NUREG/CR-4365

Design and Development of a Special Purpose SAFT System for Nondestructive Evaluation of Nuclear Reactor Vessels and Piping Components

Prepared by S. Ganapathy, B. Schmult, W. S. Wu,
T. G. Dennehy, N. Moayeri, P. Kelly

Department of Electrical and Computer Engineering
The University of Michigan

Prepared for
U.S. Nuclear Regulatory Commission

NUREG/CR-4365
R5

Design and Development of a Special Purpose SAFT System for Nondestructive Evaluation of Nuclear Reactor Vessels and Piping Components

Manuscript Completed: July 1985
Date Published: August 1985

Prepared by
S. Ganapathy, B. Schmult, W. S. Wu,
T. G. Dennehy, N. Moayeri, P. Kelly

Department of Electrical and Computer Engineering
The University of Michigan
Ann Arbor, MI 48109-1092

Prepared for
Division of Engineering Technology
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
NRC FIN A4047

Abstract

This report describes the design details of a special purpose system for real-time nondestructive evaluation of reactor vessels and piping components. The system consists of several components and the report presents the results of the research aimed at the design of each component and recommendations based on the results. One major component of the NDE system, namely the real-time SAFT processor, was designed with sufficient details to enable the fabrications of a prototype by GARD Inc. under a subcontract from The University of Michigan and the report includes their results and conclusions.

TABLE OF CONTENTS

LIST OF FIGURES

Foreword

This technical report describes research work performed by the Ultrasonic Imaging Laboratory for the United States Nuclear Regulatory Commission (NRC) in fulfillment of NRC contract No. NRC-04-75-182, "Ultrasonic Imaging Techniques for Real-Time In-service Inspection of Nuclear Power Reactors."

The purpose of this multi-year contract (1980-1983) was to develop a field implementable real-time system based on Synthetic Aperture Focusing Techniques (SAFT) that had been developed previously at The University of Michigan. To accomplish this objective, the first year's effort was spent on implementation of improved display techniques and on the analysis of potential bottlenecks in the development of a field implementable real-time system. The second year's effort concentrated on the detailed analysis of processing requirements and initial design of a SAFT processor. The results of the first two years' efforts are detailed in the NRC reports NUREG/CR-2154 (October 1981) and NUREG/CR-2703 (January 1983) respectively.

The final year was devoted to the detailed design of the special purpose SAFT processor and an overall design of a field implementable SAFT system. The design specifications were subsequently handed over to GARDine as part of a subcontract to enable them to build a prototype. This prototype was successfully fabricated by late 1983 and met the overall objectives of the design. This report, the third and final annual report, discusses the design details and the results of the construction of the SAFT processor and includes recommendations on the other components of a field implementable NDE system based on SAFT.

The work was performed in the Ultrasonic Imaging Laboratory, Department of Electrical and Computer Engineering, The University of Michigan, Ann Arbor, Michigan. The project was performed under the technical direction of Prof. S. Ganapathy. The project monitor was Dr. J. Muscara of the U. S. Nuclear Regulatory Commission.

This technical report covers work done primarily over the period January 1, 1982 to September 30, 1983. The details in this report are based heavily on the results reported in earlier reports, primarily NUREG/CR-2154 and NUREG/CR-2703. Details of prior work on all aspects of this research are available in one or more of the following NRC technical reports:

NUREG-0007-1: January 1976
NUREG-0007-2: September 1977
NUREG/CR-0135: May 1978
NUREG/CR-0581: January 1979
NUREG/CR-0909: September 1979
NUREG/CR-2154: October 1981
NUREG/CR-2703: January 1983.

1. Introduction

A field implementable SAFT system as envisaged by us consists of four major parts: Scanning, Processing, Display and Automated decision making (See Figure 1-1). It was identified, fairly early on, that real-time SAFT processing is a bottleneck and efforts were devoted to analyzing the bottleneck and designing solutions to overcome it. It was also felt that improved display techniques were badly needed to help an NDE technician to interpret the volumetric data put out by a SAFT processor. In this report we augment our previously published techniques to enable an operator to visually inspect a large volume of data in real-time. An attempt was also made to use well known edge-detection techniques to automate the decision making involved in flaw sizing and interpretation. This study could not be completed in its entirety but the report includes our findings and recommendations of edge-detectors for ultrasonic NDE images.

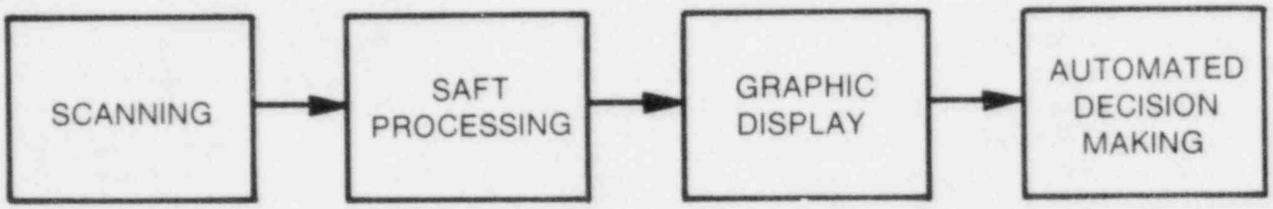


FIGURE 1-1 Different Phases of SAFT-UT

Since SAFT processing involves a tremendous computational load, we also investigated some flaw detection techniques to be used as a front end to identify those areas that needed the more elaborate and expensive SAFT processing. It was felt that good flaw detection techniques coupled with such a system as described in this report would be an ideal solution for routine inspection of reactor vessels and piping components.

