DOE/ID/13164--1(Vol. 3)
(DE95016313)

NATIONAL METAL CASTING RESEARCH INSTITUTE

Final Report

VOLUME 3: DEVELOPMENT OF AN AUTOMATED ULTRASONIC INSPECTION CELL FOR DETECTING SUBSURFACE DISCONTINUITIES IN CAST GRAY IRON

By
J. S. Burningham

August 1995

Work Performed Under Contract No. DE-FC07-92ID13164

For
U.S. Department of Energy
Office of Industrial Technologies
Washington, D.C.

By
University of Northern Iowa
Cedar Falls, Iowa

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (423) 576-8401.

Available to the public from the U.S. Department of Commerce, Technology Administration, National Technical Information Services, Springfield, VA 22161, (703) 487-4650.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DOE/ID/13164--1(Vol.3)
DE95016313
Distribution Category UC-1400

NATIONAL METAL CASTING RESEARCH INSTITUTE
Final Report

Volume 3: DEVELOPMENT OF AN AUTOMATED ULTRASONIC INSPECTION CELL FOR DETECTING SUBSURFACE DISCONTINUITIES IN CAST GRAY IRON

By
J. S. Burningham

August 1995

Work Performed Under Contract Number: DE-FC07-92ID13164

For
U.S. Department of Energy
Sponsored by the Office of the Assistant Secretary
for Energy Efficiency and Renewable Energy
Office of Industrial Technologies
Washington, D.C.

By
Metal Casting Center
Department of Industrial Technology
University of Northern Iowa
Cedar Falls, Iowa

ABSTRACT

An ultrasonic inspection cell was developed for the detection of subsurface discontinuities in cast gray iron parts as Task 3 (Robotic, Ultrasonic Inspection and Testing) of the DOE Cooperative Agreement (No. DE-FC07-92ID131-64) with the University of Northern Iowa Metal Casting Center. The cell consisted of an ultrasonic flaw detector (UFD), transducer, robot, immersion tank, computer, and software. Normal beam pulse-echo ultrasonic nondestructive testing, using the developed automated cell, was performed on 17 bosses on each rough casting.

Using test blocks and castings supplied by an industrial partner (John Deere Company) and working with a skilled ultrasonic inspector; ultrasonic transducer selection, initial inspection criteria, and UFD setup parameters were developed for the gray iron castings used in this study. The skilled ultrasonic inspector's operation of the UFD was noted for development of the cell software.

The ultrasonic inspection cell control software (UICCS) was designed and developed to perform the necessary functions for control of the robot and UFD in real-time. The UICCS performed two main tasks; emulating the manual operation of the UFD through the communication link with the unit, and evaluation of the ultrasonic signatures for detection of subsurface discontinuities.

The next phase of the cell development involved the testing of a random lot of 105 castings. These casting were processed through the automated inspection cell. The 100 castings which passed the inspection criteria were returned to the manufacturer for machining into finished parts where they were visibly inspected for defects after machining.

Five castings had one boss each which had ultrasonic signatures consistent with subsurface discontinuities. The five suspected bosses were manually inspected by the skilled ultrasonic inspector, with the manual inspection time recorded for comparison to the automated cycle time. The castings then were inspected using destructive testing techniques for detecting subsurface material voids.

The automated ultrasonic inspection cell was successful in quantifying the ultrasonic echo signatures for the existence of signature characteristics consistent with Go/NoGo criteria developed from simulated defects. The manual inspection showed that no defects in the areas inspected by the automated cell

avoided detection in the 100 castings machined into finished parts. Of the five bosses found to have subsurface discontinuities, two were verified by manual inspection after the rough casting surface was machined for the use of ultrasonic contact transducer inspection. The three remaining bosses showed no subsurface discontinuities after surface preparation for manual inspection. The developed automated ultrasonic inspection cell correctly classified 1782 of the 1785 bosses (99.832%) inspected.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
Chapter	
I INTRODUCTION	1
Background of the Problem	1
Significance of the Problem	3
Statement of the Problem	3
Limitations	3
Assumptions	3
II METHODOLOGY	5
Introduction	5
Overview	5
The Problem	6
Inspection Cell Description	7
Test Blocks	7
Ultrasonic Transducer Selection	8
Ultrasonic Signature	8
Ultrasonic Signature Evaluation Criteria	8
Signal Processing	10
Robot Programming and Interfacing	10
The Software	11
Manual Inspection of Suspected Castings	12
Understanding Cell Operation	12

III PRESENTATION OF RESULTS 15
Overall Results 15
Results of Good Castings 16
Results of Suspected Defective Castings . . . 16
Inspection Cycle Time 16
Projected Direct Labor Cost Savings 17
Process Problems 17
IV SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS . . . 19
Summary 19
Conclusions 20
Recommendations 20
REFERENCES 22
TABLES 23
FIGURES 27
APPENDIX A: UICCS PROGRAM ACTION DIAGRAM 40
APPENDIX B: CELL OPERATION FLOW CHART 70

LIST OF TABLES

Table — Page

1. Software Development Verification Dynamic Test Block Run 1 24

2. Software Development Verification Dynamic Test Block Run 5 25

3. Inspection Results of Castings Failing UICCS Inspection Criteria 26

LIST OF FIGURES

Figure		Page
1.	Ultrasonic A-Scan, test block C	28
2.	Ultrasonic A-Scan, test block F	28
3.	Ultrasonic A-Scan, peak go signals	29
4.	Ultrasonic A-Scan, AA-01	29
5.	Ultrasonic A-Scan, AF-13	30
6.	Ultrasonic A-Scan, DH-06	30
7.	Ultrasonic A-Scan, AZ(1)-12	31
8.	Ultrasonic A-Scan, AZ(2)-12	31
9.	Ultrasonic A-Scan, AZ(3)-12	32
10.	Ultrasonic A-Scan, BJ(1)-04	32
11.	Ultrasonic A-Scan, BJ(2)-04	33
12.	Ultrasonic A-Scan, BJ(3)-04	33
13.	Ultrasonic A-Scan, BS(1)-14	34
14.	Ultrasonic A-Scan, BS(2)-14	34
15.	Ultrasonic A-Scan, BS(3)-14	35
16.	Ultrasonic A-Scan, DK(1)-15	35
17.	Ultrasonic A-Scan, DK(2)-15	36
18.	Ultrasonic A-Scan, DK(3)-15	36
19.	Ultrasonic A-Scan, DX(1)-17	37
20.	Ultrasonic A-Scan, DX(2)-17	37
21.	Ultrasonic A-Scan, DX(3)-17	38
22.	Automated inspection cycle time for one part .	38
23.	Manual inspection time for one boss	39

CHAPTER I
INTRODUCTION

Task 3 of the DOE Cooperative Agreement (No. DE-FC07-92ID13164) is titled Robotic, Ultrasonic Inspection and Testing. This report (Subtask 3.3) details the research conducted as Subtasks 3.1 and 3.2. as identified in the proposal.

Subtask 3.1: *A robotic system will be developed to manipulate an ultrasonic probe to detect sub-surface defects in cast parts. The system will consist of a robotic unit and ultrasonic test equipment donated by the John Deere Company.*

Subtask 3.2: *The robotic test system will be automated to collect data from random sampling of heads from the John Deere 8000 Foundry. The defects will be evaluated and cataloged for identification and analysis.*

Background of the Problem

Since the pouring of the first castings, discontinuities have been a problem. Discontinuities are irregularities, breaks, or gaps in the material structure. Most of the different types of casting discontinuities are visible to the naked eye, caused by variables in the casting process. Some casting discontinuities are not detectable by visual inspection because the defect is below the surface of the material. These subsurface discontinuities must be detected and identified before remedies to resolve the problems can be applied or value added work is performed on the casting that will later need to be rejected because of the defect.

Until the development and application of X-ray and ultrasonic inspection technology, subsurface discontinuities were not detectable until after value added processes were performed on the casting, or worst yet by the failure of a casting product in testing, or while in service. Today it is common practice, and many times required, for castings and other manufactured products to be 100% inspected, especially in the aerospace and nuclear industries. In castings for industries other than aerospace and nuclear, subsurface inspection is limited because of cost.

Every foundry would prefer to have a reputation of producing zero defects, but this reality is often far from ideal. The inspection process is but one step in the total quality assurance programs of most manufacturers. Manufacturers want to detect discontinuities early in the manufacturing process. If the defect is unrepairable or the rework costs are excessive, the part will be scrapped.

In foundries, the defective castings will be scrapped for remelt and recast, saving the investment in raw material. Scrapping defective parts costs money, not only for the material involved, but also for the value added processing that takes place prior to the detection of the defect. Early detection of flaws and defects in a manufactured part reduces the value added processing cost lost because of discontinuities.

Inspection processes for detecting subsurface casting discontinuities are costly and labor intensive, adding to the cost of the final product. Quality assurance programs, as applied in many industries today, will often only statistically sample a production lot, passing or rejecting the lot on the result of inspecting only a few. As the cost of scrapping a casting goes up, there is a need for more thorough inspection to detect discontinuities before the value added operations have been performed via the manufacturing process.

After the foundry has delivered the casting to the customer and a defective casting is detected during the customer's manufacturing processes, foundries making the casting normally are required to replace the defective casting. Contractional agreements between the foundry and the customer also may involve a number of compliance parameters that cause financial burden to be placed on the vendor (the foundry). Manufacturing of raw materials and value added processes by companies usually requires the vendor to meet certain minimum standards, SAE, ASTM, ISO, etc. In a global economy, as a manufacturer for the year 2000 and beyond, preferred vendors will need to become ISO 9000 certified to maintain a market share of produced goods. A foundry's business relationship with a customer can be influenced by the quality of the castings delivered in both a negative and positive manner.

When a company has a captive foundry, they absorb all the costs associated with the defective casting. When foundries bid on jobs, they add the cost of scrap into the bid. Foundries with lower scrap rates can bid lower prices while still maintaining the necessary margin of profit, thus underbidding competitors and becoming more competitive in the marketplace.

This project was designed to investigate existing technology and develop a prototype automated ultrasonic inspection cell for detecting subsurface discontinuities in a cast iron part. The cell needs to control the ultrasonic nondestructive evaluation (NDE) equipment, robot, analyze collected data, decide about the quality of the casting, and save inspection data for future analysis.

Significance of the Problem

The early detection of casting discontinuities is important to the foundry industry allowing a reduction in scrap costs and helps to achieve 100% quality of the product in every delivery. A cost effective, advanced technology NDE system is needed to achieve quality assurance goals that will enable the American foundry industry to remain competitive in the national and international markets.

Statement of the Problem

The problem of this study is to develop a prototype automated inspection cell for the detection of subsurface casting discontinuities while holding the investment of time and labor to a minimum. This involves interfacing existing technologies in ultrasonic inspection, robotics, and computers; developing inspection criteria and standards; producing software for emulating the necessary operator skills, decision making capacity, and cell supervisory control.

Limitations

This research was funded in part by a grant from the John Deere 8000 Foundry and the U.S. Department of Energy. The iron casting used in this study was selected by the foundry, based on their identification of need to detect subsurface discontinuities. The casting to be analyzed in this study has 17 specific locations where subsurface discontinuities have a history of occurring.

Assumptions

For developing and calibrating the inspection system, simulated flaws are necessary. Flat bottom drilled holes at varying depths in sample castings will be used. These flat bottom holes have been shown to represent the type of echo condition that discontinuities of similar characteristics would present to ultrasonic inspection. The equipment in the ultrasonic cell identified for this study is representative in accuracy and capabilities to those commonly used in industry.

CHAPTER II
METHODOLOGY

Introduction

The purpose of this project was to test the feasibility of automated testing of cast iron to enhance the efficiency and, perhaps, the effectiveness of manual methods of quality control in a production setting. The work was done in conjunction with the John Deere 8000 Foundry, at the University of Northern Iowa's Department of Industrial Technology Metal Casting Center. An overview of the work is provided below and details of the methodology follows.

Overview

This project consisted of two tasks--inspection cell design (including software development and integration with the inspection cell), and testing of a random sampling of actual castings, and follow-up of the tested castings. A general discussion of each of the steps in each of the two tasks follows.

The first task involved the design of the apparatus (inspection cell) necessary for the automated testing which was to be carried out using ultrasonic inspection of actual castings. The specific make-up of the inspection cell had to be determined and components selected to: perform the ultrasonic A-Scan and collect the echo signatures, automatically position the transducer at the various points to be inspected, and integrate all the testing activities.

Once general decisions about the inspection cell were made, it was possible to begin design of the software which would analyze the echo signatures and indicate whether the signature suggested the existence of subsurface discontinuities in the regions of the castings that were to be tested. Development of the software involved working with a skilled ultrasonic inspector from the industrial partner to understand the methods and procedures for inspecting the specific casting using ultrasonic equipment; this knowledge was emulated in the control software. This process had several steps: initial design of the software, an interactive process of scanning test blocks (of known quality) supplied by the industrial partner and revising the software until satisfactory assessments of the test blocks were achieved, and integration of the testing software with the automatic positioning equipment of the inspection cell.

The next phase of the cell development involved the testing of a random sampling of 105 castings. These casting were processed through the inspection cell. The castings passing the developed inspection criteria were returned to the manufacturer for machining into finished parts where they were visibly inspected after machining for defects. The castings found to have ultrasonic signatures consistent with subsurface discontinuities were manually inspected by the skilled ultrasonic inspector, with the manual inspection time recorded for comparison to the automated cycle time. The castings then were inspected using destructive testing techniques for detecting subsurface material voids.

The Problem

The John Deere 8000 Foundry, the industrial partner in this research, identified a problem of defects, subsurface shrinkage cavities (one type of subsurface discontinuity), near the top of 17 bosses in a specific iron casting. "A shrinkage cavity is a jagged hole or spongy area lined with fernlike crystals called dendrites" (American Foundrymen's Society, 1966, p. 111). The causes of shrinkage cavities include abrupt changes in section size (American Foundrymen's Society, 1972), typical of the 17 identified problem locations. Hénon, Mascré, and Blanc (1971/1974) identify net expansion in cast iron as one of the most frequent causes:

> The expansion which takes place within the solidified surface areas of the casting causes displacement of the liquid from the central region, creating a void. This void is not filled when the residual liquid solidifies because feeding is impaired by a dense network of dendritic crystals. (p. 107)

Because of the resources necessary to perform 100% manual ultrasonic inspection of the problem areas, a less expensive approach is necessary to detect the defects to reduce scrap costs associated with the additional work that is performed on the castings before finding the defects in later manufacturing processes. The industrial partner in this research has specified that the inspection process is to take place prior to any machining of the casting. The castings used in this study to develop and test the inspection cell were supplied by the foundry in the typical condition that exists on the production line at the required specified stage in the manufacturing process.

Inspection Cell Description

The automated ultrasonic inspection cell consisted of an immersion tank, Panametrics EPOCH 2002 digital ultrasonic flaw detector (UFD), Panametrics 5.0 Mhz V309-SU ultrasonic transducer in a normal beam pulse-echo arrangement, Hitachi M5030 robot, and a 80386 CPU based microcomputer. The immersion tank was fitted with a part holding fixture, supplied by the foundry, for locating the part while under inspection. The parts were manually loaded and unloaded for testing and evaluation purposes.

The Panametrics EPOCH 2002 digital UFD was used to transmit and receive the ultrasonic signals, perform the analog-to-digital conversion of the signal echo of the A-Scan from the transducer, and average multiple A-Scan signatures together. The UFD has an optional RS-232 communication port, running at 19.2 kilobaud for full command and communication capability with the cell computer. This is the same type of UFD typically used for manual inspection, only with the addition of a communication interface.

The computer program to perform the necessary zeroing procedures on the UFD was developed in conjunction with the skilled ultrasonic inspector. This involved the observation of UFD setup and zeroing by the inspector, as well as emulating the process and decision logic with the developed software.

The Ultrasonic Inspection Cell Control Software (UICCS) performs the zeroing routine to adjust the UFD for variations in casting height, which required taking an initial reading to determine the transducer distance to the part surface, adjusting the signal peaking the echo signature of the part surface, and adjusting the zero offset of the UFD to place the part surface at the zero reference of the flaw detector display. In manual operation, the inspector adjusted the UFD by viewing the echo signature on the display and adjusting front panel controls.

Test Blocks

A set of nine test blocks, supplied by the foundry and machined from a sample casting, was used for evaluation and development of the system. Seven test blocks had 0.089 inch flat bottom holes drilled from the back side at varying distances from the part entrance surface, one hole in each block, representative of the location and minimum size of defects to be detected.

Ultrasonic Transducer Selection

Working with a skilled ultrasonic inspector, a series of tests were run using 2.25 Mhz, 3.5 Mhz, and 5.0 Mhz transducers. The inspector calibrated the UFD according to standard calibration procedures. All three transducers produced acceptable results for the inspector to locate and identify the simulated defects in the test blocks. For computer analysis of the ultrasonic echo signature, the 5.0 Mhz transducer was selected because it produced the signature with the maximum differentiation between the relative echo signal amplitude of the simulated defects and the echo noise in the surrounding part.

The Panametrics V309-SU (SN:124007) unfocused 5.0 Mhz immersion transducer that was selected for use in the cell has a nominal element size of 0.50 in. The transducer specifications and technical notes (Panametrics, 1991) calculate the near field far limit at 5.287 inches using a water coupler. "The minimum and maximum practical focal lengths have been determined by considering the acoustic and mechanical limitations" (p. 32). For the 5.0 Mhz transducer using a water coupler, the minimum practical focal length is specified at 0.75 inches, and the maximum at 4.20 inches A transducer to part distance of one inch was used for programming the transducer placement. This allowed for minor part height variations in the holding fixture without violating the minimum practical focal length.

Ultrasonic Signature

The ultrasonic inspection data collected from each inspection location consisted of 200 digitized data points, representing the ultrasonic signature of the location under inspection, for a depth of 1.0 inch Each digitized data point represents 0.005 inches of material thickness. This signature is called an A-Scan. "The A-Scan plots reflection amplitude versus time" (Wolters, 1980, p. 35).

Ultrasonic Signature Evaluation Criteria

The development of the ultrasonic signature evaluation criteria was based upon the problem areas in the casting identified by the foundry. They specified that shrinkage cavities were known to occur near the surface of the 17 bosses on the part. The part bosses were designed so the top 0.150 inches are machined off in the manufacturing process. The foundry identified that the defects can fall in the top 0.750 inches of the boss area after machining and have a larger concentration near the surface. The ultrasonic signature evaluation criteria were developed from test blocks having simulated defects of varying depths.

The parameters for evaluating the ultrasonic signature were developed using the echo signatures from the test blocks. Working with a skilled ultrasonic inspector, UFD inspection settings were developed for inspecting the bosses. This involved taking a series of A-Scans of the test blocks, interpreting the data, and constructing the acceptance/rejection criteria. Sample signatures were collected from test blocks A-G (Figures 1 and 2 typify the set collected).

The developed criteria were a series of data point values, representing the minimum peak relative signal levels for part rejection. The developed parameters were used to evaluate each inspection signature for a Pass/Fail or Go/NoGo decision. Echo signatures that pass the inspection criteria were defined not to have a defect; echo signatures that fail the criteria were classified as having suspected defects.

Initial testing and development was performed in a static setup where the transducer was fixed above the test block under inspection. The test block runs for verifying the software and finding the error rates were performed in a dynamic setup where the robot was programmed to move the transducer into position for each A-Scan. It was found that the robot induced a vibration into the dynamic setup that resulted in very high levels of signal noise and unstable images. This problem was very apparent in that A-Scans of the test blocks void of defects had noise levels sufficient to violate the Go/NoGo parameters in 48% of the cases in the initial dynamic test run. The total error rate for the test blocks with simulated defects in the initial dynamic test run was 1.14% (see Table 1).

The solution to the problem involved four basic modifications to the cell operation and software. First, the robot's approach speed to the inspection point was decreased. This reduced the vibrations injected into the system by the robot. Second, a programmed delay between the robot arriving at the inspection point and the start of the A-Scan was added. This delay dampened the robotic induced vibrations. Third, the number of A-Scans averaged together for each signature was increased to four from an initial value of three. This digital signal processing further helped in filtering out noise, both internal to the system and externally induced. Finally, the test procedure was changed to repeat any A-Scan that did not pass the inspection criteria. This test procedure modification helped in two ways--it allowed a minimum programmed delay before the start of the A-Scan, in keeping with the need for a minimum cycle time, and reduced random noise interference. After these modifications, the fifth dynamic test block run produced no errors in properly classifying the nine test blocks (see Table 2).

