will be shown.

5.2.1 Laboratory testing

The very first sample on which measurements were made was the Boeing Panel VI specimen, which was available at the Center for NDE, Iowa State University. The panel is from an actual Boeing aircraft. The sample consists of a top plate of corroded aircraft skin (0.08" thick) fastened to a bottom plate of the same thickness but which has no corrosion. The scan taken by the PEC instrument is shown in Figure 5.2. The dimensions of the scanned area are 19 cm x 12 cm. Although the original image was a color image it has been converted into a gray scale image in this document. The color table is given below the image. Increasing amounts of corrosion are represented by the colors in the boxes from left to right. The image obtained by conducting an ultrasonic C-scan on the same specimen is also presented in Figure 5.3. The results are similar. The corroded areas are clearly visible. Other NDE techniques, including thermal wave imaging and X-rays also revealed corrosion in the same spots as detected by the pulsed eddy current instrument.

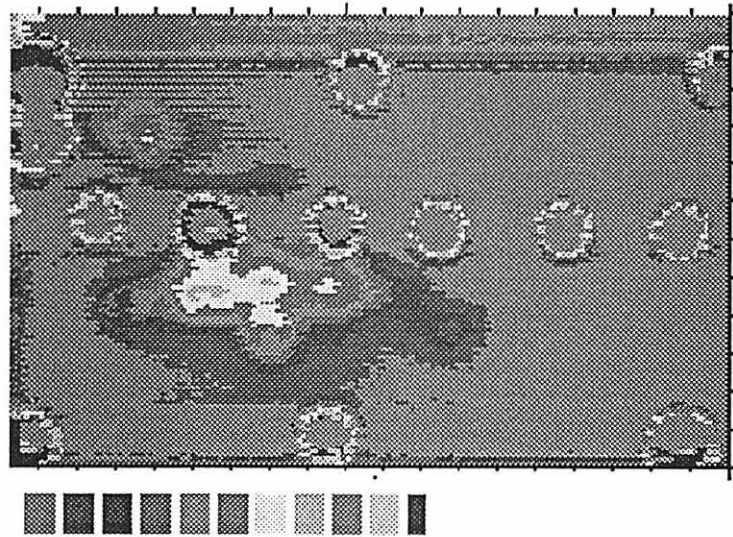


Figure 5.2: PEC Scan of the Boeing Panel VI

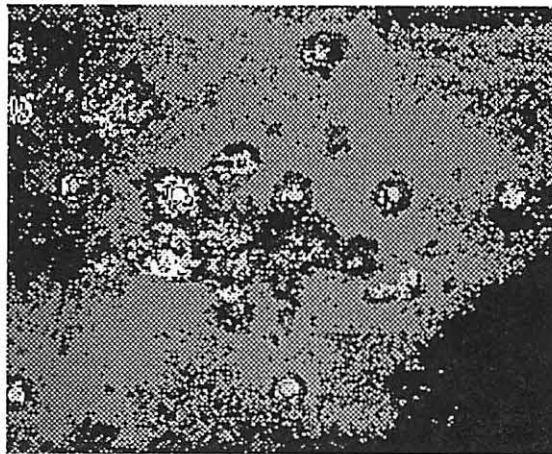


Figure 5.3: Ultrasonic C-Scan of the Boeing Panel VI

The comparison shown in the previous two images demonstrates the effectiveness of the PEC instrument in detecting corrosion. Let us now see another aspect of the instrument, the time-gating feature. By time-gating we are basically trying to isolate the signals due to corrosion. Performing time gating on the same sample gave us the result shown in Figure 5.4.

A closer look at the two images leads us to the conclusion that most of the signals due to the rivets have been eliminated. Not only that, the spots remaining are corroded areas which lie deeper in the specimen. Spots of corrosion which lie above appear to be stronger but time-gating tells us that they are closer to the surface. This discrimination is excellent when viewed in color. A converted gray scale image hampers clear distinction between the various gray scale levels. Time-gating, as has already been discussed, helps in distinguishing the various locations of the flaw. The signals due to air gap occurs when the probe moves over the rivets used in the lap joint of the specimen. Calibration goes one step further and can tell us the location and amount of metal loss.