Section 2 of this report discusses the overall system design including scanning, and Section 3 covers the design details of the special-purpose SAFT processor. In Section 4, we include a brief description of two additional Computer Graphics techniques. One, knows as WALKTHRU, is for real-time inspection and the other technique using real-time rotation of perspective images is recommended for detailed off-line analysis of volumetric flaws. Section 5 is devoted to the study of various edge detectors on SAFT processed images and the results obtained with them. It is our conclusion that a simple edge operator such as the SOBEL operator performs just as well as the more complex ones for our requirements. However, more extensive experimentation is needed to confirm our initial finding. Section 6 reports on our approach to flaw detection and classification in the presence of grain noise. It must be understood while reading Section 6 that this represents ongoing work and the results and suggestions are preliminary. Finally in Section 7 we present the details of the SAFT processor prototype and the results obtained with it. We conclude this report with a brief summary of the conclusions of our 3-year study on a field implementable SAFT system.

2. Real-time Field Implementable SAFT System

2.1. Introduction

This section is a partial account of progress in research underway to develop an advanced real-time nondestructive evaluation system for the inspection of nuclear reactor components. This project was undertaken because of the need to develop inspection methods that are more reliable than conventional techniques. The Ultrasonic Imaging Laboratory has been developing an advanced inspection system based on Synthetic Aperture Focusing Techniques, or SAFT. Previous analysis has shown that such a SAFT based system can give an order of magnitude improvement in resolution over conventional inspection methods (see earlier reports).

A large amount of research has been put into the analysis of the problems in a SAFT system and into finding possible solutions. Most of the results of these efforts are presented in a recent technical report [NUREG CR/2703]. These results show that a real-time SAFT-based inspection system is viable for field use. One of the major conclusions was that a special purpose processing system is required to achieve real-time SAFT processing speeds and that given such a processor, a field system can be constructed. This report also presented several possible architectures for a special processor and a new technique called software machining that is used to decrease the computational complexity in SAFT computations. With these results in place the Ultrasonic Imaging Laboratory has been able to design an actual SAFT-based inspection system.

This report presents the system design. The next section describes a complete initial field inspection system. Such a system will demonstrate the capabilities of SAFT on the major parts of power reactors. Section 3 gives the detailed design of the special purpose SAFT processor and can be used as a basis for fabrication of a prototype. Since it is desirable to construct a prototype SAFT processor before building an entire system a plan for building a scaled down processor is presented in Section 4, along with details on how it can be tested and demonstrated with a general purpose minicomputer. The final section contains recommendations for future field systems.

For background information on SAFT concepts and for a history of the research work done at the Ultrasonic Imaging Laboratory the reader is referred to the Annual and Technical Reports listed in the Foreword.

2.2. Description of an Initial Field Inspection System

The ultimate goal of this project is to produce a complete system for real-time inspection in the field. This section presents a high-level (details omitted) design of an initial system for field use. Since the first step in development is the prototyping of a SAFT processor, specification of the details of a complete system will be delayed. The goal of an initial field system is to prove the viability of SAFT techniques on real reactor components. Therefore the system will be simple and easy to implement and will inspect only a subset of the components that future systems might handle.

2.2.1. Inspection Capabilities

This section describes the inspection capabilities of the proposed initial system and gives examples of how it might be used in the field.

2.2.1.1. Inspectable Components

There are three major components of nuclear reactors to be scanned:

1. Longitudinal and Circumferential welds on the pressure vessel.
2. Welds of both types of piping components. Note that pipes are assumed to be cylindrical.
3. Nozzles. This term means any part not included by 1 and 2. In particular, a nozzle is where pipes join the vessel wall.

The initial field versions will not support inspection of nozzles. All design specifications discussed below will only apply to vessel and pipe welds.

2.2.1.2. Inspectable Volumes

The volume of a weld region that can be inspected at one time is limited. All welds can be imaged on their entire length (no restrictions) since image planes will be perpendicular to this direction. The maximum weld width will be no less than 18 inches. This permits inspection to 0.75T from either side of the weld center (where T is the through-wall thickness) in almost all cases. The maximum depth that can be inspected will be no less than 12 inches. Note that if the wall is too thick, any contiguous subset under 12 inches thick can be inspected on a single pass.

The stated minimum values of the limits will be the actual limits when scanning at 2.25 MHz. Use of lower frequencies will permit larger volumes to be inspected.

These limits also assume normal scanning procedures. Other procedures may degrade the system's capacity as discussed below.

2.2.1.3. Resolution and Detection Capabilities

The precise flaw detecting capabilities are not being specified at this time. The system will, however, locate flaws with adequate resolution to satisfy existing inspection codes. In particular at 2.25 MHz the resolution of SAFT is better than 0.1 inch. Codes for vessel welds require location of flaws larger than 2% of the through-wall thickness, which is larger than 0.1 inch for typical walls.

The detection capabilities will remain adequate for the full volume under inspection.