After the dynamic test block runs and revisions to the software, two castings, later serialized as AA and AB, were tested in the integrated ultrasonic inspection cell. This testing involved verifying cell operation, both hardware and software, determining cell cycle time, and verifying classification error rates on 34 additional bosses. Both castings were inspected 25 times, with each repetition inspecting 17 bosses, for a total of 850 inspection points. Both castings where found to be void of subsurface discontinuities. There were no classification errors during the test repetitions, but communication problems with the UFD were encountered that caused the system to halt the inspection cycle. The cause of the communication problem was isolated to the internal software of the UFD. The only method of reestablishing the communication link was to manually power the UFD off and back on. The UICCS was modified to detect the problem and notify the operators of the situation, which required human intervention to correct. This communication problem occurred three times during later cell testing, requiring aborting an inspection cycle and starting the part inspection over.

Signal Processing

Wolters (1980) showed that the signal processing technique of averaging A-Scans resulted in reduced echo noise in the resultant signature. As noise is an anticipated problem in cast iron from a review of the literature and preliminary testing, this signal processing technique was applied to all A-Scans internally within the UFD under software command. Initially, three A-Scans were averaged together to process out noise; later, in dynamic testing of the system, the number was increased to four.

Robot Programming and Interfacing

The Hitachi M5030 is a light duty electric 5-axis articulated-arm robot. The robot was programmed by way of a teach pendent to move along a programmed path, stopping at the 17 inspection points with the transducer positioned 1.0 in. above the inspection point and perpendicular to the surface of the part.

The robot was interfaced to the cell computer via digital I/O lines. The cell computer used an Industrial Computer Source DIO8-P optically isolated digital I/O interface for communicating with the robot. The interface was selected for the optical isolation provided between the cell computer and the robot; this allowed for safe and easy interfacing of the different signal levels used by the hardware.

The UICCS instructs the robot to select and execute a preprogrammed set of instructions. The robot sends a digital output signal to the cell computer indicating that the robot is at a predefined location (inspection point) awaiting a digital input signal from the cell computer before continuing execution of its program.

The robot was fitted with end-of-arm tooling for holding the ultrasonic transducer below the water line of the immersion tank. The end-of-arm tooling was designed to break away from the robot arm if a collision occurred.

The Software

The UICCS was written and compiled in Microsoft's QuickBasic V4.5, operating under Microsoft's MS-DOS V5.0 operating system. An action diagram, a program diagramming technique described by Martin and McClure (1985), of the program is in Appendix A. The UICCS handles the communications with the UFD and robot, analyzes ultrasonic echo signatures, interfaces with the cell operator, displays A-Scan data, and produces printed inspection reports.

The software for analyzing the ultrasonic signature was developed using nine test blocks, seven of which had flat bottom holes at varying depths. Two which were void of defects were used in the development and calibration of the cell hardware and software.

The software development goal, as specified by the industrial partner, was to have less than a 5% error in correctly classifying test blocks with simulated defects, and 1% error in properly classifying test blocks void of defects. For calculating classification error rates, each test block was inspected 100 times. The software development cycle involved analyzing the signatures of erroneously classified test blocks and developing solutions to achieve development goals.

Manual Inspection of Suspected Castings

The evaluation phase involved the testing of 105 production castings. The castings were serialized and identified as AA through EA. Production castings evaluated as passing were machined into finished products with any discovered defects in the inspected locations reported. Production castings failing the developed inspection criteria were manually inspected using contact ultrasonic inspection by a skilled inspector, and then inspected using destructive technique.

Understanding Cell Operation

Understanding how the automated ultrasonic inspection cell operates is best achieved by following an example part through the system. (A flow chart of the cell operation can be found in Appendix B.) When the part is loaded onto the holding fixture, the cell is ready to inspect the part.

The UICCS requires the operator to input the part serial number. This information is used to match the collected data with the individual part. The UICCS first instructs the robot to select a stored set of instructions that were previously programmed into the robot via a teach pendent. The UICCS then instructs the robot to start execution of the selected instruction set, causing the robot to move the transducer that is mounted on the robot arm to the first preprogrammed inspection location. While the robot is moving to the inspection location, the UICCS commands the UFD to recall a set of initial parameters that are stored in the unit's memory. These parameters control the operation of the interface between the UFD and the transducer. The UICCS then waits for a signal from the robot indicating arrival at an inspection point. Upon the robot's signal of arrival, the cell computer delays for one second to dampen the robot's vibrations that could interfere with obtaining a reliable A-Scan.

The UFD requires the operator, when using the UFD for manual ultrasonic inspection in an immersion tank, to make a series of adjustments to the unit using the UFD display to view the ultrasonic signature and UFD keypad for entering parameter adjustments. The UICCS must duplicate these operator's skills and decision making ability to perform the same setup tasks through the communication interface. The setup tasks are adaptive in nature, the software must make adjustments to external equipment based upon sensorial input.

The first adaptive control task of the UICCS is to peak the part surface echo's relative signal level. This task is required because of casting material variations in material thickness and surface condition causing the distance between the ultrasonic transducer and part surface to vary.

The task starts with the UICCS commanding the UFD to take an A-Scan; all A-Scans are programmed to be the results of four time-sequential A-Scans averaged together, digitally processing out most of the signal noise. The analog A-Scan signature is converted to a digital representation comprised of 200 data points within the UFD, with each data point containing a relative signal amplitude between 0 and 63 along a time interval calibrated to represent a distance of 0.005 in., making the data set represent a depth of 0.995 in. The UFD acknowledges successful completion of the A-Scan averaging to the UICCS. The UICCS then commands the UFD to upload the A-Scan signature data set.

The UICCS needs to identify the part surface of the casting in order to adjust the zero offset. The part surface is the peak echo signal in the A-Scan signature data set, but at low relative amplitude signal levels, resolution of the part surface from the data set is not possible, so the relative amplitude signal level must be increase to determine the relative part surface location within the data set.

If the peak echo signal, representing the part surface, is below the maximum relative amplitude of the data set the UICCS calculates the needed signal level increase necessary for the peak echo signal to approach the maximum relative amplitude. This signal level change is downloaded to the UFD, along with another request for an A-Scan. The new A-Scan is then uploaded to the UICCS. This process is repeated until the peak echo signal from the part surface is at the maximum relative amplitude.

The second adaptive control task of the UICCS is to adjust the UFD's zero offset to place the part surface echo at a depth of zero in the A-Scan signature data set. The UICCS calculates the needed zero offset for the UFD so that the part surface approaches the zero depth position in the A-Scan signature data set. Due to signal impedance variations within the casting and between different casting, the ranging capability of ultrasonics is not exact, but only an approximation; these impedance variations cause the speed of the signal to vary. The ranging error is reduced as the distance measured decreases, this necessitates the adaptive control to make adjustments that approach the desired results, repeating until the solution is achieved. The UICCS downloads to the UFD the new zero offset value, requests an A-Scan, and uploads the A-Scan signature data set. This process is repeated until the part surface is at the zero depth position in the A-Scan signature data set.

Upon successful completion of the two adaptive control tasks, the UFD is ready to inspect the boss. The UICCS sets the inspection signal level (67 dB) in the UFD for the inspection A-Scan, then commanding an A-Scan and the uploading of the A-Scan signature data set. The uploaded A-Scan signature data set is compared to the Go/NoGo criteria. The A-Scan passes the Go/NoGo criteria if all the data points relative amplitudes fall below the rejection criteria. If the A-Scan fails the Go/NoGo criteria, the A-Scan is discarded and the inspection point is reinspected; this reinspection is to reduce misclassifications caused by internal and external noise. The second A-Scan is used to determine if the inspection point passes or fails. The last A-Scan of an inspection point is saved to a data file.

The UICCS then instructs the robot to continue executing its instruction set, causing motion to the next inspection location or after the last location returning to a home position. The UICCS repeats the sequence of events for each inspection location. A part passing all inspection criteria for each inspection point is classified as a good casting; failure of any inspection criteria will classify the part as having a possible defect. If a part is found having a possible defect, the whole part is reinspected two additional times.

CHAPTER III

PRESENTATION OF RESULTS

Overall Results

The testing of 105 castings involved the ultrasonic inspection of 1785 bosses. Five bosses failed the inspection criteria, one each on five different castings. The remaining 1780 bosses had no ultrasonic signatures consistent with subsurface discontinuities. The 100 castings that had all 17 bosses passing the inspection criteria were returned to the manufacturer for machining into finished products. The manufacturer reported they found no shrinkage cavities in the inspected areas during the manufacturing or final inspection process.

Of the five castings, each with a boss failing the inspection criteria, AZ, BJ, and BS failed each of the three test repetitions. Castings DK and DX both failed only two of the three test repetitions. All five bosses were manually inspected by the foundry's ultrasonic NDE inspector using contact transducer procedures. This required that the rough casting surfaces be machined flat for good contact transducer coupling. After machining of the rough cast surface, the inspector could not identify any subsurface discontinuities in castings BJ, DK, or DX. Ultrasonic echo signatures consistent with the depth location from the automated ultrasonic A-Scans were identified by the inspector in castings AZ and BS. The automated inspection erroneously classified 3 of the 1785 bosses inspected. The UFD used by the inspector was not capable of producing either hardcopy or data file output. Table 3 summarizes the test results of the five castings failing the UICCS inspection criteria for both the UICCS analysis of the signature and the manual inspection of castings.

Destructive testing for subsurface shrinkage cavities in the five suspect castings was performed by the foundry. No subsurface shrinkage cavities (one type of subsurface discontinuity) were reported in the five suspected bosses. The destructive testing involved the machining of successive layers of material, visually inspecting each layer for shrinkage cavities breaking through the machined surface. This destructive testing was only capable of finding subsurface shrinkage cavities and not qualifying other subsurface discontinuities that can produce echoes.

Results of Good Castings

The 100 castings determined to be void of subsurface discontinuities in the inspected regions all produced A-Scans that fell within the acceptance criteria for a good part. Figure 3 shows the peak relative signal amplitude of all A-Scans that met the acceptance criteria shown by the Go/NoGo line. The Go/NoGo is displayed on all A-Scans of reference. The A-Scan of AA-01, the first boss of casting serial number AA and typical of the A-Scans passing the inspection criteria, is shown in Figure 4. Additional typical A-Scans of bosses passing the inspection criteria are shown in Figures 5-6.

Results of Suspected Defective Castings

For each casting having suspected defects, there are three A-Scans of the suspected bosses. Bosses AZ-12, BJ-04, and BS-14 were identified as failing the Go/NoGo demarcation in each of the three data sets. It should be noted that the UICCS required two sequential failures to flag the boss as failing. This repeat failing was without the repositioning of the robot. Upon failing in the first set, the operator reinspected the complete part two additional times.

Part serial number AZ, boss 12 (AZ-12) shows an echo at about the 0.175 inch. depth in all three A-Scans failing the acceptance criteria. This was verified by manual inspection (see Figures 7-9). Boss BJ-04 shows an echo violating the acceptance criteria at about the 0.150 inch depth. This was not verified by manual inspection (see Figures 10-12). Boss BS-14 shows in all three A-Scans an acceptance criteria violation at the 0.50 inch depth. This was also verified by manual inspection (see Figures 13-15). Boss DK-15 shows a strong echo at the 0.20 inch depth, but only violating the inspection criteria in two of the three scans (see Figures 16-18). Boss DX-17 shows a strong echo near the 0.15 inch depth, violating the inspection criteria in only two of the three scans (see Figures 19-21).

Inspection Cycle Time

Inspection cycle time was an important UICCS design consideration. The cycle time data was processed using SPSS/PC+ 4.0 (1990). The mean cycle time for automatic inspection of a casting (17 bosses) was 3.242 min ($\underline{N}$ = 50) with a standard deviation of 0.254 measured during the test run repetitions on casting AA and AB. The cycle time data was positively skewed (Skewness = 1.404). Figure 22 is a histogram of the inspection cycle time. The histograms were produced by the Graphic routine in SPSS/PC+ 4.0 (1990).

A large segment of the measured cycle time was comprised of communications with the UFD and waiting for the UFD to complete the A-Scan task. A minimum of five A-Scan data sets were required for instrumentation zeroing and inspection for each boss. It took 1.2 s for the UFD to receive an A-Scan request, take four A-Scans, average them together, and notify the UICCS it was ready to upload the resultant data set. The A-Scan data set consisted of a string of 613 bytes, at 19.2 kilobaud. This required 0.32 s per A-Scan upload. A minimum of 85 A-Scan data sets needed to be uploaded from the UFD for each part. This calculates to a minimum inspection time of 129.14 s for each casting not including robotic motion. The cycle time did not include casting loading nor unloading time. In a production environment this would typically be performed by automated material equipment.

The skilled ultrasonic NDE inspector's mean cycle time for inspecting each boss was 5.760 min ($\underline{N}$ = 5) with a standard deviation of 1.118 and negatively skewed (Skewness = -0.635) (see Figure 23). This cycle time included surface preparation, but not instrumentation setup time. This calculates to 97.92 min for manual inspection for 17 bosses (one casting).

Projected Direct Labor Cost Savings

Compared to the automatic inspection, manual inspection is 30 times more time consuming. Using the industrial partner's direct labor rate of $27.37 ($22.25 per hour labor plus 23% benefits) and the mean cycle times, the direct labor costs for manual ultrasonic inspection of one casting is $44.67. The direct labor costs for the automated ultrasonic inspection cell to inspect one casting is $1.48. Based upon the foundry's production of 100 castings per day, the projected direct labor cost savings is $4,319 per day. The manpower requirements are also a consideration, the automated inspection cell would require 5.4 man-hours per day to process 100 castings, the manual inspection method would require 163.2 man-hours per day.

Process Problems

One of the problems discovered in processing the sample lot was that the immersion technique caused an oxidation of the castings. This oxidation problem required an additional cleaning process, which would not be acceptable in a production environment. While a rust perverter was used in the testing, it did not perform as required. Further investigation into different rust prevention agents may help in eliminating the problem.

Further tests were run using squirter (or bubbler) technology, where a flowing stream of couplant replaced the immersion tank. This technology eliminates most of the oxidation problem encountered with the immersion method. Further investigation into alternate coupling methods is warranted at this time.

CHAPTER IV
SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Summary

The thrust of this research was to learn if a computer-controlled ultrasonic inspection cell could accurately detect subsurface casting discontinuities in cast iron and increase the efficiency of the inspection process. The developed cell used a normal beam pulse-echo transducer arrangement in an immersion tank, generating an ultrasonic energy beam which entered the boss perpendicularly to the part surface. Upon encountering a material discontinuity, part of the ultrasonic energy packet was reflected back in the direction of the ultrasonic source. Only that portion of the ultrasonic energy packet received by the transducer and converted into electrical energy was converted into an ultrasonic signature data set by the UFD and transmitted to the cell control computer for analysis by the UICCS.

The UICCS quantitatively analyzes the signature data set to decide if any data byte violated developed Go/NoGo criteria. A violation of the Go/NoGo criteria identifies a condition with the casting that reflects ultrasonic energy in excess of predetermined acceptance criteria.

The automated ultrasonic inspection cell was successful in quantifying the ultrasonic echo signatures for the existence of signature characteristics consistent with Go/NoGo criteria developed from simulated defects. The manual inspection showed that no defects in the areas inspected by the automated cell avoided detection in the 100 castings machined into finished parts. Of the five bosses found to have subsurface discontinuities, two were verified by manual inspection after the rough casting surface was machined for the use of ultrasonic contact transducer inspection. The three remaining bosses showed no subsurface discontinuities after surface preparation for manual inspection. The developed automated ultrasonic inspection cell correctly classified 1782 of the 1785 bosses (99.832%) inspected.

The automated inspection cycle time was an average of 30 times faster compared to the manual inspection of the suspected bosses. In a production situation where 100% manual inspection was required, the manual inspection cycle time could be reduced by the use of semi-automated or automated equipment for the surface preparation necessary for manual ultrasonic inspection.

Conclusions

The developed computer-controlled ultrasonic inspection cell is the interfacing of existing hardware technology, coupled with an expert system control program that emulates the necessary skills of a human inspector to perform an inspection of a specific cast iron part in an expeditious manner with the minimum of operator interaction. The system is a tool, identifying areas for further investigation by a skilled inspector. It is an inspection tool that can perform 100% inspection in a timely and cost efficient manner, passing parts found void of possible defects, and identifying those castings that have an ultrasonic signature consistent with the type of flaws that a foundry wants to detect. The developed system is quantitative in design and ability. The UICCS makes a simple Go/NoGo decision based upon the relative signal amplitude of ultrasonic echoes caused by subsurface discontinuities and acceptance criteria.

The casting surface condition caused false echoes in three of the five suspected bosses, evident by the fact that the automatically detected subsurface echoes disappeared after the part surface was machined for manual inspection. The false echoes were near the top of the boss inspection area.

The destructive testing of the suspected bosses did not locate any subsurface shrinkage cavities, this was a qualitative test for detecting material voids, as opposed to the quantitative inspection for subsurface discontinuities by both the automated and manual ultrasonic inspection.

Artificial intelligence, manufacturing intelligence, adaptive control, and soft automation are all part of the technological advances that are in the process of migrating from varying development stages to industrial utilization through technology transfer initiatives. The industrial partner was satisfied with the results, their technology transfer of the developed automated inspection cell is currently in the planning and design phase.

Recommendations

Some recommendations ultimately are derived from research conclusions and the enlightenment the researcher encounters during the research. These recommendations hopefully influence others to look in the same direction the researcher was at the terminal point of the research.

Investigation into ultrasonic inspection methodologies to filter out surface condition interference is necessary to reduce false echoes. The qualification of ultrasonic signatures is necessary for an expert system to increase the reliability and accuracy of defect detection. This may require scanning techniques other then the A-Scan used in this research. Scanning from multiple axes and using three dimensional imaging may be necessary to qualify the discontinuities.

The correlation of ultrasonic signatures with variable data from the casting process could lead to the type of quality analysis which will effect defect prevention. This also suggests that defect identification is a real possibility with more analysis and research. Other issues that need to be addressed are: probability of detection, new transducer coupling methods, focused versus unfocused transducers, signal processing, artificial intelligence, manufacturing intelligence, feedback process control, and managerial and worker resistance to new technology.

REFERENCES

American Foundrymen's Society. (1972). Casting defects handbook. Des Plaines, IL: Author.

Hénon, G., Mascré, C., & Blanc, G. (Eds.). (1974). International atlas of casting defects (M. T. Rowley Trans.). Des Plaines, IL: American Foundrymen's Society. (Original work published 1971)

Martin, J., & McClure, C. (1985). Diagramming techniques for analysts and programmers. Englewood Cliffs, NJ: Prentice-Hall.

Panametrics, Inc. (1991). Ultrasonic transducers for nondestructive testing. Waltham, MA: Author.

SPSS/PC+ 4.0 [Computer Program]. (1990). Chicago: SPSS, Inc.

Wolters, W. J. (1980). A computer-controlled ultrasonic nondestructive testing system. Unpublished doctoral dissertation, University of Missouri-Columbia, Columbia.

TABLES

Table 1
Software Development Verification
Dynamic Test Block Run 1

Block	Flaw Depth (in inches)	Go	NoGo	Error
A	.20	0	100	0%
B	.25	0	100	0%
C	.30	1	99	1%
D	.40	3	97	3%
E	.50	1	99	1%
F	.60	1	99	1%
G	.70	2	98	2%
X	---	48	52	52%
Y	---	56	44	44%

Note. Blocks X and Y do not have any flaws.

Table 2
Software Development Verification
Dynamic Test Block Run 5

Block	Flaw Depth (in inches)	Go	NoGo	Error
A	.20	0	100	0%
B	.25	0	100	0%
C	.30	0	100	0%
D	.40	0	100	0%
E	.50	0	100	0%
F	.60	0	100	0%
G	.70	0	100	0%
X	---	100	0	0%
Y	---	100	0	0%

Note. Blocks X and Y do not have any flaws.