CRACK DETECTION RESULTS: Preliminary experiments were conducted in the area of crack detection using the PEC instrument. Before looking at the results, we should note that the dimensions of the probe which was used to detect cracks are much larger than the length of the crack. The details of the crack sample is as follows:

- Type: PEC Scan of crack in second layer of riveted structure
- Layers: Two 65 mil plates riveted together
- Crack locations: The location of the cracks along with the dimensions in inches

are given just below the images. The images in Figure 5.5 also show the time-gating effect which totally removes the signals due to the rivets and shows only the signals due to the crack.

5.2.2 Field testing

The PEC instrument was taken to the Ames airport and tested on an Aero Commander airplane owned by Iowa State University. The particular plane which was scanned was a 5 seater commuter aircraft owned and operated by Iowa State University. The area which was scanned was the lower spar cap of the left wing. The results are shown in Figure 5.6.

There are two distinct regions in the image shown in Figure 5.6. The region above the row of rivets is the end plate of the spar and the region containing and below the rivets is the lower spar. There is a round patch just above the row of rivets. This is due to a hole which was present, and the patch next to the hole was a sticker on the wing of the plane. Apart from these two features there is no corrosion on the plane. Again, the converted gray scale image makes distinction a problem.

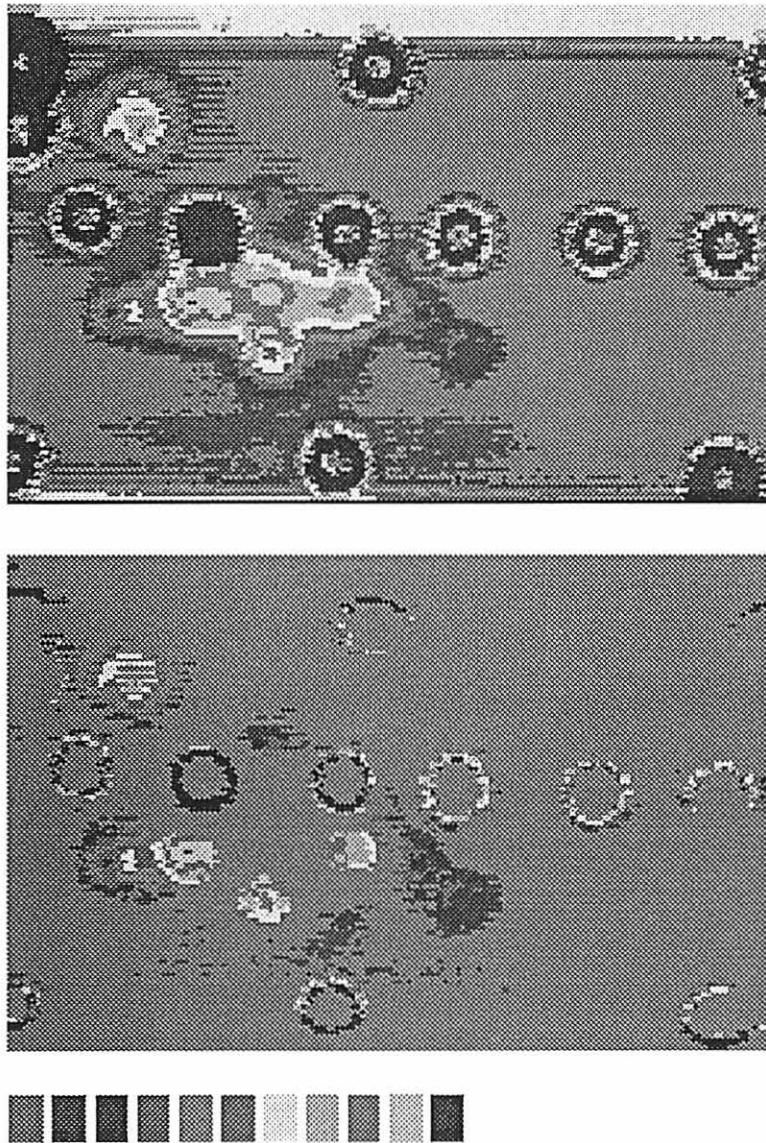


Figure 5.4: Images before and after timegating of the Boeing Panel VI specimen

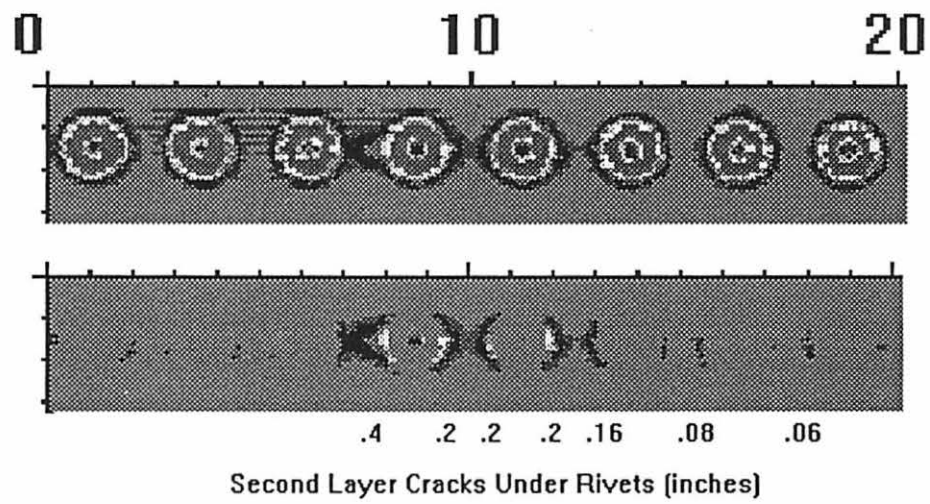


Figure 5.5: Crack detection using the PEC instrument

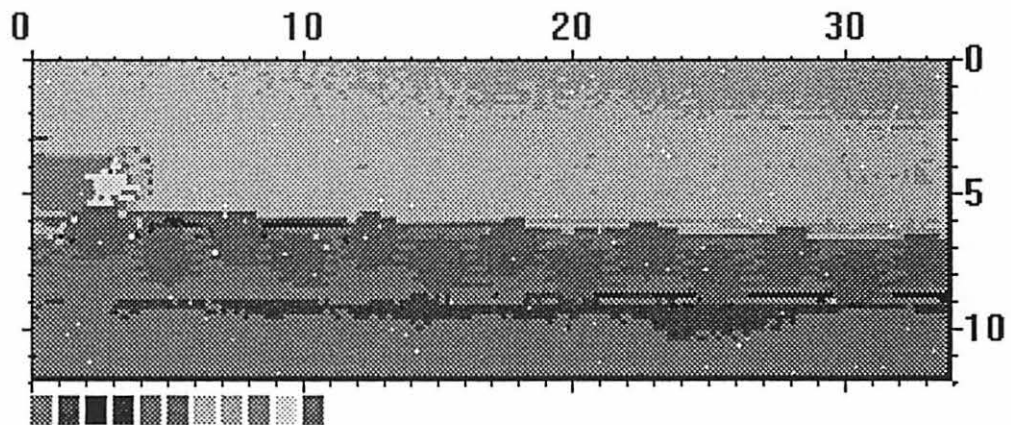


Figure 5.6: Scan of left wing, lower spar cap of a Aero Commander aircraft

CHAPTER 6. FUTURE WORK AND CONCLUSION