2.2.1.4. Scanning Modes

The normal scanning mode is defined below:

1. The transducer beam is normal to the front surface.
2. The frequency is not greater than 2.25 MHz.
3. The processing angle is not greater than 60 degrees.
4. The raster scanning is done on a $\lambda \times \lambda$ grid where λ is the wavelength of the transducer.

The above specifications on inspectable volumes hold when these parameters are in effect. Decreasing the frequency will result in a larger inspectable volume. Larger angles and frequencies may or may not be supported and would require smaller volumes. A different grid size may or may not be allowed.

Squint mode scanning (transducer not normal to the surface) will also be supported over the full range of frequencies and included angles. Squint includes use of shear waves. Note that 1.0 MHz will be the standard transducer frequency when scanning in shear mode.

2.2.1.5. Aperture Profiles

All processing will use two-dimensional apertures, for three-dimensional imaging. For processing non-squint data, an expanding circular aperture will be used. An optimal aperture for squint data is still under study. When a complete A-scan (points for all depths for a given scanner location) is processed, these apertures will form a cone, known as the aperture profile. Because of attenuation and processing considerations, this cone will not be permitted to grow normally. Aperture radii will be clipped to some value, which is expected to be three inches. The profile will be assumed not to shrink at greater depths.

2.2.1.6. Performance

All of the computer related components (SAFT processor, signal processor, storage and display) will produce processed data ready for analysis at a rate not less than 200 cubic inches per minute for the normal conditions described above. Lower frequencies and angles may permit higher rates.

Note that the actual rate will also depend on the scanning mechanism which may not operate this fast.

2.2.1.7. Display and Analysis

The principal method of display will be a walk-through method using pseudo-color images. The images can be displayed as scanning is taking place, but since a weld may take many hours to scan, display can also be done from stored data. The display system will be capable of

operating at rates sufficient for the effective implementation of walk-through independently of the processing rate.

Other means of display and analysis may be available such as shading and contour plotting. Such methods will be software driven and will operate on stored data. They will probably not operate in real time.

2.2.1.8. Data Storage

As processed data is generated it will be stored, probably on a high-speed disk drive system. Most display and analysis programs will then operate from this stored data. Once analysis is finished, the data will be saved in archive form on high-density magnetic tape. Such data can be retrieved in the future for re-analysis and comparison to current data.

2.2.1.9. Typical Use

This section gives as an example a typical procedure to inspect a weld or weld region. The operator decides on the exact region to scan and enters such information into the system control computer from a terminal. The scanner is then positioned to the starting point. SAFT processing parameters are entered and are then transferred to the SAFT processor. The scan is started and runs without operator intervention until the entire region is scanned. The processed data is now on disk. If desired, it can be saved permanently on tape at this point. The WALKTHRU program is then used to display the entire data set on the color display system. Interesting spots can be inspected from different angles (although not in real time). If other display devices are available, such as plotting, these can be used on operator-selected regions. Then, if desired, the scanner can be started over regions of interest for a higher resolution scan.

2.2.2. System Description

A block diagram of the complete system appears in Figure 2.1. It consists of a scanner, a set of special processors, a display system and a data storage system, all under control of a general purpose minicomputer. All parts except the scanner subsystem can be located away from the reactor. For normal operations an operator only needs access to the terminals.

The following sections describe the various components.

2.2.2.1. System Control Computer

The center of the inspection system is the System Control Computer (SCC). The SCC is a complete 16 or 32 bit minicomputer system. All the high-performance processors are attached to and controlled by it. All operator interaction with the inspection system is done through