Table 3
Inspection Results of Castings Failing UICCS Inspection Criteria

Serial Number	Inspection Points 01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	Summary UICCS	Manual
AZ(1)	P	P	P	P	P	P	P	P	P	P	P	F	P	P	P	P	P	F	F
AZ(2)	P	P	P	P	P	P	P	P	P	P	P	F	P	P	P	P	P	F	
AZ(3)	P	P	P	P	P	P	P	P	P	P	P	F	P	P	P	P	P	F	
BJ(1)	P	P	P	F	P	P	P	P	P	P	P	P	P	P	P	P	P	F	P
BJ(2)	P	P	P	F	P	P	P	P	P	P	P	P	P	P	P	P	P	F	
BJ(3)	P	P	P	F	P	P	P	P	P	P	P	P	P	P	P	P	P	F	
BS(1)	P	P	P	P	P	P	P	P	P	P	P	P	P	F	P	P	P	F	F
BS(2)	P	P	P	P	P	P	P	P	P	P	P	P	P	F	P	P	P	F	
BS(3)	P	P	P	P	P	P	P	P	P	P	P	P	P	F	P	P	P	F	
DK(1)	P	P	P	P	P	P	P	P	P	P	P	P	P	P	F	P	P	F	P
DK(2)	P	P	P	P	P	P	P	P	P	P	P	P	P	P	F	P	P	F	
DK(3)	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	
DX(1)	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	F	F	P
DX(2)	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	F	F	
DX(3)	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	

Note: P = Pass, F = Fail

FIGURES

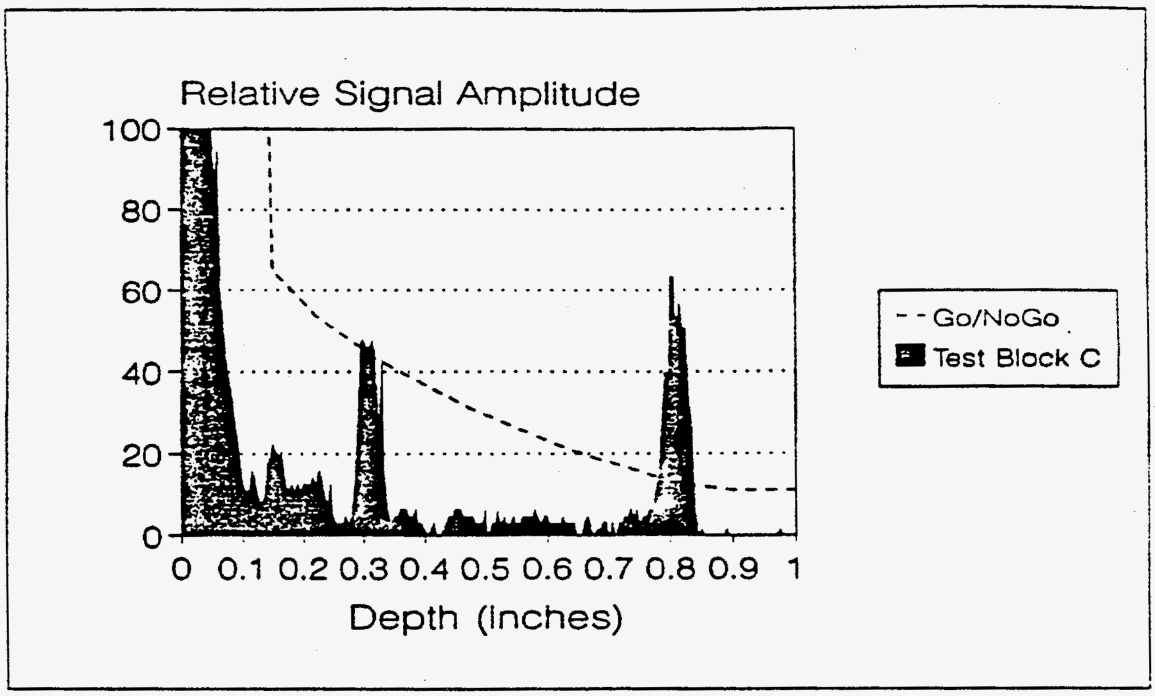


Figure 1. Ultrasonic A-Scan, test block C.

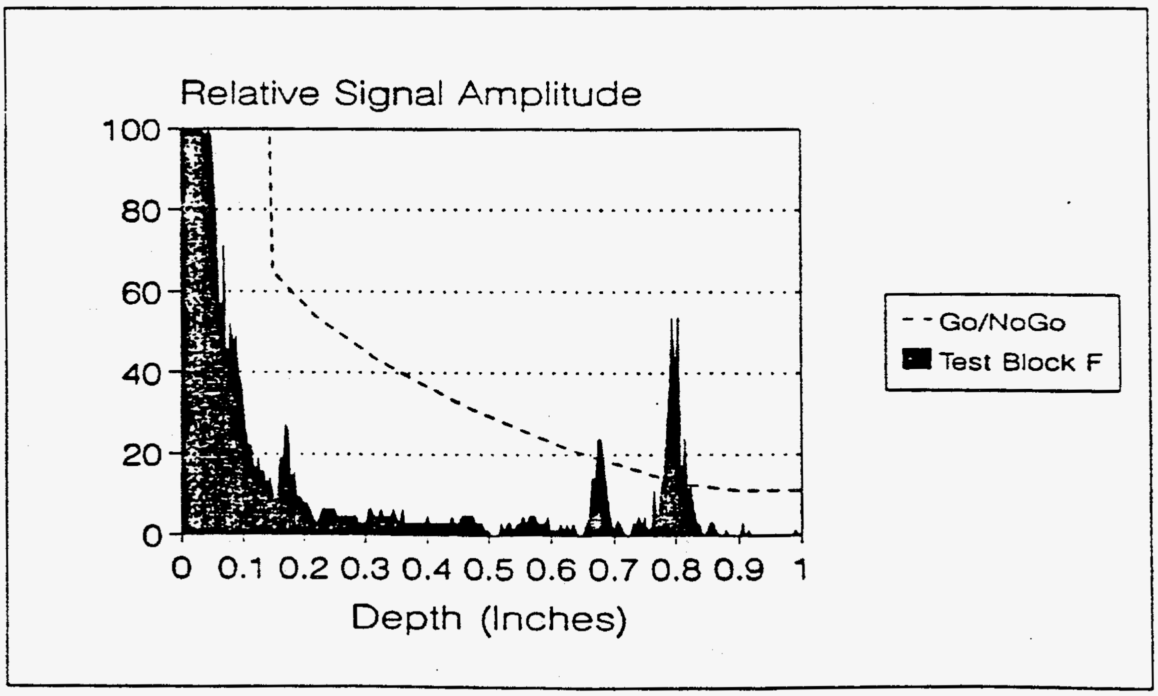


Figure 2. Ultrasonic A-Scan, test block F.

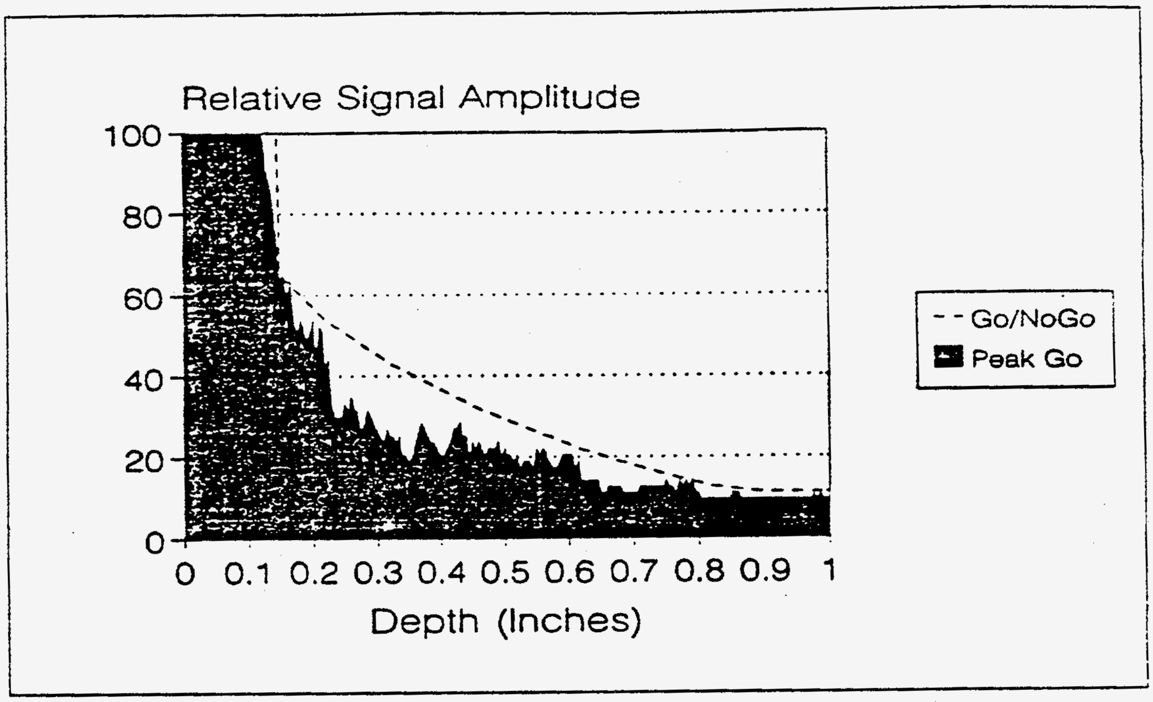


Figure 3. Ultrasonic A-Scan, peak go signals.

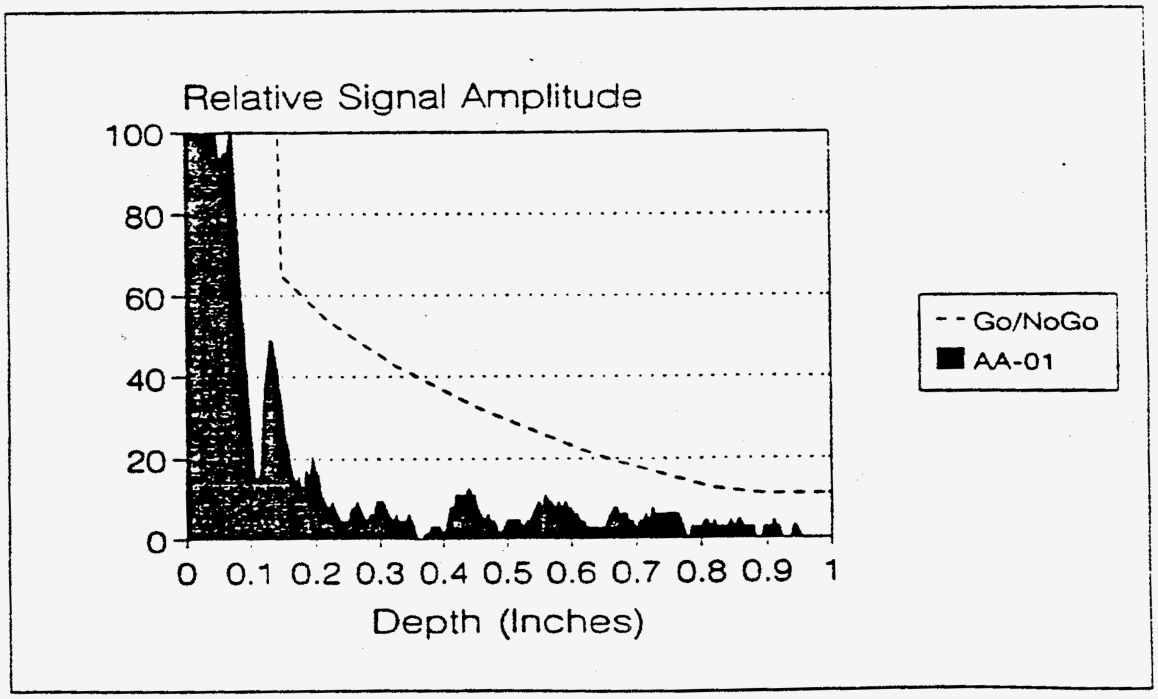


Figure 4. Ultrasonic A-Scan, AA-01.

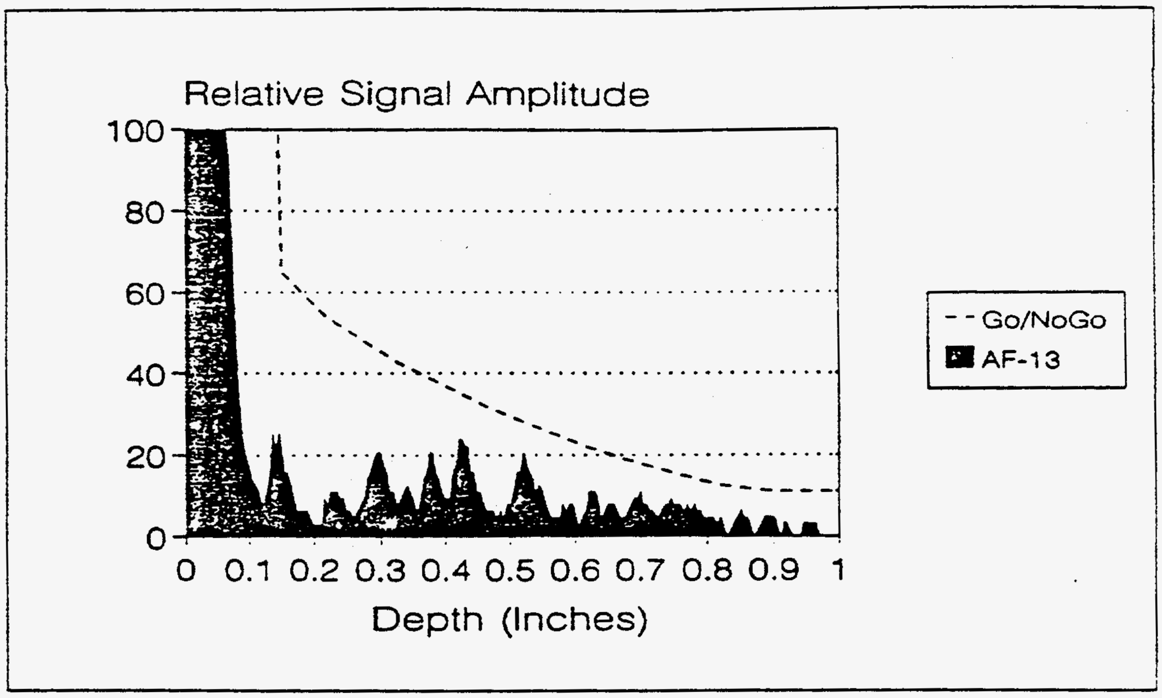


Figure 5. Ultrasonic A-Scan, AF-13.

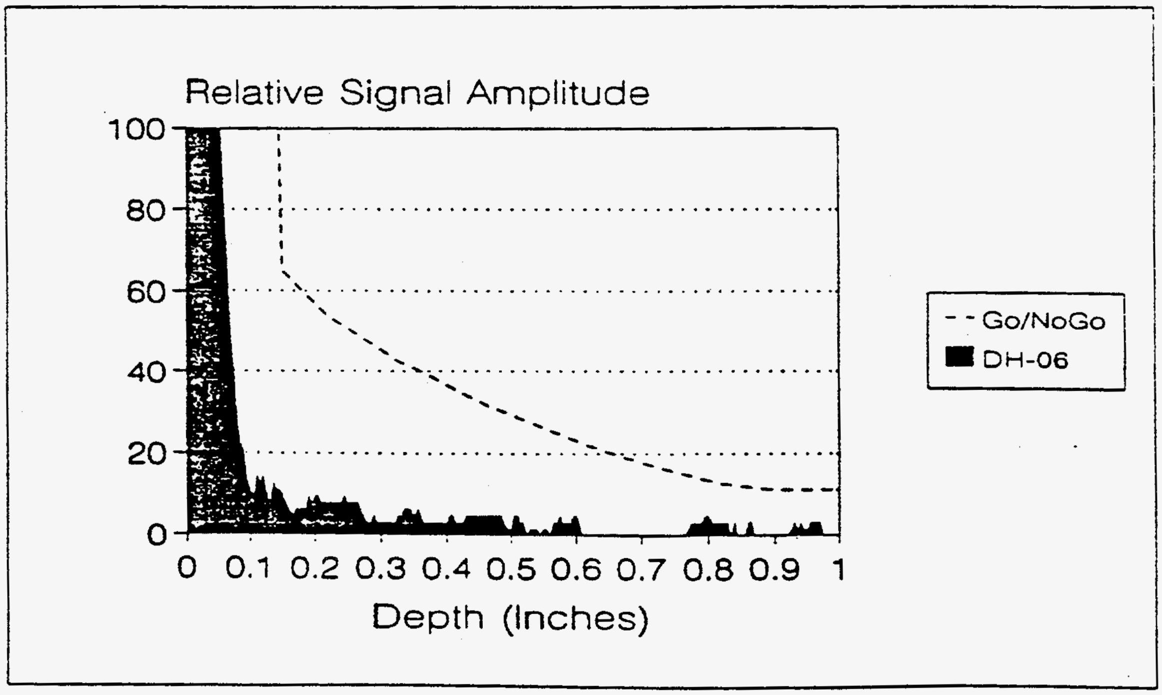


Figure 6. Ultrasonic A-Scan, DH-06.

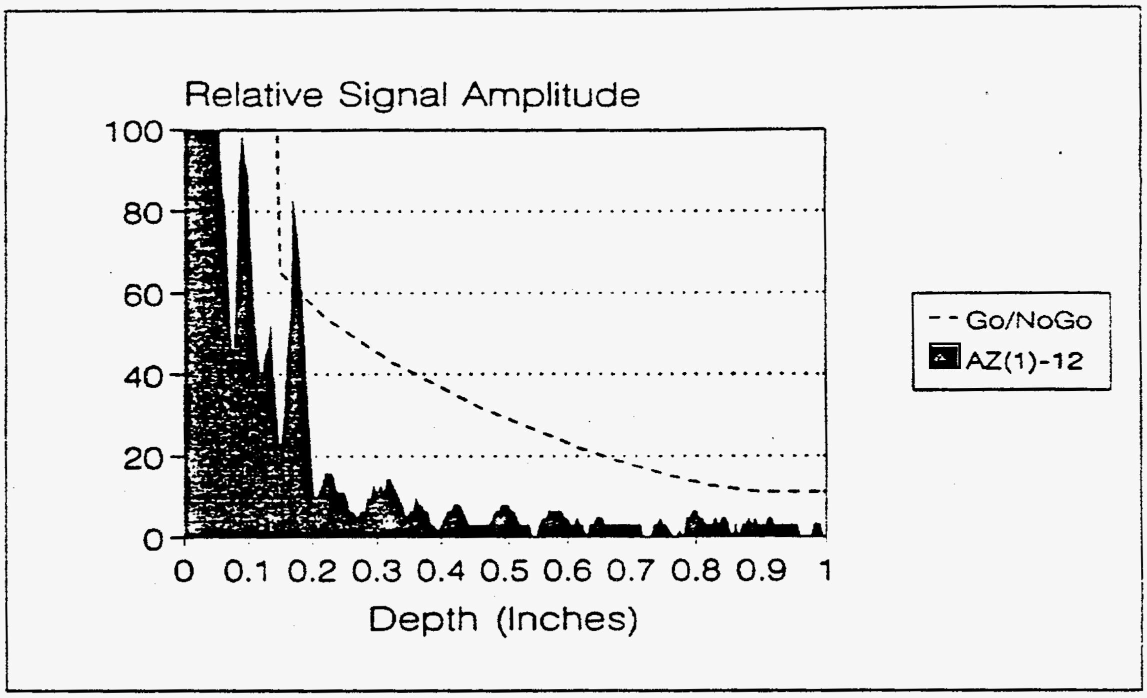


Figure 7. Ultrasonic A-Scan, AZ(1)-12.

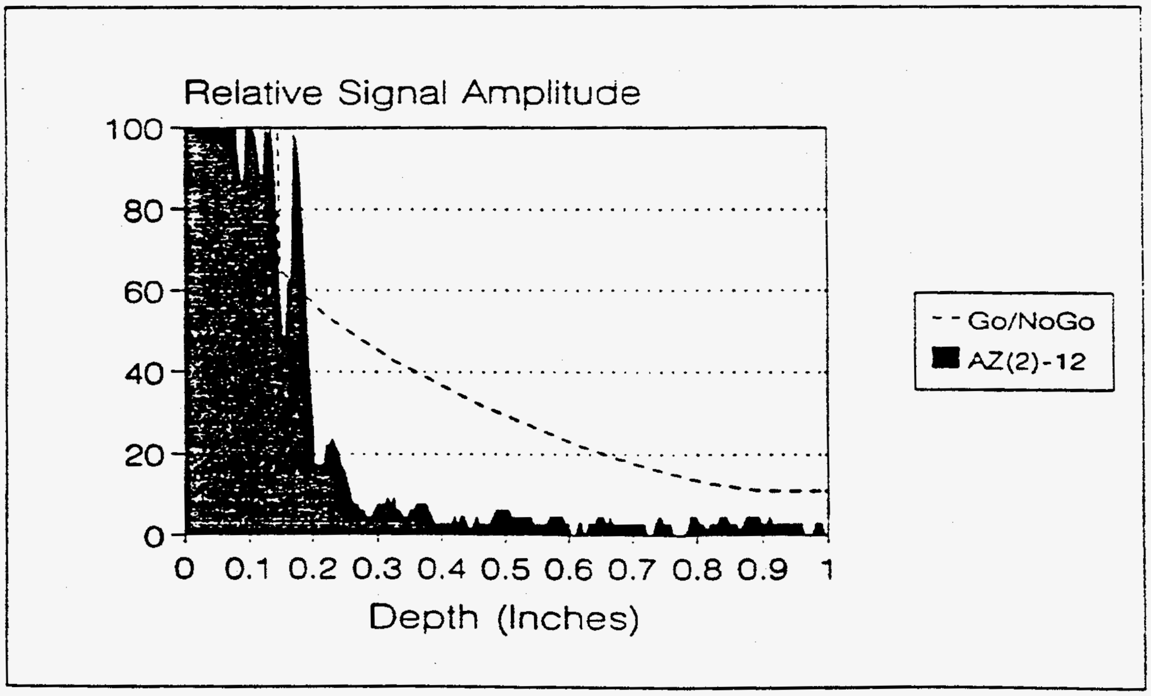


Figure 8. Ultrasonic A-Scan, AZ(2)-12.