In the previous chapters we have discussed in detail the working of the PEC instrument. We have also presented results and comparisons to verify the correctness of the measurements. This is the first prototype of the scanned pulsed eddy current instrument. Our experience in using the prototype has identified some areas for future work, which would result in improved operation of the basic instrument. In coming months, we will continue to develop and test the instrument.

6.1 Future Work

6.1.1 Scanner and probe designs

- **Scanner Design:** The total scannable area in the existing scanner is around 34 cm x 12 cm. The bulky probe holder (5 inches x 3.5 inches) occupies too much space and reduces the scannable area in both the x and the y directions. Since the scanner design was borrowed from the dripless bubbler prototype, we had to build a fixture big enough to fit the probe holder. In reality the probe we use is much smaller (almost 2 times) in diameter than the probe holder. Hence, the scanner design could be modified.

- **Scan speed:** The software introduces an artificial delay at the end of each scan line to ensure that the two motors are in synchronism. This delay makes its presence felt at higher scan resolutions. An alternative mechanism to synchronize both motors would increase the scan speed. Another technique to increase scan speed is to take fewer data points and interpolate the pixels in-between these points.
- **Scanning in Frequency Domain:** So far, we have restricted ourselves to collecting time domain data during a scan. The frequency domain trace data can also be imaged, since it is readily available to us as a frequency domain admittance difference trace. Another exciting possibility is to scan a specimen and record all signals within a particular bandwidth. This will enable us to analyze flaws in the frequency domain.
- **Probe design:** The probe in the present prototype of the PEC instrument is large. For crack detection the probe has to be made smaller. The capability of the probe to penetrate deeper into the specimen also needs to be studied. At present the group is researching the use of ferrite core coils to achieve this goal. Another point worth mentioning is the thermal drift. To get the coil to attain thermal equilibrium during a scan, it may be necessary to use pitch-catch probes.