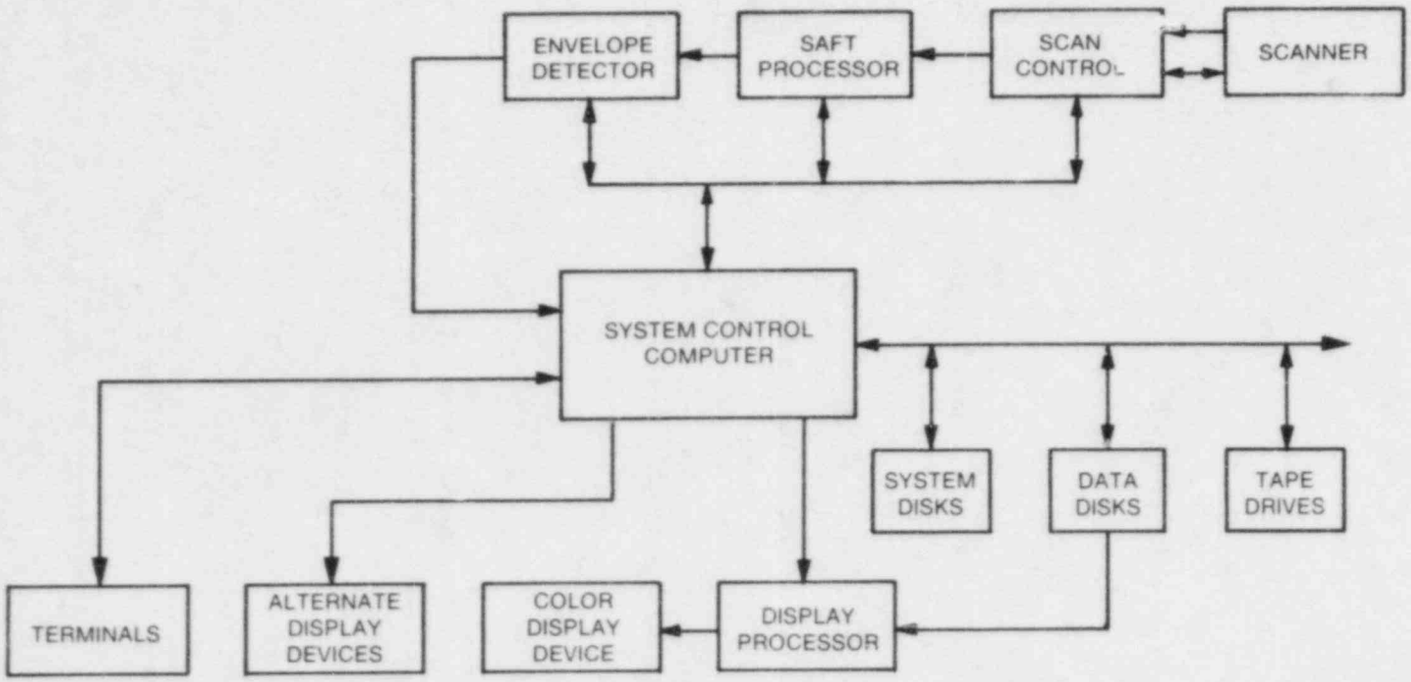


FIGURE 2-1 Block Diagram of SAFT-Based Inspection System

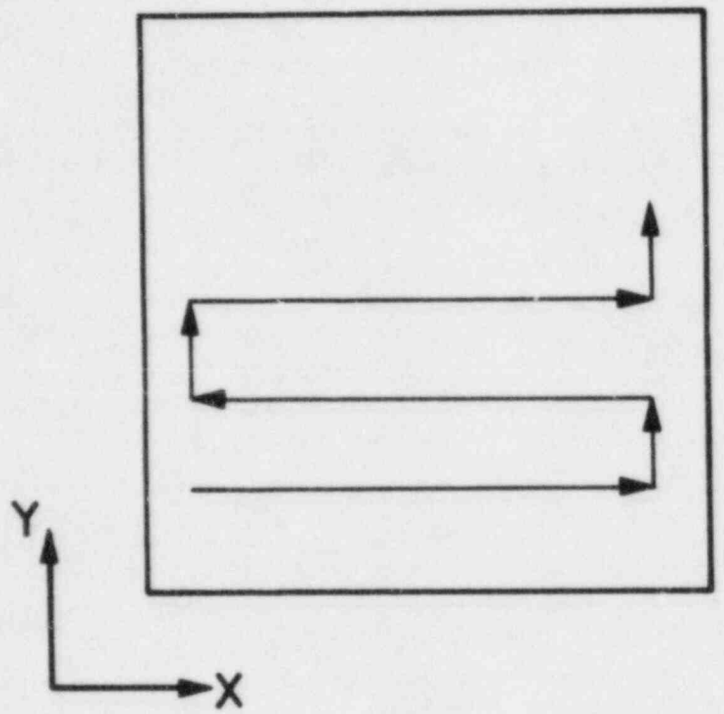


FIGURE 2-2 Scanning in a Raster Pattern

terminals connected to the SCC.

Scanning and processing functions are initiated and monitored by the SCC with software with all the processing done by the attached processors. Some display functions may be hardware assisted while others will be implemented completely with software in the SCC. New algorithms for processing, analysis and display can also be easily implemented, tested and used on the SCC.

2.2.2.2. Scanner

The scanner is divided into three parts: the mechanical scanning mechanism which contains the transducer, the electronics and digitizing equipment, and the scanner controller (SC). The mechanism itself takes the transducer along a raster scan path (shown in Figure 2.2) as instructed by the controller. It must keep the transducer at a constant height from the surface and at a constant angle with respect to the surface. The mechanism will require resolution down to approximately 0.001 inch, and will require four or five degrees of freedom.

The digitizing equipment, upon command from the SC, will pulse the transducer, process any required delays and then digitize the received data. This equipment also includes any required amplifiers, filters, etc. All parameters are set before starting by the SC.

The SC is a general purpose mini or micro computer. It is given necessary data about the region to scan and the scanning parameters (frequency, etc), from the SCC. It will then control all the scanning with no further intervention. The controller must keep track of all positional and angular parameters associated with the scanner, convert them to the appropriate form, and transfer them to the SAFT processor at the right time. It also controls the transfer of collected data and will suspend collection if the SAFT processor is getting behind.

2.2.2.3. SAFT Processor

This module accepts data and positions one A-scan at a time from the scanner. They are placed in internal mass memory and then processed. Processed data is output on an A-scan one at a time to the envelope detector.

The SAFT processor is controlled by the SCC. All processing parameters are loaded from the SCC before scanning starts. Once scanning starts the SAFT processor will run without intervention. Its only interaction with the scanner and SCC is to make sure data is available or that take can be accepted.

2.2.2.4. Envelope Detector

This is a simple piece of hardware that will envelope detect a single A-scan. It accepts an A-scan from the SAFT processor whenever one is available (and the detector is free) and transfers the detected A-scan to the SCC (if the SCC is ready). The envelope detector must be able to

operate at about the same rate as raw data is collected, which should be under 50 A-scans/second.