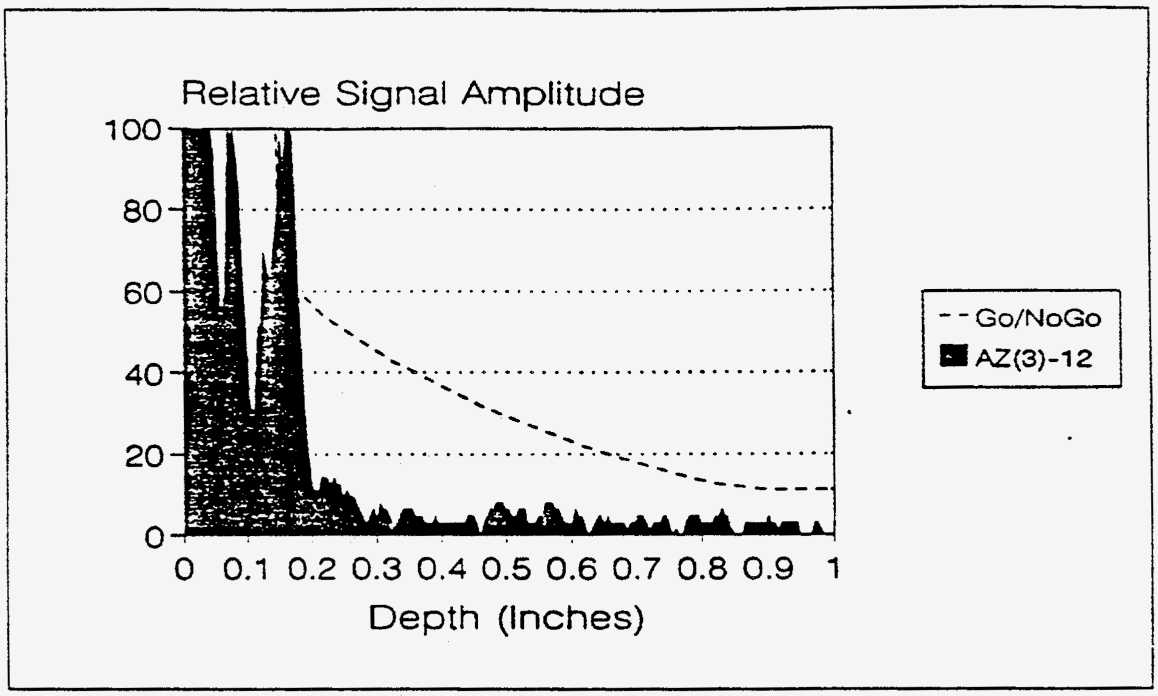


Figure 9. Ultrasonic A-Scan, AZ(3)-12.

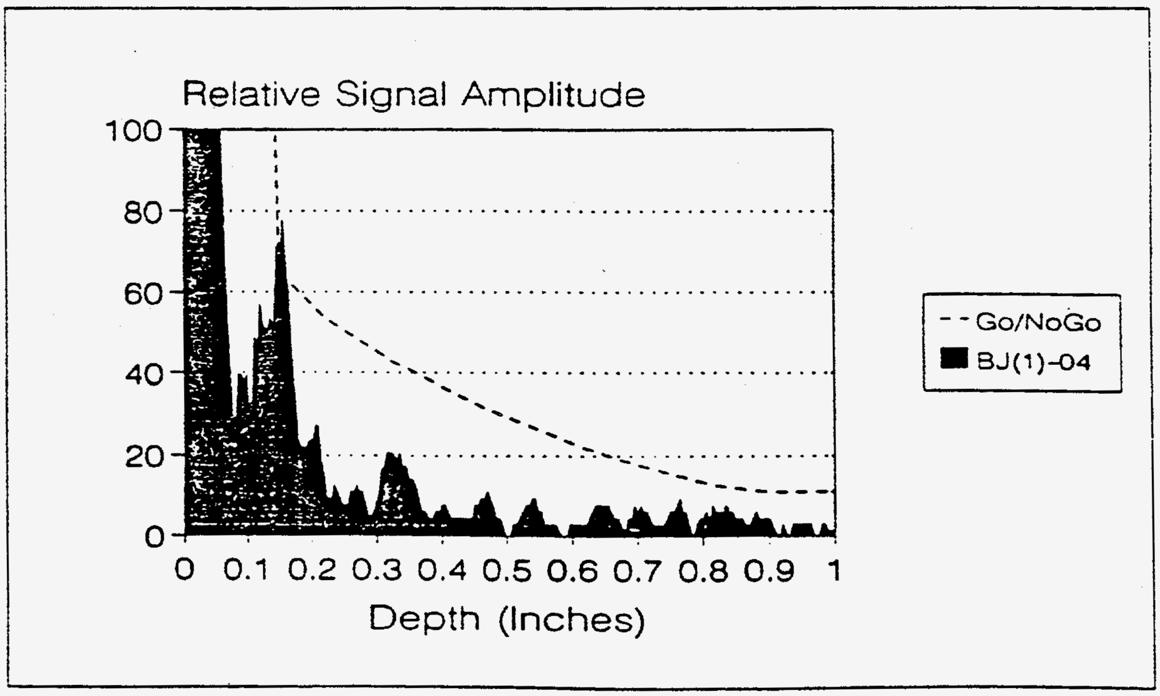


Figure 10. Ultrasonic A-Scan, BJ(1)-04.

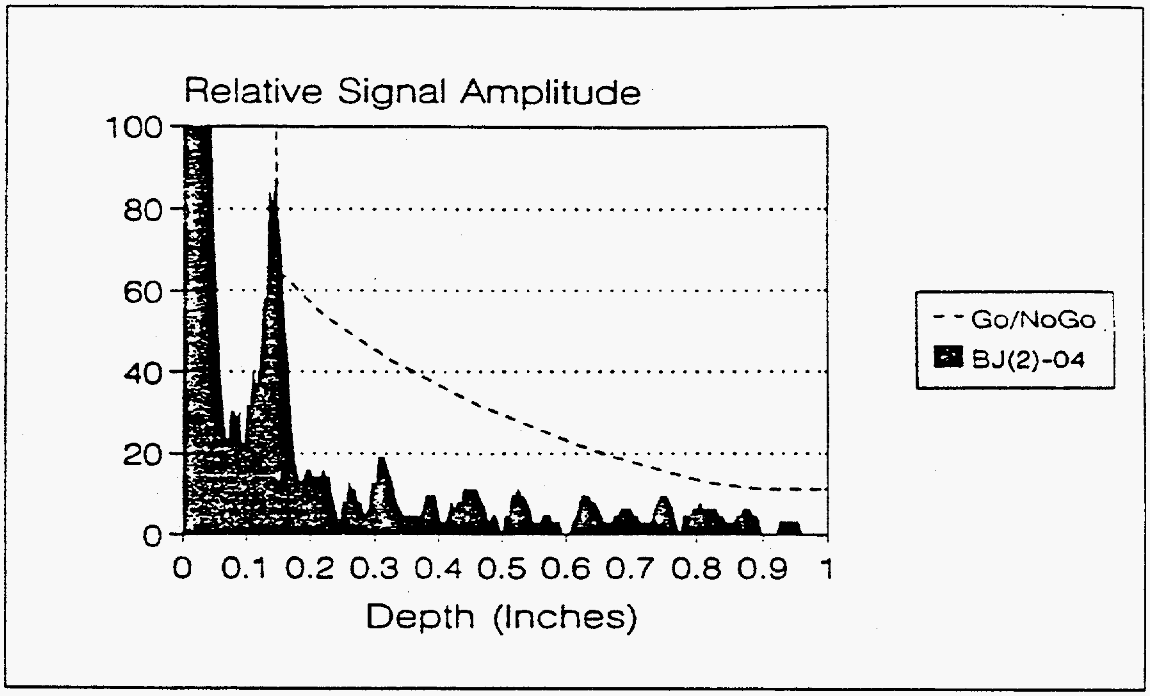


Figure 11. Ultrasonic A-Scan, BJ(2)-04.

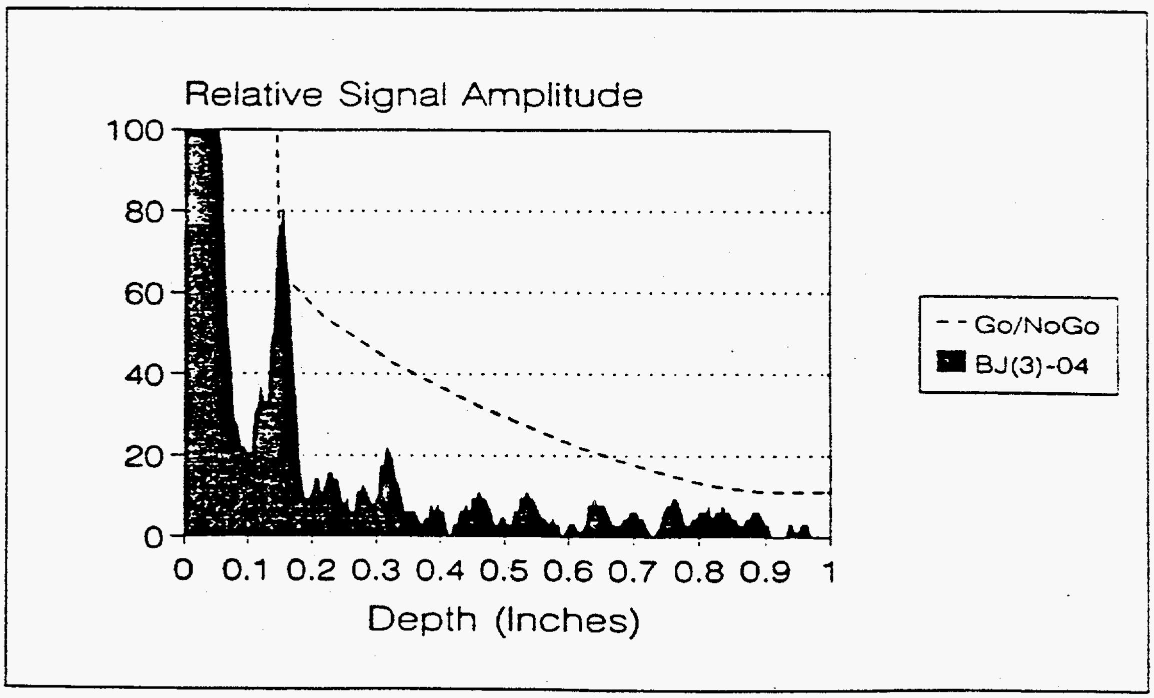


Figure 12. Ultrasonic A-Scan, BJ(3)-04.

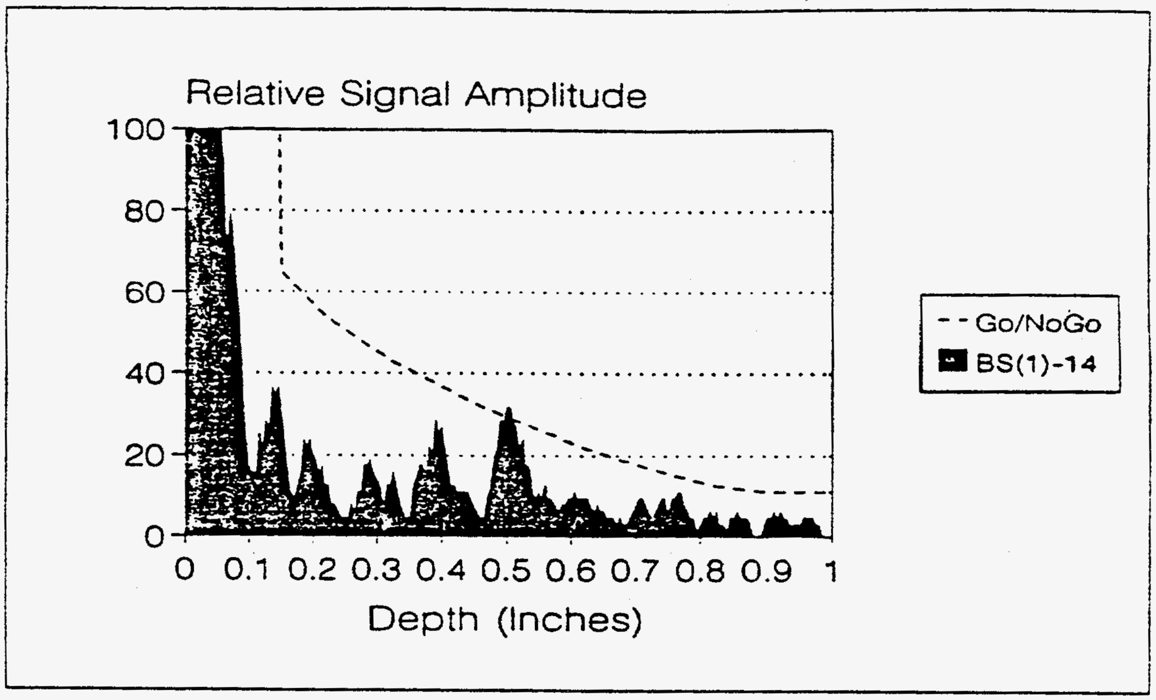


Figure 13. Ultrasonic A-Scan, BS(1)-14.

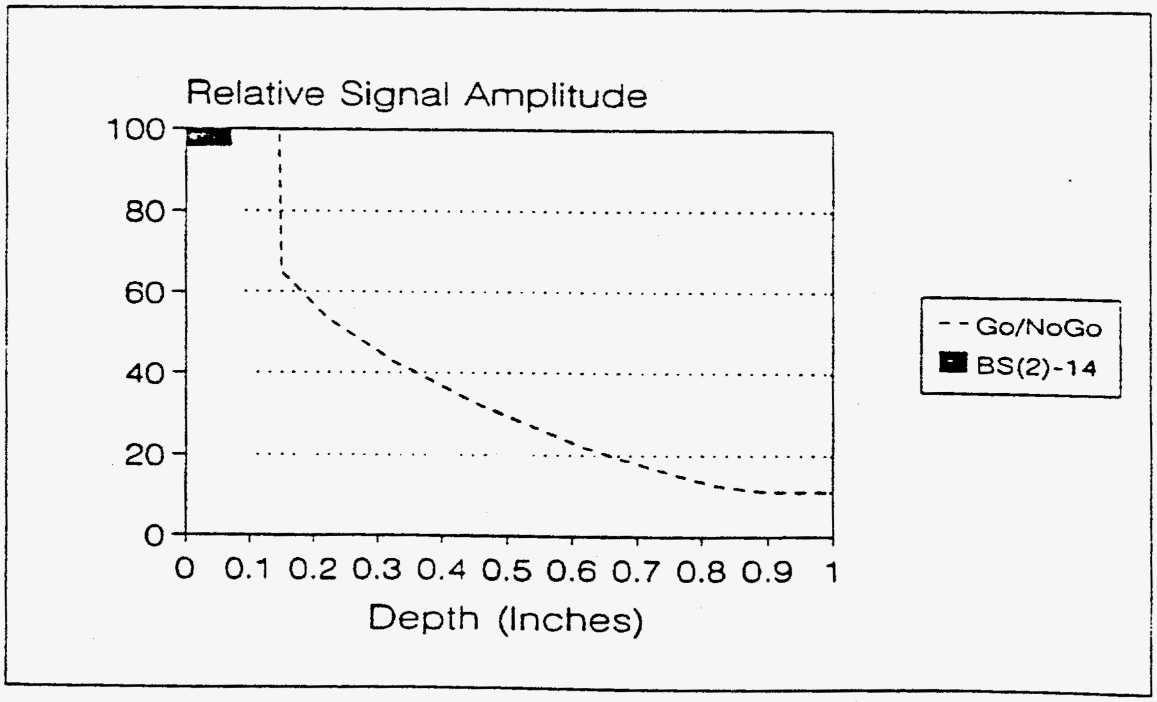


Figure 14. Ultrasonic A-Scan, BS(2)-14.

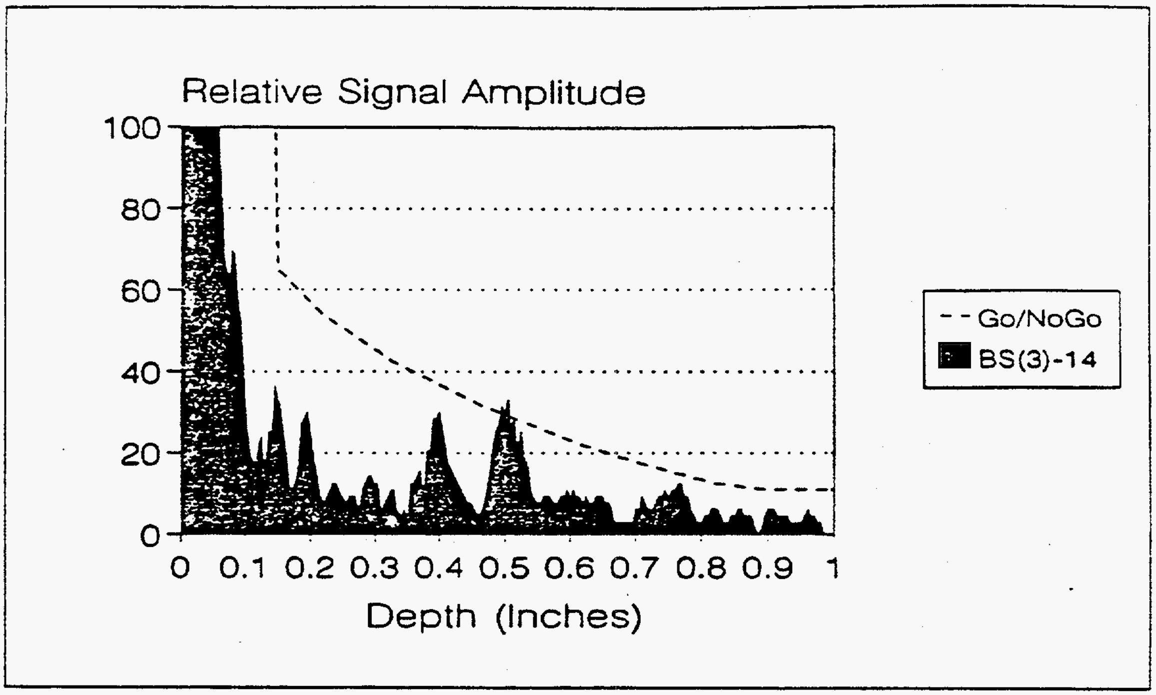


Figure 15. Ultrasonic A-Scan, BS(3)-14.