6.1.2 Crack detection

Preliminary results in detecting cracks have already been presented in the previous chapter. There is much interest in the aviation community on this use of the

instrument. Conventional eddy current scanning techniques for crack detection [16], record the change in the impedance, ΔZ , of the coil. This is measured by instruments in a swept frequency mode, which is time consuming. The PEC instrument, on the other hand, can make the same measurement within a matter of milliseconds. The only constraint being that we do not have a method of finding $\Delta Z(\omega)$ from $\Delta I(\omega)$. One of the methods which was looked into in great detail used the following formula,

$$\Delta Z(\omega) = \frac{1}{j\omega} * [(I_{flaw}(\omega))^{-1} - (I_{NULL}(\omega))^{-1}] \quad (6.1)$$

There is, however a problem associated with this method. After studying this problem it was found that the large time constant of the coil made this formula impossible to use to find $\Delta Z(\omega)$. Advanced filtering techniques like Kalman filtering need to be used in order to separate the coil response from the flaw response.

Another way of finding $\Delta Z(\omega)$ is by exciting the probe with a constant current instead of a constant voltage. In this case, the signal obtained with the instrument will be the voltage in the time domain (rather than the current), which can be Fourier transformed to a frequency domain impedance within the existing capabilities of the instrument.

6.1.3 Image processing

- **Image Processing Routines:** Presently, the image is created by assigning colors to individual pixels. There is no routine to enhance the quality of the image as a whole. Image smoothing filters can be used to get rid of the extraneous noise.
- **Image Compression:** The image stored in the bitmap format is in an uncom-

pressed form, thus requiring large amounts of memory to store. Compressed bitmaps can be formed by making certain changes to the conversion routine and pixel calculations.

- **Screen compatibility:** The pixel mapping and imaging is presently hardwired in the software to exactly fit a screen with resolution 640x480. This can be made compatible to any other screen size by making the mapping as a function of the window size returned by the built-in Windows function *OnSize()*. However, such a scheme will make computation a bit more complicated.

6.1.4 Neural networks for the inverse problem

The inverse eddy current problem with respect to corrosion deals with quantifying the amount of corrosion, knowing the peak height, peak time and the zero crossing time. We presently use calibration curves to be able to determine this. Instead, we could train a neural network to recognize the amount of corrosion and its location [17], given the above parameters for a particular kind of specimen. Some experiments were conducted with data from coatings using a regression neural network.

6.2 Conclusions

As stated in the introduction, the PEC instrument has transformed the theory behind pulsed eddy currents into a practical field instrument, one that is rugged and has so far matched the theoretical predictions of performance. The version presented in this thesis is the first prototype of the scanned system. The application of the PEC instrument is wide ranging. The scanning system could be an important inspection tool for the commuter airline industry enabling the user to inspect wings

from the outside. Conceivably, any eddy current inspection now carried out at a single frequency could be accomplished in the time domain with this instrument.

The development work on the PEC instrument is an ongoing process, and is in continuous progress even as this document is being prepared. Thrust areas in the future will be looking into ways of making the instrument a more generalized one. Crack detection is a capability that will be looked at in great detail while designing the next prototype of the PEC instrument.