2.2.2.5. Disk Storage System

The scan data currently being accessed is stored on a disk system so as to be readily available. While scanning is in progress processed A-scans are transferred to the SCC and are written to disk. Display operations will operate on data stored on disk (although online display of data as it is collected may also be possible). The disk capacity will be sufficient to store the data for at least one circumferential vessel weld. In addition to devices used for data storage, one or more "system disks" will be required to store programs and other information for the SCC.

2.2.2.6. Tape Storage System

All data that must be retained but is not currently being accessed will be stored on high-density magnetic tape. Newly collected data will be copied to tape from the disk system and old data that is to be re-displayed will be copied from tape to the disk system. A catalog of such archived data will be maintained on the SCC. In addition the type system will be used for backing up the system disks and for exchange of software and data.

2.2.2.7. Color Display System

This system consists of a standard color display terminal plus possible hardware to speed up complex display processing operations. It is controlled by the SCC and used for walk-through inspection and for other pseudo-color displays.

2.2.2.8. Other Display Devices

These are any other devices deemed desirable, such ad digital plotters. They will probably be driven completely by SCC software and probably will not operate in real time.

2.2.3. Detailed Field System Specifications

The specification of any details in the design of a complete system is being delayed for two reasons. First, the results of a prototype SAFT processor are desirable as its actual performance may influence the capabilities of the rest of the system. Second, some aspects such as data storage and display and analysis techniques can be determined only after some more experimentation. When these issues have been resolved the full system can be specified along the lines described above.

3. Design Details of a Special Purpose SAFT Processor

3.1. Introduction

This section describes the design details for a special purpose SAFT processor suitable for use with the field inspection described above. While this section does not specify all the details (particularly fabrication methods, specific device types, etc.) it is intended to provide a good hardware fabrication facility with enough information to build a working device.

Important aspects of this design have been based on decisions made from results presented in earlier reports. The reader is referred to earlier reports for explanations and justifications of decisions and particularly for background information on the SAFT algorithm and on software machining.

3.2. Capability and Performance Requirements

3.2.1. Volume Types and Scanning Modes

The processor must be able to process flat and cylindrical surfaces with limited variations in surface height from the ideal surface position. Nozzles and other irregular surfaces will not be handled in this version. The surface variations can be compensated for with the software machining technique. Both normal incident and squint data must be processed. Squint data includes both longitudinal and shear modes.

3.2.2. Aperture Profile Restrictions

An expanding aperture will always be used. The aperture shape can be somewhat arbitrary but will usually be circular or elliptical. The maximum aperture size (diameter) in any direction will be limited to about 6 inches. At 2.25 MHz operation ($\lambda = 0.1$ inch) this comes to about 60 points. For typical profiles this means a maximum of about 3000 points for a single aperture (number of off-center points accessed to process a single center point).

3.2.3. Data Volume Requirements

The volumetric requirements for field inspection outlined in the previous section require storing raw data for a 24 inch wide region (for imaging an 18 inch wide area) at frequencies up to 2.25 MHz. Since the maximum aperture size is 6 inches the processor should be able to hold an array of A-scans of size 256×64. The requirement of a 12 inch deep A-scan means A-scans should be about 1024 points long. All data points will be 8 bits (one byte) so this requires 16 megabytes of physical memory for the full-blown field version.

3.2.4. Performance

The stated goal for performance in the previous section was a 200 cubic inch per minute rate at 2.25 MHz using a 60 degree circular aperture. This requires a basic cycle time of about 20 nS, meaning that a single serial processor would need to process the inner loop with that cycle time. This rate will allow the processor to keep up with the scanner.

3.2.5. General Interfacing

This processor will interface directly to the scanner and envelope detector hardware as explained in the previous section. Data is transferred to one complete A-scan at a time and each A-scan is transmitted byte-serially. Handshaking must be provided so that the SAFT processor knows when data is available and when data can be output. Because irregular surfaces might result in irregular data collection times the data transfer should be asynchronous. The SAFT processor waits for its input data if necessary and will also wait (and hold up the scanner) if the envelope detector gets behind. The SAFT processor is also interfaced to the System Control Computer (SCC) to facilitate loading of commands and tables.

3.3. Block Diagram

A block diagram of the SAFT processor is shown in Figure 3-1. All commands and parameters are sent to the controller by the SCC. Once all parameters are loaded the processor runs without intervention from the SCC, interacting only with its input and output streams to transfer data. The scanner interface simply communicates with the scanner and maintains flags that tell the controller when enough data has been collected. The software machining module adjusts A-scans to compensate for surface variation and also keeps memory pointers to indicate where the next A-scan goes. It also interacts with the memory to insure that the processor array always gets access when it requires it.

Data points are read into the processor array by the PE controller and broadcast to all the PEs. Processed A-scans are read out of the processor array and given to the output interface where they are sent to the envelope detector.

The lookup table holds shift and add information for each off-center A-scan in a center A-scan's aperture. Note that each center A-scan uses the same aperture profile. All the shifts are precomputed by the SCC, before scanning starts, for an ideal flat or cylindrical surface and theu loaded into the processor's lookup table memory. No other calculations are required for the rest of the processing run.