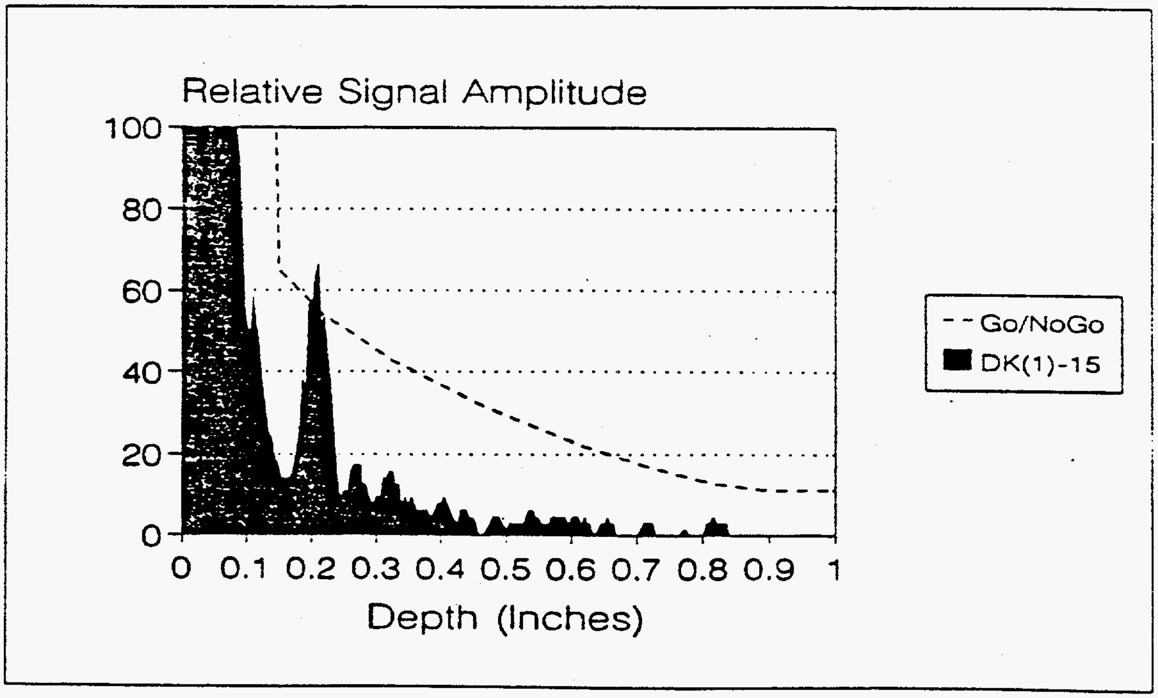


Figure 16. Ultrasonic A-Scan, DK(1)-15.

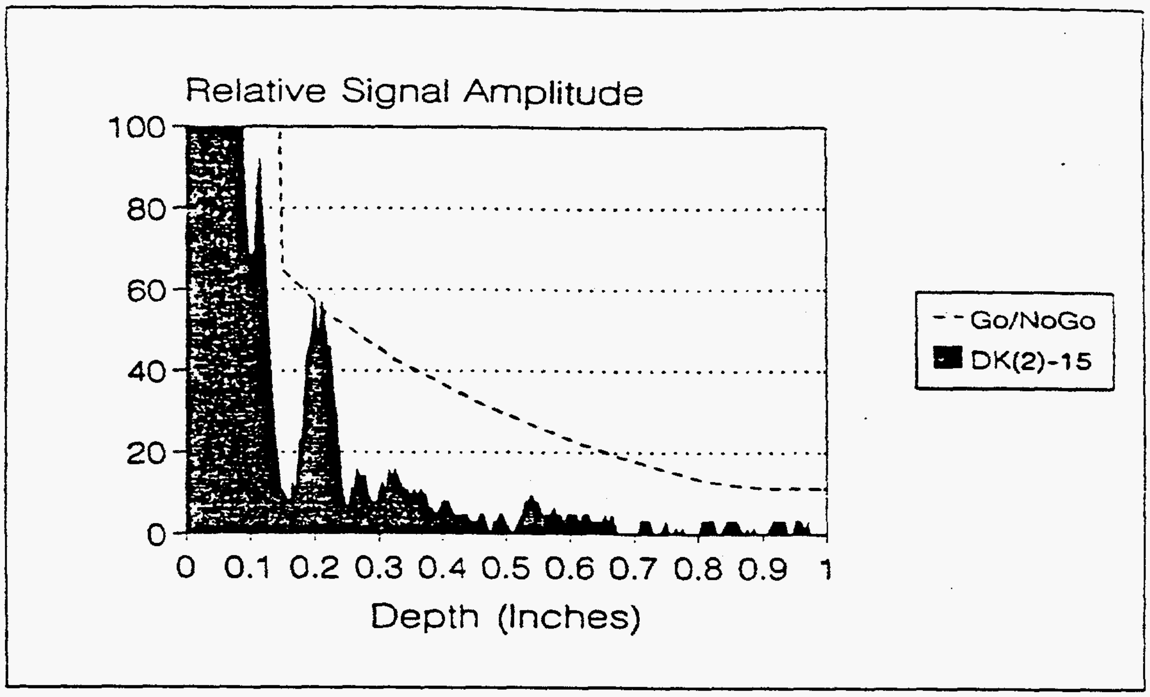


Figure 17. Ultrasonic A-Scan, DK(2)-15.

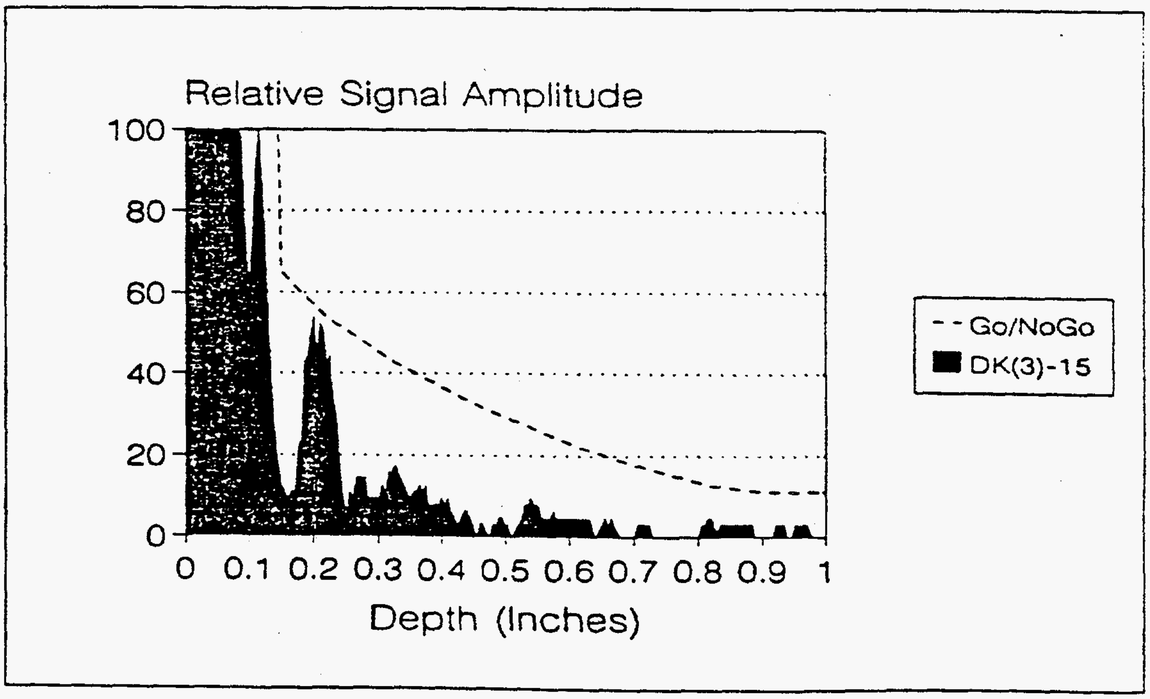


Figure 18 Ultrasonic A-Scan, DK(3)-15.

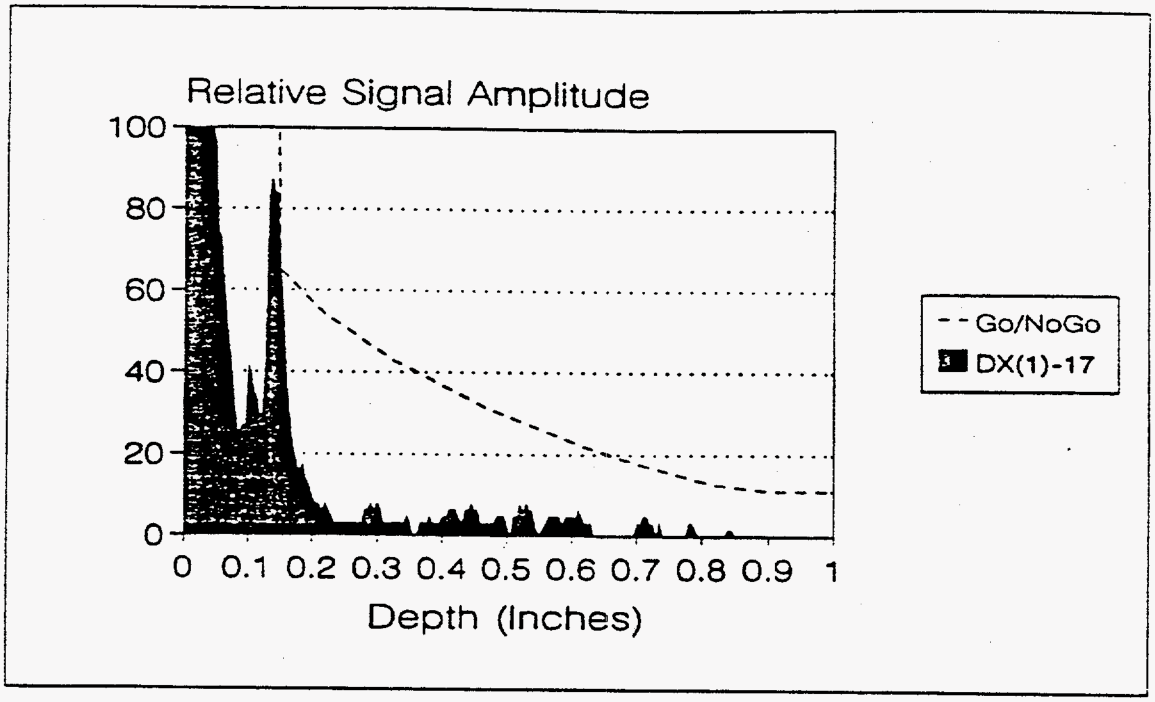


Figure 19. Ultrasonic A-Scan, DX(1)-17.

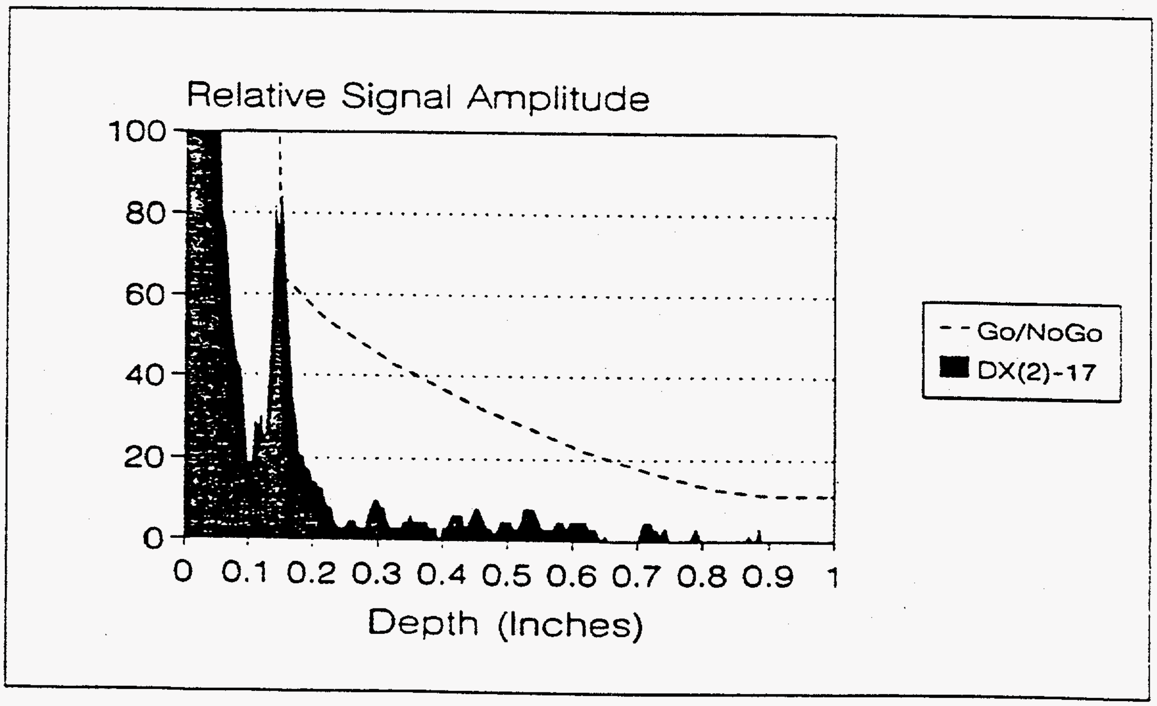


Figure 20. Ultrasonic A-Scan, DX(2)-17.

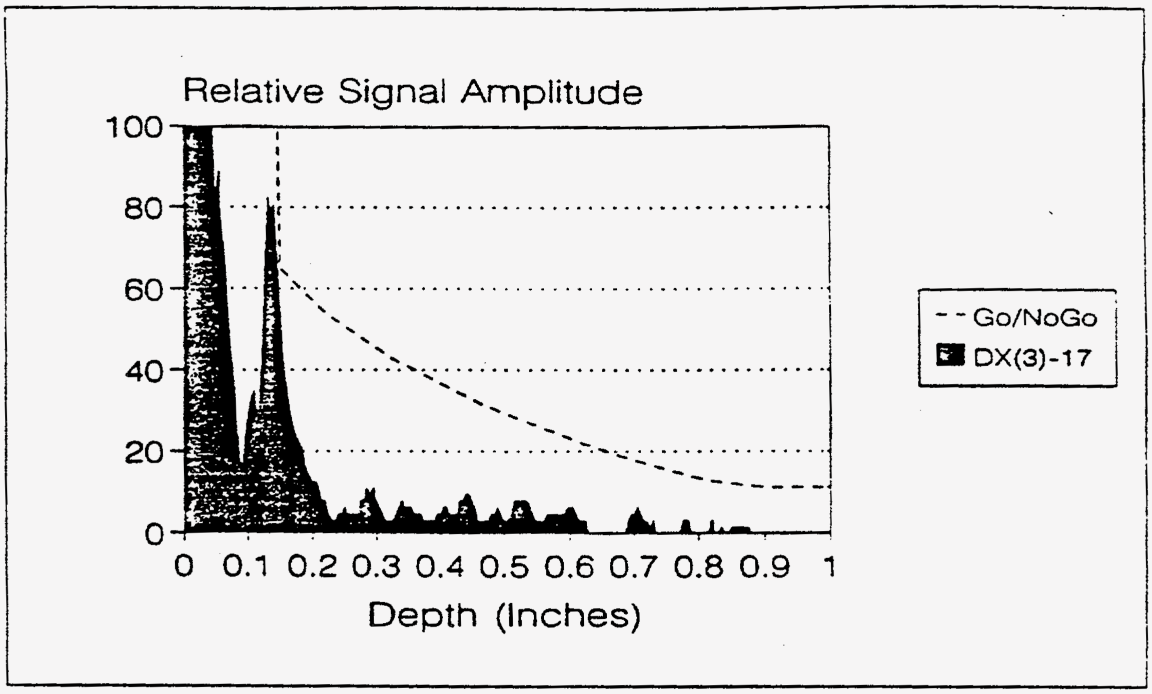


Figure 21. Ultrasonic A-Scan, DX(3)-17.

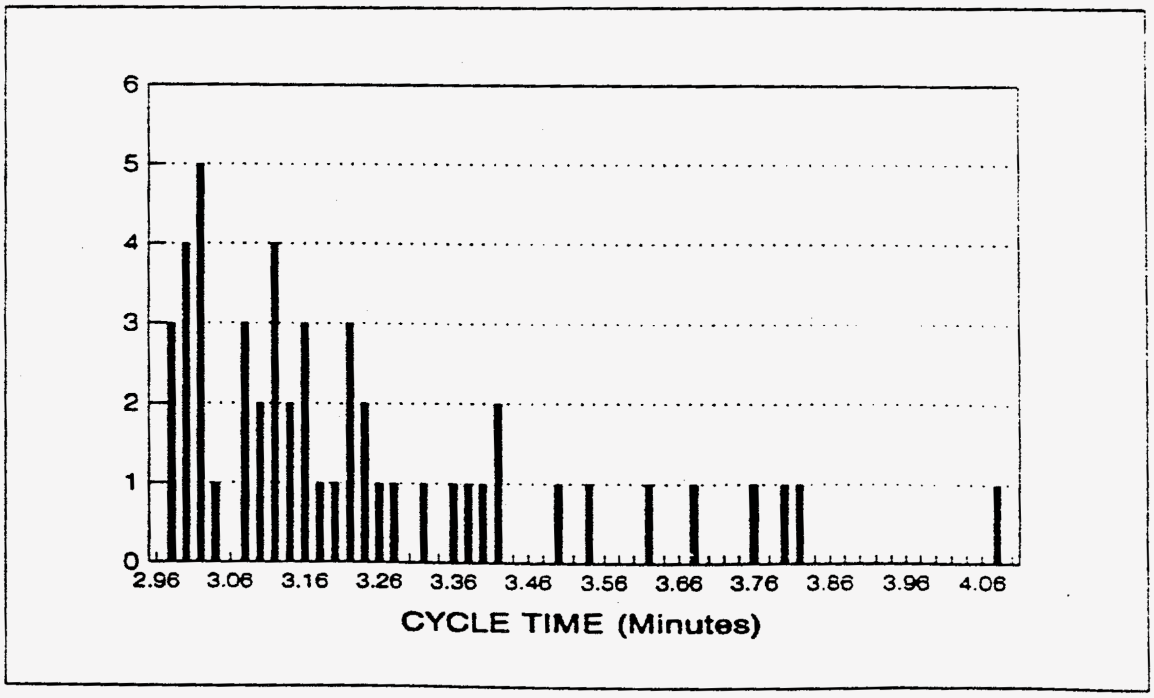


Figure 22. Automated inspection cycle time for one part.

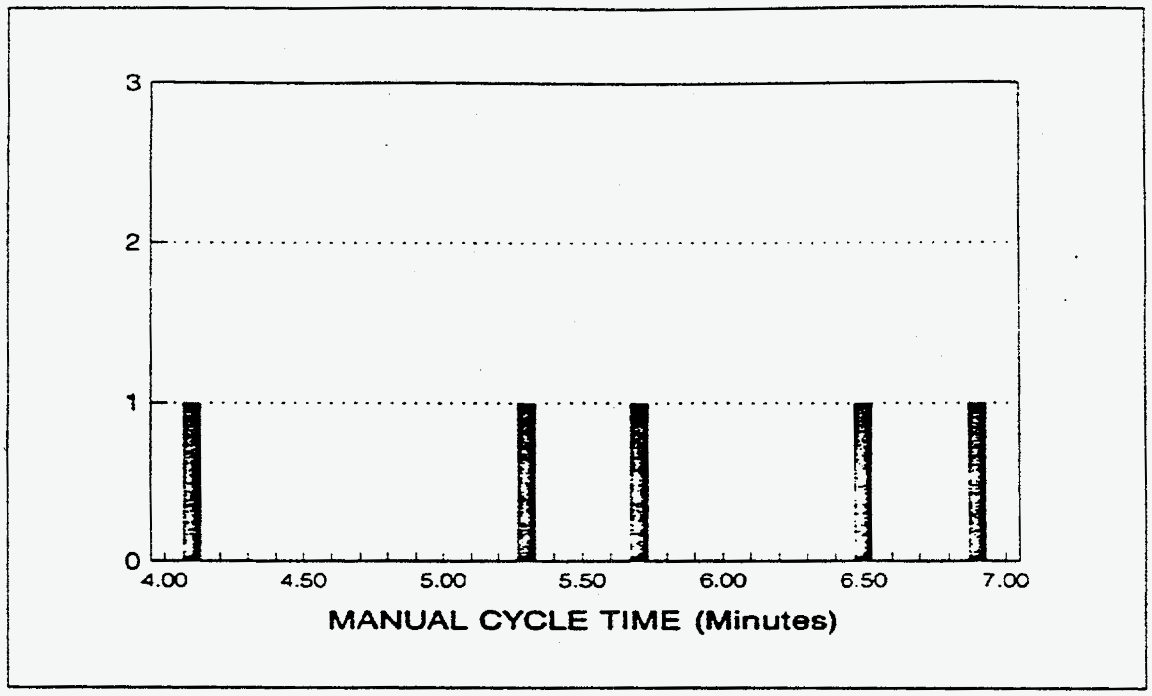


Figure 23. Manual inspection cycle time for one boss.