BIBLIOGRAPHY

- [1] Moulder, J. C, Kubovich, M. W, Uzal, E., Rose, J. II, Pulsed Eddy-Current Measurements of Corrosion-Induced Metal loss : Theory and Experiment, *Review of Progress in Quantitative Nondestructive Evaluation*, Edited by D.O. Thompson & D.E. Chimenti, Vol. 14, Plenum Press, New York, pp. 2065 - 2072.
- [2] Rose, J. H, Uzal, E., Moulder, J. C, Pulsed eddy-current characterization of corrosion in aircraft lap-splices, *SPIE*, Vol. 2160, page 164.
- [3] Beissner, R. E, Fisher, J. L, A model of pulsed eddy current detection, *Review of Progress in Quantitative Nondestructive Evaluation*, Edited by D.O. Thompson & D.E. Chimenti, Vol. 5, Plenum Press, New York, page 189.
- [4] Fisher, J. L, Beissner, R. E, Pulsed eddy-current crack-characterization experiments, *Review of Progress in Quantitative Nondestructive Evaluation*, Edited by D.O. Thompson & D.E. Chimenti, Vol. 5, Plenum Press, New York, page 199.
- [5] Bowler, J. R, Harrison, D. J, Measurement and calculation of transient eddy-currents in layered structures, *Review of Progress in Quantitative Nondestructive Evaluation*, Edited by D.O. Thompson & D.E. Chimenti, Vol. 11, Plenum Press, New York, page 241.
- [6] Harrison, D. J. The detection of Corrosion in layered structures using transient eddy currents. *Nondestructive Testing of Materials*, Eds. R. Collins et al., IOS Press, London, 1995, pp. 115 - 124.
- [7] Halmshaw, R., *Non-destructive Testing* Edward Arnold Publishers Ltd. : London, 1987.
- [8] *6000 Series Software Reference Guide*, Compumotor Division, Parker Hannifin Corporation : Rohnert Park, California, April, 1995.

- [9] *Fast Scrics Libraries Manual*, Analogic Corporation : Wakefield, Massachusetts, 1992.
- [10] Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, T. V., *Numerical Recipes in C*. Cambridge University Press : Cambridge, Massachusetts 1989.
- [11] Papoulis, A., *The Fourier Integral and its Applications*, McGraw-Hill : New York, 1962.
- [12] Oppenheim, A. V & Schafer, R. W., *Discrete-Time Signal Processing*, Prentice Hall : Englewood Cliffs, New Jersey, 1992.
- [13] Gonzales, R. C & Woods, R. E, *Digital Image Processing*, Addison-Wesley : New York, 1992.
- [14] *Reference Volume I, Class Library Reference*, Microsoft Corporation : Redmond, Washington, 1993.
- [15] *Class Library User's Guide*, Microsoft Corporation : Redmond, Washington, 1993.
- [16] Auld, B. A., Jefferies, S. R, Moulder, J. C. Eddy-Current Signal Analysis and Inversion for Semielliptical Surface Cracks. *Journal of Nondestructive Evaluation*, Edited by D.O. Thompson & D.E. Chimenti, Vol. 7, Nos. 1/2, 1988, pp. 79 - 94.
- [17] Mann, J. M, Schmerr, L. W, Moulder, J. C, Kubovich M. W., Inversion of Uniform field Eddy Current Data Using Neural networks. *Review of progress in Quantitative Nondestructive Evaluation*, Edited by Thompson D. O, and Chimenti D. E, Vol. 9, Plenum Press, New York, pp. 681 - 688.

APPENDIX. INSTRUMENT SPECIFICATIONS

The Specifications for the PEC instrument are as follows:

PULSED EDDY CURRENT CARD

Waveform : 1 kHz square wave

Excitation Voltage : 0 - 10V DC (steps of 39 mV)

Maximum Current : 150 mA

Gain on Amplifiers : Variable(1 - 10,000)

Power Consumption : 2.0 watts maximum(obtained from PC supply)

PULSED EDDY CURRENT SCANNER

Probe Fixture : Fully articulating damped gimbal head

Articulation : Vertical and overhead dual curvatures ($R > 215cm$)

Max. Scan Speed : 17 inches per second

Scan Area : Approx. 0.5 sq. ft.

Mounting : Suction cups driven by compressed air(100psi)

Motor Control : Commercial AT6400 card in PC

Weight : 20 lbs

COMPUTER

Processor : 486, 66 MHz

RAM : 8 MB

Hard Drive : 540 MB

Display : 640 x 480 Active Matrix

DATA COLLECTION AND STORAGE

Image file(data) : Program specific ascii .img file

Image file(display) : Windows bitmap(DIB format)

Signal file : Ascii file format

A/D converter : 16 bit, 1 MHz sampling rate

A/D power consumption : 33 watts max., @ 5 volts

DISPLAY

Time gating : 0 - 512 microseconds

Amplitude gating : Any range within processor constraints

Display time base : 2 - 50 microseconds per division

Display frequency scale : 1.95 - 48.8 kHz per division