3.4. Processor Architecture Decisions

The two most important decisions affecting the design are on the processor architecture and the method of calculating time shifts. In an earlier report [NUREG CR/2703] several SAFT

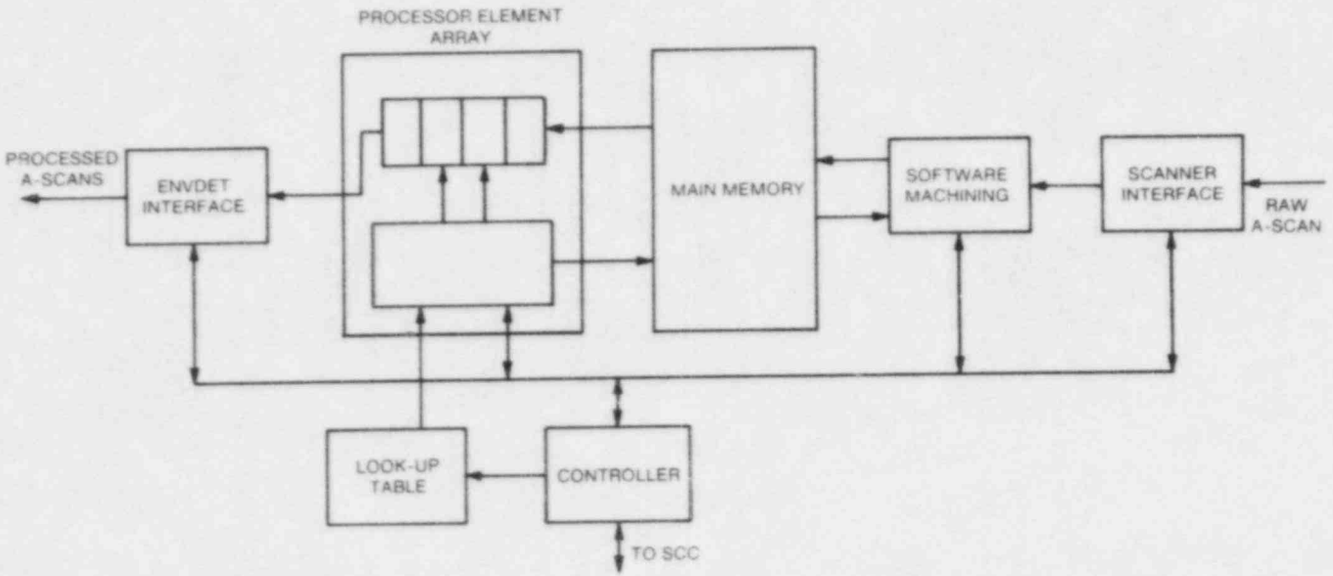


FIGURE 3-1 Block Diagram of a SAFT Processor

processor architectures were developed and one, known as the "data broadcast" or "MISD" architecture was chosen for implementation. This architecture serves as the framework for the core of the SAFT processor.

It was also pointed out in NUREG CR/2703 that there are several ways to calculate SAFT time shifts. Table lookup methods are simple and very fast but can be inaccurate on irregular surfaces. Exact calculations give perfect shifts for any surface geometry, but require time consuming arithmetic calculations and therefore complex hardware to achieve required speeds. Section 4 in NUREG CR/2703 describes a compromise called *software machining* which is almost identical to table lookup methods in speed and complexity but gives good accuracy for surfaces with limited variations. We have chosen to implement software machining in this first generation SAFT processor primarily because it will greatly simplify design, fabrication and maintenance, all of which are important considerations in an experimental system. We also feel that almost all surfaces that this system is intended to work with are within the bounds required for good accuracy.

3.5. Processor Coordinate System

The data stored in the processor is viewed as a plane of A-scans as shown in figure 3.2. The Y direction always runs parallel to the weld center. Primary scanner movement is always in the X direction. The memory holds exactly one aperture width in the Y direction and enough in the X direction to cover the weld plus additional used to get complete A-scans over the region being imaged. The X and Y sizes (NX and NY) are parameters read from the SCC.

Data collection in the Y direction is done from top to bottom (as it appears in the figure) or in increasing Y. The most recently collected row of A-scans is always written into the (logically) last row of memory in use (row NY). (Row numbers start from 0 and since there are a maximum of 64 rows NY can be up to 63.) As a new row is entered the center row of the existing data is being processed. When processing and collection of the next row is complete, the topmost (Y=0) row is discarded and the remaining rows are (logically) shifted and the process repeats.

No positional information is included in the A-scans. All A-scans are assumed to be collected on a constant spacing and this spacing is incorporated into the lookup table. A surface position parameter comes with each A-scan from the scanner. The software machining module uses this to adjust the A-scans so the height is not saved in main memory.

3.6. SAFT Processing Algorithm

The SAFT processor is basically a hardware implementation of a specific algorithm. This section presents the proposed algorithm in several levels of detail, along with descriptions of parameters and nomenclature. The basic SAFT algorithm is described in NUREG CR/2703.