APPENDIX A

UICCS PROGRAM ACTION DIAGRAM

```
' **************************************************************************
' *    Ultrasonic Inspection Computer Control Software (UICCS) Project     *
' **************************************************************************
' *    University of Northern Iowa                                         *
' *    Dept of Industrial Technology                                       *
' *    Metal Casting Center (MCC)                                          *
' *    Cedar Falls, Iowa  50614-0178                                       *
' **************************************************************************
' *    Copyright 1992, Metal Casting Center, University of Northern Iowa   *
' *    All rights reserved.                                                *
' **************************************************************************
'      Written by: John S. Burningham, D.I.T.
'                  PO Box 616
'                  Mahopac, NY  10541-0616
'                  (914)628-7952
'
'      Hardware Requirements:  80286/80386/80486 IBM compatible
'                              One Serial (RS-232) Port
'                              One Parallel Printer Port
'                              VGA Graphics
'                              1 MB Ram Memory min.
'                              Hard Drive
'
'                              (Digital Isolated I/O Board)
'                              Model DIO8-P
'                              bus address: &H300
'                                 Industrial Computer Source
'                                 4837 Mercury St.
'                                 San Diego CA 92111
'                                 (619)279-0084
'
'                              (Digital Ultrasonic Flaw Detector)
'                              EPOCH 2002 w/RS-232 Interface (19200 baud)
'                                 Panametrics, Inc.
'                                 221 Crescent Street
'                                 Waltham MA 02254
'                                 (617)899-2719
'
'      Software Development System: MS-DOS V5.00
'                                   QuickBasic V4.5
'
' ************************************************************************
' *    Documentation Section            Revised 10/21/92  jsb            *
' ************************************************************************
'
'      Inspection Data Output File  --  [serialnumber.INS] (3643 bytes)
'
'      POSITION     DESCRIPTION
'      0001-0008    Part Serial Number
'      0009-0022    Date-Time stamp [yyyymmddhhmmss]
'      0023-0026    Decibel Level (Single percission variable)
'      0027-0043    Pass/Fail summary [P/F] for points 1-17
'      0044-0243    Inspection Reject Table
'      0244-0443    A-Scan Data Set -- Inspection Point 01
'      0444-0643    A-Scan Data Set -- Inspection Point 02
'      0644-0843    A-Scan Data Set -- Inspection Point 03
'      0844-1043    A-Scan Data Set -- Inspection Point 04
'      1044-1243    A-Scan Data Set -- Inspection Point 05
'      1244-1443    A-Scan Data Set -- Inspection Point 06
'      1444-1643    A-Scan Data Set -- Inspection Point 07
'      1644-1843    A-Scan Data Set -- Inspection Point 08
'      1844-2043    A-Scan Data Set -- Inspection Point 09
'      2044-2243    A-Scan Data Set -- Inspection Point 10
'      2244-2443    A-Scan Data Set -- Inspection Point 11
'      2444-2643    A-Scan Data Set -- Inspection Point 12
'      2644-2843    A-Scan Data Set -- Inspection Point 13
'      2844-2043    A-Scan Data Set -- Inspection Point 14
'      3044-3243    A-Scan Data Set -- Inspection Point 15
'      3244-3443    A-Scan Data Set -- Inspection Point 16
'      3444-3643    A-Scan Data Set -- Inspection Point 17
'
'
```

```
DEFINT A-Z      ' Default Variable type
CONST true = -1
CONST false = 0
CONST nul = ""
CONST Star = "*"
DIM AdumpValue(200) AS INTEGER
DIM RejectString AS STRING * 200
DIM DateTimeString AS STRING * 14
DIM FrontZeroOffset AS SINGLE
DIM PartSerialNumber AS STRING
DIM PutPSN AS STRING * 8
DIM Realtmp AS SINGLE
DIM RejectTable(200) AS INTEGER
DIM RobotDelayTimer AS SINGLE
DIM Decibel AS SINGLE
DIM Sort1 AS STRING * 8
DIM Sort2 AS STRING * 8
DIM Sort(5000) AS STRING * 8
Decibel = 67!
EOB$ = CHR$(23)
ESC$ = CHR$(27)
CR$ = CHR$(13)
OK$ = "OK"
'
'  Define Inspection Record
'
TYPE Type1
     PSN AS STRING * 8       '  Part Serial Number
     DTS AS STRING * 14      '  Date/Time
     DB AS SINGLE            '  Signal Level
     PF AS STRING * 17       '  Pass/Fail
     RT AS STRING * 200      '  Reject Table
     DAT AS STRING * 3400    '  Inspection Date  (200 bytes * 17 points)
END TYPE
DIM InspRecord AS Type1
'
'  Read COMMAND Line for runtime options
'
IF INSTR(COMMAND$, "/D") > 0 THEN
     DebugFlag = true
ELSE
     DebugFlag = false
END IF
IF INSTR(COMMAND$, "/Q") > 0 THEN
     SoundFlag = false
ELSE
     SoundFlag = true
END IF
IF INSTR(COMMAND$, "/M") > 0 THEN
     colorf = 7
ELSE
     colorf = 14
END IF
'
'  User Instructions for command line "?"
'
IF INSTR(COMMAND$, "?") THEN
     PRINT
     PRINT "Command Line Options:"
     PRINT "    /D     Debug"
     PRINT "    /Q     Quite (No sound)"
     PRINT "    /M     Monochrome (No Color)"
     GOTO byebyeend
END IF
'
'  Check for DIO8 Board at &H300 address
'
OUT &H300, 0    '    Force DIO8 to zero
IF INP(&H300) = 255 THEN '    If no board, value will be 255
     CLS
     PRINT "Robot Digital I/O Board not detected at address &H300"
     IF NOT DebugFlag THEN
          PRINT "Disabling Inspection Module"
          PRINT "(You can restart the program with a /D option to enable)"
     END IF
     INPUT "Press <enter> to continue: ", s$
```

```
│       DIO8Flag = false
├ELSE
│       DIO8Flag = true
└END IF
```

```
FOR loopcount = 1 TO 8
     RobotBit(loopcount) = false
NEXT
'
'  Allow Fn keys to toggal Output bits on DIO8 board for debugging
'
IF DebugFlag THEN
     ON KEY(1) GOSUB F1key
     KEY(1) ON
     ON KEY(2) GOSUB F2key
     KEY(2) ON
     ON KEY(3) GOSUB F3key
     KEY(3) ON
     ON KEY(4) GOSUB F4key
     KEY(4) ON
     ON KEY(5) GOSUB F5key
     KEY(5) ON
     ON KEY(6) GOSUB F6key
     KEY(6) ON
     ON KEY(7) GOSUB F7key
     KEY(7) ON
     ON KEY(8) GOSUB F8key
     KEY(8) ON
     DisplayBoxTop$ = CHR$(201) + STRING$(8, CHR$(205)) + CHR$(187)
     DisplayBoxMiddle$ = CHR$(186) + "        " + CHR$(186)
     DisplayBoxBottom$ = CHR$(200) + STRING$(8, CHR$(205)) + CHR$(188)
END IF
'
'  Reject Table (Go/NoGO) Table
'
DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
DATA 00,00,00,00,00,00,00,00,00,00,41,41,40,40,39,39,38,38,37,37
DATA 36,36,35,35,34,34,34,33,33,33,32,32,32,31,31,31,30,30,30,29
DATA 29,29,28,28,28,27,27,27,27,26,26,26,26,25,25,25,25,24,24,24
DATA 24,23,23,23,23,22,22,22,22,21,21,21,21,20,20,20,20,20,19,19
DATA 19,19,19,18,18,18,18,18,17,17,17,17,17,16,16,16,16,16,15,15
DATA 15,15,15,14,14,14,14,14,14,13,13,13,13,13,13,12,12,12,12,12
DATA 12,12,11,11,11,11,11,11,11,10,10,10,10,10,10,10,09,09,09,09
DATA 09,09,09,09,08,08,08,08,08,08,08,08,08,08,08,08,08,08,08,08
DATA 07,07,07,07,07,07,07,07,07,07,07,07,07,07,07,07,07,07,07,07
'
'  This routine reads the reject table and creates RejectString
'
tmp$ = nul
FOR SubScript = 1 TO 200
     READ RejectTable(SubScript)
     tmp$ = tmp$ + CHR$(RejectTable(SubScript))
NEXT
RejectString = tmp$
```

```
'
'  Screen Mode 12 (VGA) with blue background for color
'
ON ERROR GOTO NoVGA
SCREEN 12
IF colorf = 14 THEN
     PALETTE 0, 65536 * 25
END IF
'
'  Setup Error trapping
'
ON ERROR GOTO ErrorTrap
'
'  Clear Robot activity flag
'
RobotActiveFlag = false
'
'  Initialize screen width and foreground color
'
WIDTH 80, 30
CLS
COLOR colorf
'
'  Initialize Clock Display
'
ON TIMER(1) GOSUB ClockDisplay
'
'  Display Intro Screen
'
GOSUB IntroScreen
```

```
'
'  Main Menu Loop
'
DO WHILE MainMenuSelection <> 4
    '
    '  Display Main Menu
    '
    GOSUB ClearViewPort
    s$ = "M A I N   M E N U"
    LOCATE 10, 40 - LEN(s$) / 2
    PRINT s$;
    column = 26
    LOCATE 13, column
    IF DIO8Flag OR DebugFlag THEN
        PRINT "1. INSPECT PART";
    ELSE
        PRINT "1. <<disabled>>";
    END IF
    LOCATE 15, column
    PRINT "2. Report Menu";
    LOCATE 17, column
    PRINT "3. Display Inspection Record";
    LOCATE 19, column
    PRINT "4. Quit (Exit to DOS)";
    LOCATE 21, column
    COLOR 15
    PRINT "Enter Selection: ";
    COLOR colorf
    PRINT CHR$(178);
    LOCATE 21, column + 17
    '
    '  Get menu selection
    '
    DO
        MainMenuSelection$ = INKEY$
    LOOP WHILE MainMenuSelection$ = nul
    PRINT MainMenuSelection$;
    selection = VAL(MainMenuSelection$)
    IF (selection > 1 AND selection < 5) OR (selection = 1 AND (DIO8Flag OR DebugFlag)) THEN
        IF SoundFlag THEN SOUND 1000, .5
        SELECT CASE selection
            CASE 1
                GOSUB InspectPart
            CASE 2
                GOSUB ReportMenu
            CASE 3
                GOSUB DisplayInspectionRecord
            CASE 4
                EXIT DO
        END SELECT
        selection = false '  force continued looping
    ELSE
        GOSUB InvalidEntry
    END IF
LOOP
IF SoundFlag THEN SOUND 2000, .5
GOTO byebye

byebye:
    TIMER OFF
    '
    '  Force DIO8 board to zero
    '
    OUT &H300, 0
    '
    '  Reset screen and terminate execution
    '
    SCREEN 0
    COLOR 7, 0      ' reset screen colors
    CLS
byebyeend:
    END
```

```
ClearViewPort:
     TIMER STOP
     VIEW PRINT 9 TO 30
     CLS 2
     VIEW PRINT
     TIMER ON
     RETURN

ClockDisplay:
     ClockDisplayRow = CSRLIN
     ClockDisplayColumn = POS(0)
     LOCATE 5, 31
     PRINT DATE$; " "; TIME$;
     LOCATE ClockDisplayRow, ClockDisplayColumn
     GOSUB DebugDisplayIO
     IF TimeOutTimer < 32767 THEN TimeOutTimer = TimeOutTimer + 1      ' Increment timer
     '
     '  Force error on lack of Robot motion
     '
   ┌─IF RobotActiveFlag AND TimeOutTimer > 30 THEN
   │      ERROR 254
   └─END IF
   ┌─IF IntroScreenFlag THEN
   │    ┌─DO
   │    │       IntroScreenColor = RND * 11 + 1
   │    └─LOOP WHILE IntroScreenColor = LastIntroScreenColor
   └─END IF
     RETURN

ComOpen:
     ErrorFlag = false
     OPEN "COM2:19200,N,8,1,BIN,CS0,DS0,CD0,RB1024" FOR RANDOM AS #6
     GOSUB SendStar
     PRINT #6, "DISP=G"
     GOSUB ReadResponse
     RETURN

ComClose:
     CLOSE #6
     RETURN
```

```
dBcalculate:
     dBreal! = 0
     AdumpPeakLoop = AdumpPeak
    DO WHILE AdumpPeakLoop < 5
          dBreal! = dBreal! + 2.2
          AdumpPeakLoop = AdumpPeakLoop + 1
    LOOP
     '
    DO WHILE AdumpPeakLoop < 16
          dBreal! = dBreal! + .6
          AdumpPeakLoop = AdumpPeakLoop + 1
    LOOP
     '
    DO WHILE AdumpPeakLoop < 27
          dBreal! = dBreal! + .3
          AdumpPeakLoop = AdumpPeakLoop + 1
    LOOP
     '
    DO WHILE AdumpPeakLoop < 40
          dBreal! = dBreal! + .2
          AdumpPeakLoop = AdumpPeakLoop + 1
    LOOP
     '
    DO WHILE AdumpPeakLoop < 62
          dBreal! = dBreal! + .15
          AdumpPeakLoop = AdumpPeakLoop + 1
    LOOP
     '
    IF AdumpPeakLoop = 62 THEN
          dBreal! = dBreal! + .19
          IF DB > 3 AND dBreal! = .19 THEN dBreal! = .2
    ELSE
          dBreal! = 0    '    Done
    END IF
     DB = INT(dBreal! * 10)
     '
     '  Force Error if excessive dB
     '
    IF ReaddB + DB > 1000 THEN
          ERROR 253
    END IF
     RETURN

dBchange:
    IF DB <> 0 THEN
         IF ReaddB = 0 THEN
               GOSUB SendStar
               PRINT #6, "DB=?"
               GOSUB ReadResponse
               ReaddB = CINT(VAL(MID$(ResponseString$, INSTR(ResponseString$, CHR$(10) + "DB=") +
                                                    4)) * 10)
         END IF
          '
          '  SET SYSTEM SENSITIVITY
          '
          GOSUB SendStar
          PRINT #6, USING "DB=###.#"; (ReaddB + DB) / 10
          GOSUB ReadResponse
          ReaddB = ReaddB + DB
    END IF
     RETURN
```

```
DebugDisplayIO:
  IF DebugFlag THEN
        '
        '  Force DebugFlag to prevent recurrsive call
        '
        DebugFlag = false
        TIMER OFF
        '
        '  Save current cursor position
        '
        DebugDisplayIOrow = CSRLIN
        DebugDisplayIOcolumn = POS(0)
        FOR DebugDisplayIOloop1 = 1 TO 3
            LOCATE DebugDisplayIOloop1 + 1, 71
            SELECT CASE DebugDisplayIOloop1
                CASE 1
                    CmdValue1 = CmdValue
                CASE 2
                    IF DIO8Flag THEN
                        CmdValue1 = INP(&H300)
                    ELSE '    DIO8 Board not installed, Allow FunKeys to force condition
                        CmdValue1 = CmdValue
                    END IF
                    IF CmdValue1 <> InHex300 THEN
                        SOUND 750, 1
                    END IF
                    InHex300 = CmdValue1
                CASE 3
                    GOSUB Hex301Get
            END SELECT
            DS$ = nul
            FOR DebugDisplayIOloop2 = 7 TO 0 STEP -1
                IF CmdValue1 >= 2 ^ DebugDisplayIOloop2 THEN
                    DS$ = DS$ + "1"
                    CmdValue1 = CmdValue1 - 2 ^ DebugDisplayIOloop2
                ELSE
                    DS$ = DS$ + "0"
                END IF
            NEXT
            PRINT DS$;
        NEXT
        LOCATE 2, 1
        PRINT USING "FRE(-1):######"; FRE(-1)
        PRINT USING "FRE(-2):######"; FRE(-2)
        PRINT USING "FRE(-3):######"; FRE(-3)
        PRINT USING "Timeout:######"; TimeOutTimer
        LOCATE DebugDisplayIOrow, DebugDisplayIOcolumn
        '
        '  Restore DebugFlag
        '
        DebugFlag = true   '    Reset flag
        TIMER ON
  END IF
  RETURN
```

```
DisplayInspectionRecord:
     GOSUB ClearViewPort
     LOCATE 13, 27
     PRINT "DISPLAY INSPECTION RECORD";
     GOSUB GetPartSerialNumber
     IF PartSerialNumber <> nul AND tmpASC <> 27 THEN
          '
          ' Open Data File
          '
          DataFile$ = RTRIM$(PartSerialNumber) + ".INS"
          OPEN DataFile$ FOR BINARY AS #1
          IF LOF(1) = 0 THEN
               '
               ' File is empty (DID NOT EXIST)
               '
               CLOSE #1
               KILL DataFile$
               LOCATE 17, 28
               PRINT "Data File does not exist";
               IF SoundFlag THEN
                    FOR Scan = 1 TO 20
                         SOUND 1300, .4
                         SOUND 1000, .4
                         SOUND 700, .4
                    NEXT
               END IF
          ELSE
               '
               ' Get Data from file
               '
               GET #1, 1, InspRecord
               '
               ' Close Data File
               '
               CLOSE #1
               '
               ' Display Part Serial Number
               '
               LOCATE 30, 1
               PRINT "Serial #: "; PartSerialNumber;
               '
               ' Display Date/Time of Inspection
               '
               LOCATE 30, 24
               PRINT "Date/Time: "; MID$(InspRecord.DTS, 5, 2); "/";
               PRINT MID$(InspRecord.DTS, 7, 2); "/"; LEFT$(InspRecord.DTS, 4); " ";
               PRINT MID$(InspRecord.DTS, 9, 2); ":";
               PRINT MID$(InspRecord.DTS, 11, 2); ":";
               PRINT MID$(InspRecord.DTS, 13, 2);
               '
               ' Display Signal Level
               '
               LOCATE 30, 60
               PRINT USING "Signal Level:###.#dB"; InspRecord.DB;
               '
               ' Display Inspection Point Status
               '
               FOR Scan = 1 TO 17
                    LOCATE 8 + Scan, 75
                    IF MID$(InspRecord.PF, Scan, 1) = "P" THEN
                         COLOR 2
                    ELSE
                         COLOR 4
                    END IF
                    PRINT USING "##"; Scan;
                    COLOR colorf
               NEXT
```

```
FOR Scan = 1 TO 17
    LOCATE 8 + Scan - 1, 72
    PRINT SPACE$(2);
    LOCATE 8 + Scan + 1, 72
    PRINT SPACE$(2);
    LOCATE 8 + Scan, 72
    COLOR 15
    PRINT ">>";
    COLOR colorf
    GOSUB DisplayScan
    DO
        DO
            key$ = INKEY$
        LOOP WHILE key$ = nul
        IF LEN(key$) = 2 THEN
            IF RIGHT$(key$, 1) = CHR$(72) THEN
                IF Scan > 1 THEN
                    Scan = Scan - 2
                    key$ = CR$
                ELSE
                    key$ = nul
                END IF
            ELSEIF RIGHT$(key$, 1) = CHR$(80) THEN
                IF Scan < 17 THEN
                    key$ = CR$
                ELSE
                    key$ = nul
                END IF
            END IF
        ELSEIF key$ = CHR$(27) THEN
            Scan = 17
            key$ = CR$
        END IF
    LOOP WHILE key$ <> CR$
NEXT
END IF
END IF
RETURN
```

```
DisplayScan:
     RejectString = InspRecord.RT
     GOSUB DrawGraphicScreen
     '
     '  Display Scan
     '
    FOR SubScript = 1 TO 200
          temp1 = ASC(MID$(InspRecord.DAT, (Scan - 1) * 200 + SubScript))
          temp2 = ASC(MID$(RejectString, SubScript, 1))
          LINE (SubScript * 2 - 1, 254)-(SubScript * 2, 254 - temp1 * 4), 2, B
          IF temp2 > 0 AND temp1 > temp2 THEN
               LINE (SubScript * 2 - 1, 254 - temp2 * 4)-(SubScript * 2, 254 - temp1 * 4), 4, B
          END IF
    NEXT
     GOSUB DrawRejectLine
     RETURN

DrawGraphicScreen:
     '
     '  Setup Graphic View Port
     '
     VIEW (120, 136)-(520, 390), 8, 1
     '
     '  Draw division lines
     '
    FOR i = 40 TO 360 STEP 40
          LINE (i, 0)-(i, 254), 14, , &HF0F0
    NEXT
     GOSUB DrawRejectLine
     '
     '  Lable Graphic Screen
     '
     LOCATE 26, 16
     PRINT "0        .2        .4        .6        .8        1.0"
     RETURN

DrawRejectLine:
     '
     '  Draw Reject line on screen
     '
    FOR SubScript = 1 TO 200
          temp = ASC(MID$(RejectString, SubScript, 1))
          IF temp > 0 THEN
               PSET (SubScript * 2 - 1, 254 - temp * 4), 3
               PSET (SubScript * 2, 254 - temp * 4), 3
          END IF
    NEXT
     RETURN
```

```
ErrorTrap:
     GOSUB ClearViewPort
     ecode = ERR
    -IF ecode = 57 OR ecode = 255 OR ecode = 253 THEN
          '
          '  Error code 57 is Device I/O error
          '  Error code 255 is program generated for a device timeout.
          '
          LOCATE 10, 7
          IF ecode = 253 THEN
               PRINT "The Panametrics EPOCH 2002 is not reading a signal (+100dB gain)"
          ELSE
               PRINT "There is a communication problem with the Panametrics EPOCH 2002"
          END IF
          LOCATE 12, 7
          PRINT "Press any key to reset the Robot to Home.  You will need to cycle"
          LOCATE 14, 7
          PRINT "the EPOCH 2002 off and back on again, and then restart the program."
          LOCATE 18, 30
          IF ecode = 57 THEN
               PRINT "Device I/O Error"
          ELSE
               PRINT "Device Timeout Error"
          END IF
          DO
               key$ = INKEY$
          LOOP WHILE key$ = nul
          GOSUB ResetTOT
          RobotActiveFlag = true
          FOR ErrorLoop = InspPoint TO 17
               '
               ' Move Robot to next InspPoint
               '
               RobotBitSubScript = 1
               GOSUB RobotBitSetTrue
               GOSUB RobotControl
               '
               '  Clear Robot Control Bit
               '
               GOSUB RobotBitSetFalse
               GOSUB RobotControl
               '
               '  Wait until Robot Clears [sets False] positon ready bit
               '
               DO
                    GOSUB Hex301Read
               LOOP WHILE Hex301(0)
               IF ErrorLoop < 17 THEN
                    '
                    '  Wait until Robot is in position
                    '
                    DO
                         GOSUB Hex301Read
                    LOOP WHILE NOT Hex301(0)
               END IF
          NEXT
          GOTO byebye
     ELSEIF ecode = 254 THEN
          '
          '  Error code 254 is program generated for a device timeout on Robot
          '
          LOCATE 10, 7
          PRINT "There is a communication problem with the Hitachi M5030 Robot"
          LOCATE 12, 7
          PRINT "Press any key to terminate program.  You will need to reset"
          LOCATE 14, 7
          PRINT "the Robot, if this problem continues, the interface or the Robot"
          LOCATE 16, 7
          PRINT "program may be the error or the Robot is not in REMOTE MODE."
          DO
               key$ = INKEY$
          LOOP WHILE key$ = nul
          GOTO byebye
     ELSE
          SELECT CASE ecode
               CASE 2: Error.Msg$ = "Syntax Error"
```

```
│    ├──────CASE 3: Error.Msg$ = "RETURN without GOSUB"
│    ├──────CASE 4: Error.Msg$ = "Out of DATA"
│    ├──────CASE 5: Error.Msg$ = "Illegal function Call"
│    ├──────CASE 6: Error.Msg$ = "Overflow"
│    ├──────CASE 7: Error.Msg$ = "Out of Memory"
│    ├──────CASE 9: Error.Msg$ = "Subscript out of Range"
│    ├──────CASE 10: Error.Msg$ = "Duplicate Defination"
│    ├──────CASE 11: Error.Msg$ = "Division by Zero"
│    ├──────CASE 13: Error.Msg$ = "Type Mismatch"
│    ├──────CASE 14: Error.Msg$ = "Out of String Space"
│    ├──────CASE 16: Error.Msg$ = "String Formula too complex"
│    ├──────CASE 19: Error.Msg$ = "No RESUME"
│    ├──────CASE 20: Error.Msg$ = "RESUME without error"
│    ├──────CASE 24: Error.Msg$ = "Device timeout"
│    ├──────CASE 25: Error.Msg$ = "Device fault"
│    ├──────CASE 52: Error.Msg$ = "Bad filename or number"
│    ├──────CASE 53: Error.Msg$ = "File not found"
│    ├──────CASE 54: Error.Msg$ = "Bad file mode"
│    ├──────CASE 55: Error.Msg$ = "File already open"
│    ├──────CASE 57: Error.Msg$ = "Device I/O error"
│    ├──────CASE 58: Error.Msg$ = "File already exists"
│    ├──────CASE 61: Error.Msg$ = "Disk full"
│    ├──────CASE 64: Error.Msg$ = "Bad file name"
│    ├──────CASE 67: Error.Msg$ = "Too many files"
│    ├──────CASE 68: Error.Msg$ = "Device unavailable"
│    ├──────CASE 70: Error.Msg$ = "Write protected disk"
│    ├──────CASE 71: Error.Msg$ = "Disk-drive door is open or no disk in drive"
│    ├──────CASE 72: Error.Msg$ = "Disk media error - disk is defective"
│    ├──────CASE 75: Error.Msg$ = "Path file access error"
│    ├──────CASE 76: Error.Msg$ = "Path not found"
│    ├──────CASE ELSE: Error.Msg$ = "Error code" + STR$(ecode)
│    └─END SELECT
│      LOCATE 15, (72 - LEN(Error.Msg$)) / 2
│      PRINT "ERROR - "; Error.Msg$
│    ┌─DO
│    │      PRINT "Press <RETURN> to continue or <ESC> to exit"
│    │      BEEP
│    │    ┌─DO
│    │    │      key$ = INKEY$
│    │    └─LOOP WHILE key$ = nul
│    └─LOOP WHILE key$ <> CR$ AND key$ <> ESC$
│      IF key$ = ESC$ THEN GOTO byebye
│      RESUME
└─END IF
  STOP '   ***This line should never be executed***
```

```
      '
      '  The Fn keys are only used for debugging
      '  Define Fn keys for toggling DIO8 output bits
      '
F1key:
     RobotBitSubScript = 1
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN

F2key:
     RobotBitSubScript = 2
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN

F3key:
     RobotBitSubScript = 3
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN

F4key:
     RobotBitSubScript = 4
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN

F5key:
     RobotBitSubScript = 5
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN

F6key:
     RobotBitSubScript = 6
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN

F7key:
     RobotBitSubScript = 7
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN

F8key:
     RobotBitSubScript = 8
     GOSUB RobotBitToggle
     GOSUB RobotControl
     RETURN
```

```
GetAdump:
     ErrorFlag = false
     '
     '  Average 4 A-Scans together in EPOCH 2002
     '
     GOSUB SendStar
     PRINT #6, "AVE=4"
     GOSUB ReadResponse
     IF NOT ErrorFlag THEN
          '
          '  Get ADUMP from EPOCH 2002
          '
          GOSUB SendStar
          PRINT #6, "ADUMP=?"
          ResponseLength = 613
          GOSUB ReadResponse
          IF LEN(ResponseString$) < ResponseLength THEN
               ERROR 255
          ELSE
               '
               '  Convert Adump data (hex) to Base10
               '
               Response$ = RIGHT$(ResponseString$, ResponseLength)
               SubScript = 1
               Position = 1
               DO
                    tmp$ = MID$(Response$, Position, 1)
                    Position = Position + 1
                    HexToBase10 = -1
                    IF (ASC(tmp$) >= 48 AND ASC(tmp$) <= 57) THEN
                         HexToBase10 = (ASC(tmp$) - 48) * 16
                    END IF
                    IF (ASC(tmp$) >= 65 AND ASC(tmp$) <= 70) THEN
                         HexToBase10 = (ASC(tmp$) - 55) * 16
                    END IF
                    IF HexToBase10 > -1 THEN
                         tmp$ = MID$(Response$, Position, 1)
                         Position = Position + 1
                         IF (ASC(tmp$) >= 48 AND ASC(tmp$) <= 57) THEN
                              HexToBase10 = HexToBase10 + (ASC(tmp$) - 48)
                         END IF
                         IF (ASC(tmp$) >= 65 AND ASC(tmp$) <= 70) THEN
                              HexToBase10 = HexToBase10 + (ASC(tmp$) - 55)
                         END IF
                         AdumpValue(SubScript) = HexToBase10
                         SubScript = SubScript + 1
                    END IF
               LOOP WHILE SubScript < 200
          END IF
     ELSE '  Com error
          ERROR 255
          STOP
     END IF
     RETURN

GetAdumpPeak:
     GOSUB GetAdump
     AdumpPeak = 0
     AdumpPeakPosition = 0
     FOR Position = SubscriptStart TO SubscriptEnd
          IF AdumpValue(Position) > AdumpPeak THEN
               AdumpPeak = AdumpValue(Position)
               AdumpPeakPosition = Position
          END IF
     NEXT
     RETURN
```

```
GetDateTime:
     DTS$ = DATE$
     DTS$ = MID$(DTS$, 7, 4) + LEFT$(DTS$, 2) + MID$(DTS$, 4, 2) + TIME$
     DTS$ = LEFT$(DTS$, 10) + MID$(DTS$, 12, 2) + RIGHT$(DTS$, 2)
     DateTimeString = DTS$
     RETURN

GetPartSerialNumber:
     '
     '  Get Part Serial Number
     '
     PartSerialNumber = nul
     LOCATE 15, 20
     COLOR 15
     PRINT "Enter Part Serial Number: [        ]";
     COLOR colorf
     LOCATE 15, 46
    ┌DO
    │    tmp$ = UCASE$(INKEY$)
    │   ┌IF tmp$ = nul THEN
    │   │     tmpASC = 0
    │   ├ELSE
    │   │     tmpASC = ASC(tmp$)
    │   └END IF
    │    LenPSN = LEN(PartSerialNumber)
    │   ┌IF (tmpASC >= 48 AND tmpASC <= 57) OR (tmpASC >= 65 AND tmpASC <= 90) THEN
    │   │    ┌IF LenPSN < 8 THEN
    │   │    │     '
    │   │    │     '  Letter or Number Character
    │   │    │     '
    │   │    │     PartSerialNumber = PartSerialNumber + tmp$
    │   │    │     LOCATE 15, 47
    │   │    │     PRINT PartSerialNumber; SPACE$(7 - LenPSN);
    │   │    ├ELSE
    │   │    │     '
    │   │    │     '  Already 8 Characters (Max)
    │   │    │     '
    │   │    │     BEEP
    │   │    └END IF
    │   ├ELSEIF tmpASC = 8 THEN
    │   │     '
    │   │     '  Backspace Character
    │   │     '
    │   │    ┌IF LenPSN > 1 THEN
    │   │    │     PartSerialNumber = LEFT$(PartSerialNumber, LenPSN - 1)
    │   │    ├ELSE
    │   │    │     PartSerialNumber = nul
    │   │    └END IF
    │   │     LOCATE 15, 47
    │   │     PRINT PartSerialNumber; SPACE$(9 - LenPSN);
    │   ├ELSEIF tmpASC <> 13 AND tmpASC <> 0 AND tmpASC <> 27 THEN
    │   │     '
    │   │     '  Invalid character
    │   │     '
    │   │     BEEP
    │   └END IF
    └LOOP WHILE tmpASC <> 13 AND tmpASC <> 27
     RETURN
```

```
Hex301Get:
    IF DIO8Flag THEN
        CmdValue1 = INP(&H301)
    ELSE
        CmdValue1 = DebugInHex301
    END IF
    InHex301 = CmdValue1
    RETURN

Hex301Read:
    GOSUB Hex301Get
    InHex301Temp = InHex301
    FOR Hex301ReadLoop = 7 TO 0 STEP -1
        IF InHex301Temp >= 2 ^ Hex301ReadLoop THEN
            Hex301(Hex301ReadLoop) = true
            InHex301Temp = InHex301Temp - 2 ^ Hex301ReadLoop
        ELSE
            Hex301(Hex301ReadLoop) = false
        END IF
    NEXT
    RETURN
```

```
InspectPart:
     GOSUB ClearViewPort
     LOCATE 13, 34
     PRINT "INSPECT PART";
     GOSUB GetPartSerialNumber
     IF PartSerialNumber = nul OR tmpASC = 27 THEN GOTO InspectReturn
     '
     '  Display Part Serial Number
     '
     LOCATE 30, 1
     PRINT "Serial #: "; PartSerialNumber;
     '
     '  Save File Header and initialize InspRecord
     '
     GOSUB GetDateTime
     InspRecord.PSN = PartSerialNumber
     InspRecord.DTS = DateTimeString
     InspRecord.PF = nul
     tmp$ = nul
     FOR SubScript = 1 TO 200
          tmp$ = tmp$ + CHR$(RejectTable(SubScript))
     NEXT
     InspRecord.RT = tmp$
     InspRecord.DAT = nul
     '
     '  Open COM Port
     '
     GOSUB ComOpen
     '
     '  ResetTOT for Robot timeout
     '
     GOSUB ResetTOT
     '
     '  Set Robot activity flag
     '
     RobotActiveFlag = true
     '
     '  Select Robot Program #1
     '
     RobotBitSubScript = 3
     GOSUB RobotBitSetTrue
     RobotBitSubScript = 4
     GOSUB RobotBitSetTrue
     GOSUB RobotControl
     '
     '  Start Robot Execution
     '
     RobotBitSubScript = 2
     GOSUB RobotBitSetTrue
     GOSUB RobotControl
     '
     '  Clear Robot Control Bits
     '
     RobotBitSubScript = 2
     GOSUB RobotBitSetFalse
     RobotBitSubScript = 3
     GOSUB RobotBitSetFalse
     RobotBitSubScript = 4
     GOSUB RobotBitSetFalse
     GOSUB RobotControl
     '
     '   Graphic Screen
     '
     GOSUB DrawGraphicScreen
     '
     '  Display Inspection Point Status
     '
     FOR InspPoint = 1 TO 17
          LOCATE 8 + InspPoint, 75
          PRINT USING "##"; InspPoint;
     NEXT
     '
     '  Clear PartDefectFlag
     '
     PartDefectFlag = false
```

```
'
' Inspect 17 InspPoints
'
FOR InspPoint = 1 TO 17
    '
    ' Clear InspRepeatFlag
    '
    InspRepeatFlag = false
    LOCATE 8 + InspPoint, 75
    COLOR 15
    PRINT USING "##"; InspPoint;
    COLOR colorf
    '
    ' Program Entry Point for reinspection
    '
InspRepeatEntryPoint:
    '
    ' Adjust Panametrics EPOCH 2002 for instpection
    '
    GOSUB ZeroTransducer
    '
    ' Set dB Level for inspection
    '
    GOSUB SendStar
    PRINT #6, USING "DB=###.#"; Decibel
    GOSUB ReadResponse
    '
    ' Get Inspection data dump
    '
    GOSUB GetAdump
    '
    ' Redraw Graphic Screen
    '
    GOSUB DrawGraphicScreen
    '
    ' Display Inspection Point #
    '
    LOCATE 30, 60
    PRINT USING "Inspection Point: ##"; InspPoint;
    '
    ' Is there a defect????
    '
    DefectFlag = false
    FOR SubScript = 1 TO 200
        LINE (SubScript * 2 - 1,254)-(SubScript * 2, 254 - AdumpValue(SubScript) * 4), 2, B
        IF RejectTable(SubScript) > 0 AND AdumpValue(SubScript) > RejectTable(SubScript)
                                  THEN
            LINE (SubScript * 2 - 1, 254 - RejectTable(SubScript) * 4)-(SubScript * 2, 254
                                  - AdumpValue(SubScript) * 4), 4, B
            DefectFlag = true
        END IF
    NEXT
    GOSUB DrawRejectLine
    '
    ' If defect found, reinspect InspPoint
    '
    IF NOT InspRepeatFlag AND DefectFlag THEN
        LOCATE 30, 33
        PRINT "Insp:           ";
        COLOR 15
        LOCATE 30, 39
        PRINT "Retesting";
        COLOR colorf
        InspRepeatFlag = true
        GOTO InspRepeatEntryPoint
    END IF
    '
    ' Move Robot to next InspPoint
    '
    RobotBitSubScript = 1
    GOSUB RobotBitSetTrue
    GOSUB RobotControl
    '
    ' Display & Save DefectFlag
    '
    IF DefectFlag THEN
```

```
            LOCATE 8 + InspPoint, 75
            COLOR 4
            PRINT USING "##"; InspPoint;
            COLOR colorf
            LOCATE 30, 33
            PRINT "Insp:           ";
            COLOR 4
            LOCATE 30, 39
            PRINT "FAILED";
            COLOR colorf
            IF InspPoint = 1 THEN
                InspRecord.PF = "F"
            ELSE
                InspRecord.PF = LEFT$(InspRecord.PF, InspPoint - 1) + "F"
            END IF
            '
            '  Set PartDefectFlag
            '
            PartDefectFlag = true
        ELSE
            LOCATE 8 + InspPoint, 75
            COLOR 2
            PRINT USING "##"; InspPoint;
            COLOR colorf
            LOCATE 30, 33
            PRINT "Insp:           ";
            COLOR 2
            LOCATE 30, 39
            PRINT "PASSED";
            COLOR colorf
            IF InspPoint = 1 THEN
                InspRecord.PF = "P"
            ELSE
                InspRecord.PF = LEFT$(InspRecord.PF, InspPoint - 1) + "P"
            END IF
        END IF
        '
        '  Convert Data to string and Save for data file
        '
        tmp$ = nul
        FOR SubScript = 1 TO 200
            tmp$ = tmp$ + CHR$(AdumpValue(SubScript))
        NEXT
        IF InspPoint = 1 THEN
            InspRecord.DAT = tmp$
        ELSE
            InspRecord.DAT = LEFT$(InspRecord.DAT, (InspPoint - 1) * 200) + tmp$
        END IF
        '
        '  Clear Robot Control Bit
        '
        GOSUB RobotBitSetFalse
        GOSUB RobotControl
        '
        '  Wait until Robot Clears [sets False] positon ready bit
        '
        DO
            GOSUB ClockDisplay
            GOSUB Hex301Read
        LOOP WHILE Hex301(0)
    NEXT
    '
    '  Clear Robot activity flag
    '
    RobotActiveFlag = false
    '
    '  Close COM Port
    '
    GOSUB ComClose
    '
    '  Save Inspection Signal Level
    '
    InspRecord.DB = Decibel
    '
    '  Open Data File
    '
```

```
DataFile$ = RTRIM$(PartSerialNumber) + ".INS"
OPEN DataFile$ FOR BINARY AS #1
'
'  Save Data to file
'
PUT #1, 1, InspRecord
```

```
'
'  Close Data File
'
CLOSE #1
'
'  Rerun Part?
'
IF PartDefectFlag THEN
     LOCATE 28, 30
     COLOR 15
     PRINT "Rerun Part [y/N]: ";
     COLOR colorf
     DO
          tmp$ = UCASE$(INKEY$)
          IF tmp$ = CR$ THEN tmp$ = "N"
     LOOP WHILE tmp$ <> "Y" AND tmp$ <> "N"
     PRINT tmp$;
     IF tmp$ = "Y" THEN
          LOCATE 28, 30
          PRINT SPACE$(20);
          LOCATE 30, 38
          PRINT SPACE$(11);
          LOCATE 30, 78
          PRINT SPACE$(2);
          RerunFlag = true
     ELSE
          RerunFlag = false
     END IF
ELSE
     RerunFlag = false
END IF
IF RerunFlag THEN GOTO InspectPart
InspectReturn:
RETURN
```

```
IntroScreen:
   ┌FOR i = 1 TO LEN(s$)
   │    ┌IF VAL(MID$(s$, i, 1)) THEN
   │    │     MID$(s$, i, 1) = CHR$(178)
   │    ├ELSE
   │    │     MID$(s$, i, 1) = CHR$(32)
   │    └END IF
   └NEXT
    GOSUB ScreenHeader
    TIMER STOP
    row = 11
    column = 5
    COLOR colorf - 2
   ┌FOR i = 0 TO 11
   │     LOCATE row + i, column
   │     PRINT MID$(s$, i * 70 + 1, 70);
   └NEXT
    COLOR colorf
    TIMER ON
    LOCATE 25, 14
    PRINT "Ultrasonic Inspection Cell Control Software (UICCS)";
    COLOR 15
    LOCATE 27, 24
    PRINT "<< Press any Key to Continue >>";
    COLOR colorf
    row = 11
    column = 5
    RANDOMIZE TIMER
    IntroScreenFlag = true
    IntroScreenColor = 12
   ┌DO
   │    ┌IF colorf = 14 AND IntroScreenColor <> LastIntroScreenColor THEN
   │    │     TIMER STOP
   │    │     COLOR IntroScreenColor
   │    │    ┌FOR i = 0 TO 11
   │    │    │     LOCATE row + i, column
   │    │    │     PRINT MID$(s$, i * 70 + 1, 70);
   │    │    └NEXT
   │    │     LastIntroScreenColor = IntroScreenColor
   │    │     COLOR colorf
   │    │     TIMER ON
   │    └END IF
   │     tmp$ = INKEY$
   └LOOP WHILE tmp$ = nul
    IntroScreenFlag = false
    GOSUB ScreenHeader
    RETURN

InvalidEntry:
    LOCATE 21, column + 17
    PRINT "***Invalid Entry***";
    BEEP
    SLEEP 2
    LOCATE 21, column + 17
    PRINT "                    ";
    ErrorFlag = true
    RETURN

NoVGA:
    '
    '  Error Routine for computers without VGA graphics
    '
    PRINT "This program requires a VGA graphics card to run."
    PRINT
    GOTO byebyeend
    '
    '  This program should never process the next two lines
    '
    RESUME
    RETURN
```

```
ReadResponse:
     GOSUB ResetTOT
     ResponseString$ = nul
     '
     '  Wait for EOB$ character or timeout
     '
    ┌DO
    │     ResponseString$ = ResponseString$ + INPUT$(LOC(6), #6)
    │     IF INSTR(ResponseString$, EOB$) > 0 THEN EXIT DO
    └LOOP WHILE TimeOutTimer < 2
     '
     '  Check for timeout
     '
    ┌IF INSTR(ResponseString$, EOB$) = 0 THEN
    │     ERROR 255
    │     STOP
    └END IF
     RETURN

ReportMenu:
     COLOR colorf
     ReportMenuSelection = false
    ┌DO WHILE ReportMenuSelection <> 4
    │     GOSUB ClearViewPort
    │     s$ = "R E P O R T   M E N U"
    │     LOCATE 10, 40 - LEN(s$) / 2
    │     PRINT s$;
    │     column = 26
    │     LOCATE 13, column
    │     PRINT "1. Print Inspection Summary";
    │     LOCATE 15, column
    │     PRINT "2. Print Today's Inspection Summary";
    │     LOCATE 17, column
    │     PRINT "3. <<Unavailable>>";
    │     LOCATE 19, column
    │     PRINT "4. Return to Main Menu";
    │     LOCATE 21, column
    │     COLOR 15
    │     PRINT "Enter Selection: ";
    │     COLOR colorf
    │     PRINT CHR$(178);
    │     LOCATE 21, column + 17
    │     GOSUB ResetTOT
    │     '
    │     '  Get selection or force return to main menu
    │     '
    │    ┌DO
    │    │     ReportMenuSelection$ = INKEY$
    │    │     IF TimeOutTimer > 60 THEN ReportMenuSelection$ = "4"     '    Force menu exit
    │    └LOOP WHILE ReportMenuSelection$ = nul
    │     PRINT ReportMenuSelection$;
    │     selection = VAL(ReportMenuSelection$)
    │    ┌IF selection > 0 AND selection < 5 THEN
    │    │     IF SoundFlag THEN SOUND 1000, .5
    │    │    ┌SELECT CASE selection
    │    │    ├──────CASE 1
    │    │    │           ReportSummaryTodayFlag = false
    │    │    │           GOSUB ReportSummary
    │    │    ├──────CASE 2
    │    │    │           ReportSummaryTodayFlag = true
    │    │    │           GOSUB ReportSummary
    │    │    ├──────CASE 3
    │    │    │           REM GOSUB
    │    │    ├──────CASE 4
    │    │    │           EXIT DO
    │    │    └END SELECT
    │    │     ReportMenuSelection = 4
    │    │     selection = true '  force continued looping
    │    ├ELSE
    │    │     GOSUB InvalidEntry
    │    └END IF
    └LOOP
     IF SoundFlag THEN SOUND 2000, .5
     RETURN
```

```
ReportSummary:
     LOCATE 25, 33
     PRINT "<<Processing>>";
     GOSUB ReportSummaryInit
     GOSUB GetDateTime
     OPEN "UICCS.PRT" FOR OUTPUT AS #5
     LineNumber = 1
     PageNumber = 1
     DateTimePrint$ = DATE$ + "   " + TIME$
     PRINT #5, ""
     FOR RecordNumber = 1 TO MaxRecordNumber
          GET #3, RecordNumber, Sort1
          Filename$ = RTRIM$(Sort1) + ".INS"
          OPEN Filename$ FOR RANDOM ACCESS READ AS #1 LEN = 3643
          GET #1, 1, InspRecord
          CLOSE #1
          IF (ReportSummaryTodayFlag AND LEFT$(InspRecord.DTS, 8) = LEFT$(DateTimeString, 8)) OR
                                        (NOT ReportSummaryTodayFlag) THEN
               IF LineNumber = 60 THEN
                    PRINT #5, CHR$(12)
                    LineNumber = 1
               END IF
               IF LineNumber = 1 THEN
                    '
                    '  Print Report Header
                    '
                    PRINT #5, ""
                    PRINT #5, TAB(22); "ULTRASONIC INSPECTION REPORT SUMMARY"; TAB(70);
                    PRINT #5, USING "Page: ###"; PageNumber
                    PRINT #5, TAB(30); DateTimePrint$
                    IF ReportSummaryFlag THEN
                         PRINT #5, TAB(30); "Today's Records Only"
                    ELSE
                         PRINT #5, TAB(31); "Cumulative Records"
                    END IF
                    PRINT #5, ""
                    PRINT #5, ""
                    PRINT #5, "Serial #                    dB    Inspection Pts
                                        Summary"
                    PRINT #5, "--------------------------------------------------------------
                                        ---------"
                    LineNumber = 10
                    PageNumber = PageNumber + 1
               END IF
               PRINT #5, InspRecord.PSN; TAB(11); TAB(28);
               PRINT #5, USING "##.#  "; InspRecord.DB;
               PassFailFlag = false
               FOR i = 1 TO 17
                    tmp$ = MID$(InspRecord.PF, i, 1)
                    IF tmp$ = "F" THEN PassFailFlag = true
                    PRINT #5, tmp$; " ";
               NEXT
               PRINT #5, TAB(73);
               IF PassFailFlag THEN
                    PRINT #5, "FAIL"
               ELSE
                    PRINT #5, "Pass"
               END IF
               LineNumber = LineNumber + 1
          END IF
     NEXT
     PRINT #5, CHR$(12);
     CLOSE
     IF PageNumber = 1 AND LineNumber = 1 THEN
          LOCATE 25, 20
          PRINT "Request Terminated - No matching Records";
          IF SoundFlag THEN
               FOR Scan = 1 TO 20
                    SOUND 1300, .4
                    SOUND 1000, .4
                    SOUND 700, .4
               NEXT
          END IF
```

```
├─ELSE
│       SHELL "COPY UICCS.PRT PRN:"
└─END IF
    KILL "UICCS.TMP"
    KILL "UICCS.PRT"
    RETURN

ReportSummaryInit:
    '
    '  Write Directory to File
    '
    SHELL "DIR *.INS > UICCS.DIR"
    '
    '  Read in directory and save filenames (serial numbers)
    '
    OPEN "UICCS.DIR" FOR INPUT AS #2
    OPEN "UICCS.TMP" FOR RANDOM AS #3 LEN = 8
    RecordNumber = 0
┌─DO WHILE NOT EOF(2)
│       LINE INPUT #2, tmp$
│   ┌─IF MID$(tmp$, 10, 3) = "INS" THEN ' filename extension
│   │       RecordNumber = RecordNumber + 1
│   │       PutPSN = tmp$
│   │       PUT #3, RecordNumber, PutPSN
│   └─END IF
└─LOOP
    CLOSE #2
    KILL "UICCS.DIR"
    MaxRecordNumber = RecordNumber
    '
    '  Sort Filenames (Serial Numbers)
    '
┌─IF MaxRecordNumber > 5000 THEN
│       '
│       '  Sort to Disk
│       '
│   ┌─DO
│   │       SortFlag = false
│   │   ┌─FOR RecordNumber = 1 TO MaxRecordNumber - 1
│   │   │       GET #3, RecordNumber, Sort1
│   │   │       GET #3, RecordNumber + 1, Sort2
│   │   │   ┌─IF Sort1 > Sort2 THEN
│   │   │   │       PUT #3, RecordNumber, Sort2
│   │   │   │       PUT #3, RecordNumber + 1, Sort1
│   │   │   │       SortFlag = true
│   │   │   └─END IF
│   │   └─NEXT
│   └─LOOP WHILE SortFlag = true
├─ELSE
│       '
│       '  Sort in memory
│       '
│   ┌─FOR RecordNumber = 1 TO MaxRecordNumber
│   │       GET #3, RecordNumber, Sort(RecordNumber)
│   └─NEXT
│   ┌─DO
│   │       SortFlag = false
│   │   ┌─FOR RecordNumber = 1 TO MaxRecordNumber - 1
│   │   │   ┌─IF Sort(RecordNumber) > Sort(RecordNumber + 1) THEN
│   │   │   │       SWAP Sort(RecordNumber), Sort(RecordNumber + 1)
│   │   │   │       SortFlag = true
│   │   │   └─END IF
│   │   └─NEXT
│   └─LOOP WHILE SortFlag = true
│   ┌─FOR RecordNumber = 1 TO MaxRecordNumber
│   │       PUT #3, RecordNumber, Sort(RecordNumber)
│   └─NEXT
└─END IF
    RETURN
```

```
ResetTOT:
     TimeOutTimer = 0
     RETURN

RobotBitSetFalse:
     RobotBit(RobotBitSubScript) = false
     RETURN

RobotBitSetTrue:
     RobotBit(RobotBitSubScript) = true
     RETURN

RobotBitToggle:
    ┌IF RobotBit(RobotBitSubScript) THEN
    │     GOSUB RobotBitSetFalse
    ├ELSE
    │     GOSUB RobotBitSetTrue
    └END IF
     RETURN

RobotControl:
     '
     '  Calculate CmdValue for controlling DIO8-P interface board
     '
     CmdValue = 0
     IF RobotBit(1) THEN CmdValue = CmdValue + 1
     IF RobotBit(2) THEN CmdValue = CmdValue + 2
     IF RobotBit(3) THEN CmdValue = CmdValue + 4
     IF RobotBit(4) THEN CmdValue = CmdValue + 8
     IF RobotBit(5) THEN CmdValue = CmdValue + 16
     IF RobotBit(6) THEN CmdValue = CmdValue + 32
     IF RobotBit(7) THEN CmdValue = CmdValue + 64
     IF RobotBit(8) THEN CmdValue = CmdValue + 128
     '
     '  Make sure .3 seconds have elapsed since last OUT &H300
     '     Note: This is required so that the HITACHI M5030 has
     '           time to read the control line
     '
    ┌DO
    └LOOP UNTIL RobotDelayTimer + .3 < TIMER OR TIMER < RobotDelayTimer
     '
     '  Send control signal to HITACHI M5030 via DIO8-P interface board
     '
     OUT &H300, CmdValue
     '
     '  Save time for robot delay loop
     '
     RobotDelayTimer = TIMER
     GOSUB DebugDisplayIO
     RETURN
```