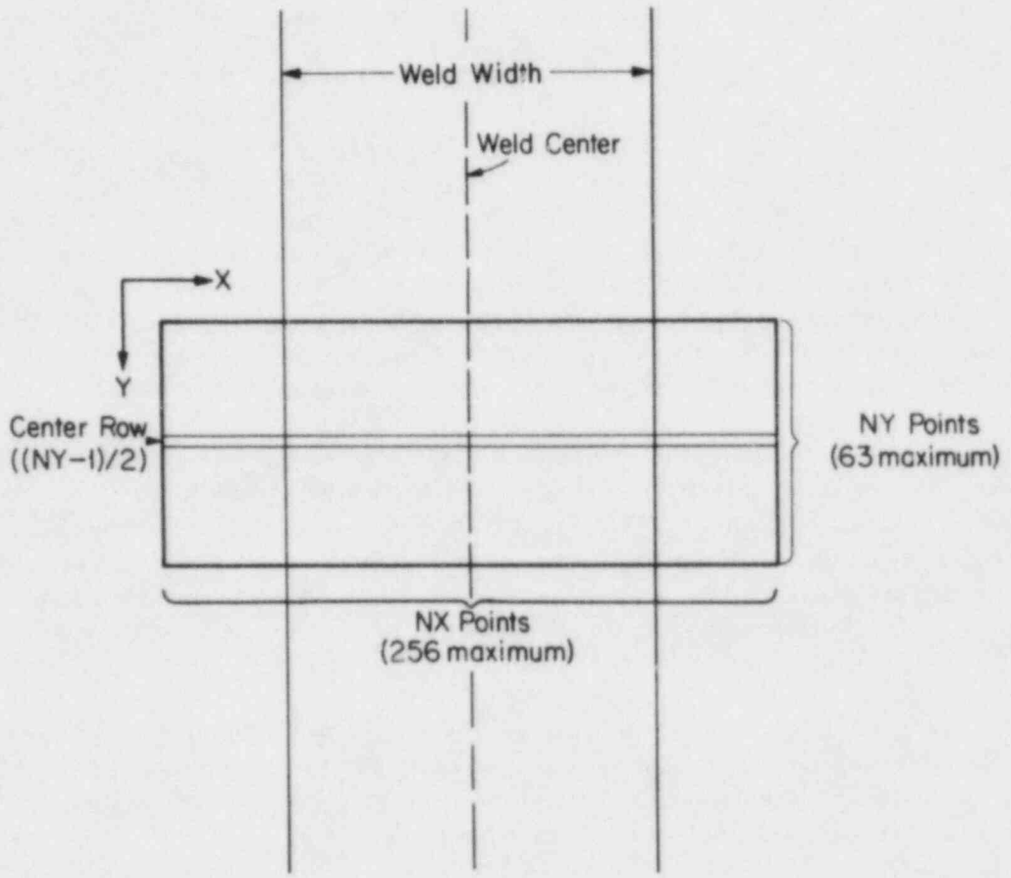


FIGURE 3-2 Processor Coordinate System

3.6.1. Application of Parallelism

The basic serial SAFT algorithm must be split up among several processing elements (PEs). Since the data broadcast algorithm will be used each PE will execute the same (complete) serial algorithm but will operate on a separate center A-scan. The A-scans contained in the union of the apertures of all center A-scans will be broadcast, one point at a time, to all PEs. This is described in detail in Section 3 of NUREG CR/2703.

PEs will always be assigned to a linear sequence of A-scans in the current center row of the stored data. This assignment is illustrated in Figure 3.3A. On each pass each A scan assigned to a PE is completely processed before the next pass is started.

Figure 3.3B shows the order in which off-center A-scans will be broadcast. The aperture shown is a square one but the same order must be used for any type of aperture.

3.6.2. Top-Level Algorithm

At the highest level of control is the management of raw data in main memory, the assignment of PEs to center A-scans and the broadcasting of data points and control tables.

The main memory is a dual port type that permits data to be read out and written simultaneously. It can be interleaved instead as long as reading out data points always gets priority over writing in new data. As points are read out for processing one center row of A-scans, the scanner interface is collecting new A-scans and writing them into the free row in memory. The interface will write one complete row, update its pointers, and wait to be restarted. It must also maintain a flag to tell the controller when it is done so that processing doesn't continue until all data is available.

The top level steps are as follows:

1. Entire SAFT processor is initialized by SCC.

2. Processing parameters (like A-scan size) and control tables are loaded from SCC.

3. Processor starts scanner interface running and continues until NY rows are in memory.

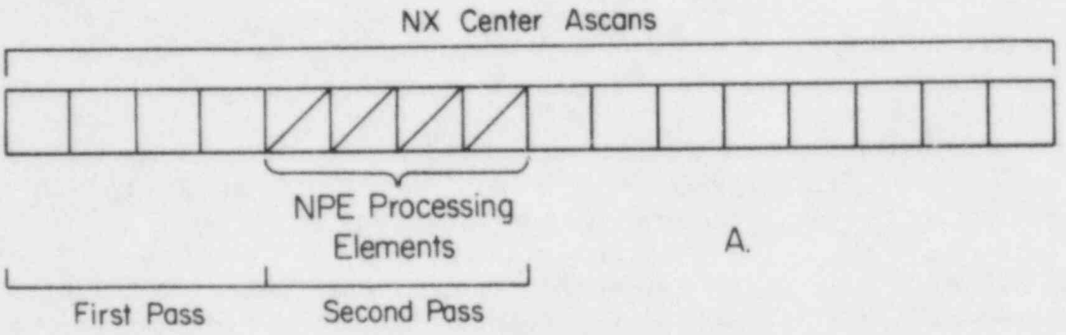
4. Execute loop until halted.