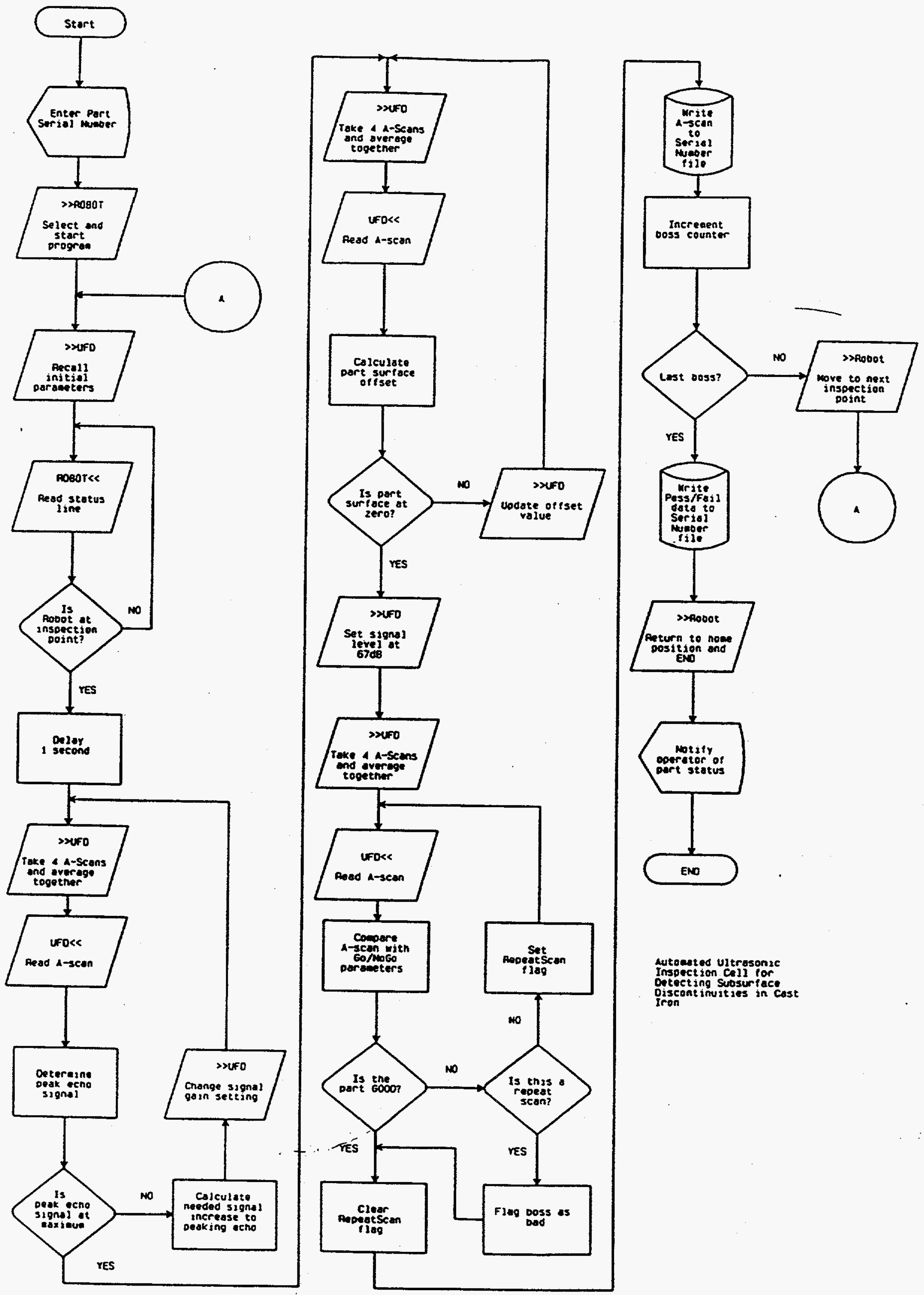
```
ScreenHeader:
     COLOR colorf
     CLS 0
     IF DebugFlag THEN
          LOCATE 1, 70
          PRINT DisplayBoxTop$;
          LOCATE 2, 70
          PRINT DisplayBoxMiddle$;
          LOCATE 3, 70
          PRINT DisplayBoxMiddle$;
          LOCATE 4, 70
          PRINT DisplayBoxMiddle$;
          LOCATE 5, 70
          PRINT DisplayBoxBottom$;
          GOSUB DebugDisplayIO
     END IF
     LOCATE 1, 27
     PRINT "University of Northern Iowa";
     LOCATE 2, 23
     PRINT "Department of Industrial Technology";
     LOCATE 3, 30
     PRINT "Metal Casting Center";
     LOCATE 4, 27
     PRINT "Cedar Falls, IA  50614-0178";
     GOSUB ClockDisplay
     LOCATE 6, 21
     PRINT "Copyright 1991-1992, All Rights Reserved";
     LOCATE 7, 34
     PRINT "Version 0.51";
     TIMER ON
     RETURN

SendStar:
     ResponseLength = 1
     '
     '  Clear COM Input Buffer
     '
     IF LOC(6) > 0 THEN Response$ = INPUT$(LOC(6), #6)
     '
     '  Send attention character [*]
     '
     PRINT #6, Star;
     '
     '  Wait for a response w/timeout
     '
     GOSUB ResetTOT
     DO
          IF TimeOutTimer > 2 THEN EXIT DO
     LOOP WHILE LOC(6) < ResponseLength
     '
     '  Read COM Buffer
     '
     ResponseStar$ = INPUT$(LOC(6), #6)
     '
     ' Is acknowledgement correct [*]
     '
     IF ResponseStar$ <> "*" THEN
          IF ErrorFlag THEN
               ERROR 255
               STOP
          ELSE ' Try again
               ErrorFlag = true
               GOSUB SendStar ' Recursive call
               '
               '  Clear ErrorFlag if second try succeeds
               '
               ErrorFlag = false
          END IF
     END IF
     RETURN
```

```
ZeroTransducer:
     DB = 0
     AdumpPeak = 0
     ReaddB = 0
     '
     '  Reset EPOCH 2002 display
     '
     GOSUB SendStar
     PRINT #6, "DISP=S"
     GOSUB ReadResponse
     GOSUB SendStar
     PRINT #6, "RCL=1"
     GOSUB ReadResponse
     GOSUB SendStar
     PRINT #6, "DISP=G"
     GOSUB ReadResponse
     '
     '  Set starting subscript range
     '
     SubscriptStart = 1
     SubscriptEnd = 200
     '
     '  Wait until Robot is in position
     '
     DO
          GOSUB Hex301Read
     LOOP WHILE NOT Hex301(0)
     '
     '  Wait 1 Second for robot to settle (it bounces at the end of motion)
     '
     SettleTimer! = TIMER
     DO
     LOOP UNTIL SettleTimer! + 1 < TIMER OR TIMER < SettleTimer!
     '
     '  Zero Transducer
     '
     DO
          GOSUB dBchange
          GOSUB GetAdumpPeak
          IF AdumpPeak > 20 THEN
               SubscriptStart = AdumpPeakPosition - 10
               IF SubscriptStart < 1 THEN SubscriptStart = 1
               SubscriptEnd = AdumpPeakPosition + 10
               IF SubscriptEnd > 200 THEN SubscriptEnd = 200
          END IF
          GOSUB dBcalculate
     LOOP WHILE AdumpPeak < 63 OR DB > 3 OR DB < 0
     GOSUB SendStar
     PRINT #6, "DB=?"
     GOSUB ReadResponse
     FrontdB = VAL(MID$(ResponseString$, INSTR(ResponseString$, CHR$(10) + "DB=") + 4))
     '
     '  Left justify Top Surface
     '
     GOSUB SendStar
     PRINT #6, "ZERO=?"
     GOSUB ReadResponse
     FrontZeroOffset = VAL(MID$(ResponseString$, INSTR(ResponseString$, CHR$(10) + "ZERO=") + 6))
     SubscriptStart = 1
     LOOPFlag = false
     DO
          '
          '  SET ZERO OFFSET
          '
          IF LoopFlag THEN
               GOSUB SendStar
               IF FrontZeroOffset < 100 THEN
                    s$ = "ZERO=##.##"
               ELSE
                    s$ = "ZERO=###.#"
               END IF
               PRINT #6, USING s$; FrontZeroOffset
               GOSUB ReadResponse
               GOSUB GetAdumpPeak
```

```
    ├─ELSE
    │     LoopFlag = true
    └─END IF
      SubscriptEnd = AdumpPeakPosition + 10
      IF SubscriptEnd > 200 THEN SubscriptEnd = 200
    ┌─SELECT CASE AdumpPeakPosition
    ├───────CASE IS > 3
    │           FrontZeroOffset = FrontZeroOffset + AdumpPeakPosition / 19
    ├───────CASE ELSE
    │           FrontZeroOffset = FrontZeroOffset + .1
    └─END SELECT
└─LOOP WHILE AdumpPeakPosition > 1
  RETURN
```

APPENDIX B
CELL OPERATION FLOW CHART


Start
Enter Part Serial Number
>>ROBOT Select and start program
A
>>UFD Recall initial parameters
ROBOT<< Read status line
Is Robot at inspection point?
NO
YES
Delay 1 second
>>UFD Take 4 A-Scans and average together
UFD<< Read A-scan
Determine peak echo signal
Is peak echo signal at maximum
NO
YES
Calculate needed signal increase to peaking echo
>>UFD Change signal gain setting
>>UFD Take 4 A-Scans and average together
UFD<< Read A-scan
Calculate part surface offset
Is part surface at zero?
NO
>>UFD Update offset value
YES
>>UFD Set signal level at 67dB
>>UFD Take 4 A-Scans and average together
UFD<< Read A-scan
Compare A-scan with Go/NoGo parameters
Is the part GOOD?
NO
Is this a repeat scan?
NO
Set RepeatScan flag
YES
Flag boss as bad
YES
Clear RepeatScan flag
Write A-scan to Serial Number file
Increment boss counter
Last boss?
NO
>>Robot Move to next inspection point
A
YES
Write Pass/Fail data to Serial Number file
>>Robot Return to home position and END
Notify operator of part status
END
Automated Ultrasonic Inspection Cell for Detecting Subsurface Discontinuities in Cast Iron