    5. Restart scanner interface to collect next row.

    6. Do gint(NX/NPE) times.

        7. Assign PEs to next set of A-scans in center row.

        8. Broadcast all necessary off-center A-scans and control tables.



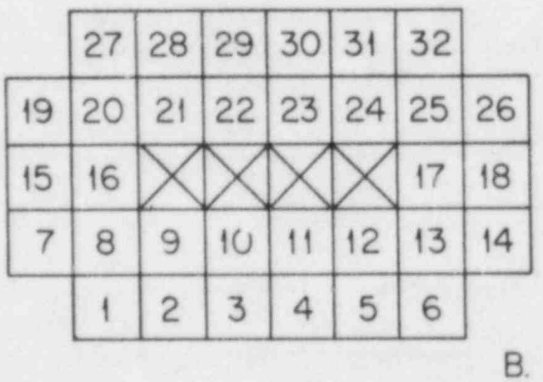


FIGURE 3-3 Assignment and Ordering of Processors

9. Read and output processed center A-scans from PEs.

10. END.

11. Wait for interface if necessary.

12. Adjust memory pointers.

13. END

Note that some steps, such as outputting processed A-scans, might be done in parallel with the rest of the processing.

3.6.3. A-scan Broadcasting

To enable control tables to be passed between PEs, the off-center A-scans must be broadcast in a line parallel to the PE lineup as shown in Figure 3-3A. The control memory can have the tables organized in this same order, so table selection is just a serial read of the control memory on each pass.

Note that there are only tables for one complete aperture but the number of A-scans broadcast for any pass is greater due to the presence of several apertures. During these holes, however, processors without tables must actually be idle since the data being broadcast is outside their apertures. This problem can be handled easily. During system initialization all the delay buffers between PEs are cleared to contain only SHIFT commands. Thus all PEs but the first will ignore the data that is outside their apertures when a row is started. When all tables for a row have been sent, the next A-scan broadcast is outside the aperture of the first PE, so the controller starts transmitting tables of all SHIFT commands. These will fill up all the buffers so when the next row is begun, the PE array is in the same state it was in at the start of the last row.

A few additional pieces of data are required to define which A-scans are to be broadcast. The first are the start and end rows which define one extent of the aperture. Then there is a vector of data, one slot for each row. This slot contains the first A-scan to send relative to the position of the first PE position and the number of A-scans that must be broadcast in that row. Like the control tables, these numbers are determined by the aperture profile and are precalculated and loaded into the SAFT processor before processing begins.

3.6.4. Processor Element Algorithm

Each PE only has to maintain a center A-scan vector and an input data queue in addition to performing simple register to register arithmetic operations. On each cycle one data point and one control command are received by each PE, with housekeeping done each time a new center A-scan is started. The algorithm is as follows:

1. After initialization, read A-scan size off the data bus.

2. Do for each center A-scan.

    3. Clear center A-scan vector and count vector.

    4. Do for each off-center A-scan broadcast.

        5. Pointer = 0.

        6. Do for each datapoint in A-scan.

            7. Read datapoint into input data queue.

            8. Execute command:
                If ADD: Add first point in queue to center [POINTER] and
                increment count [POINTER].
                If SHIFT: Remove first point from queue.
                Increment POINTER.

        9. Next datapoint

    10. Next off-center A-scan.

    11. Average each center point and output it.

12. Next center A-scan.

Note that if two sets of internal vectors are implemented, the processing and divide/output/clear operations can be done simultaneously. Since processing is by table lookup it is also possible to remove the counters and division hardware from the PEs and put a single divider in the SAFT processor. Since the aperture size is known for each depth in this case they can be stored in the SAFT processor and the divisions done serially as the data points are read out.

3.6.5. Software Machining

Software machining is done on raw data before it is written into main memory. This is simply a process of shifting the A-scan up or down so that its first point is at the position of the front surface where it was collected. Excess data on one end is then truncated so that the first points of all A-scans are at the same level of some ideal surface. The position of this ideal surface must be predetermined, possible with a surface mapping technique. The coordinate of this ideal surface is then stored in the software machining module.

When an A-scan comes in from the scanner its surface position is subtracted from the ideal surface position. If, for example, the surface is three increments higher than the ideal position, the A-scan is shifted up 3 points. The first three points are lost and the A-scan has effectively been

"machined". The three holes created at the bottom must be filled with the average value of the A-scan or some other appropriate value (the center value of the transient recorder's dynamic range is another possibility).

3.7. Guidelines for Construction of a Prototype

The prototype is intended to be a scaled down version of the field processor. It also is attached only to a laboratory computer. It is useful, however, if the complete field processor is built except for size. This would make it easy to expand the prototype to full capabilities.

The memory of a prototype can be restricted significantly, probably to half a megabyte. The A-scan array can be 64 wide by 32 long. This will permit an aperture to be up to 31 points wide in the Y direction. If the aperture size in the X direction is held to 25 points then a 4 inch wide weld can be processed (12 points required on each side for overlap). The maximum A-scan size can also be limited to 256 points. This permits A-scans to be a little over 3 inches long. This memory size should be adequate for scanning most test samples and calibration blocks.

The processor should be designed with a cycle time as close to that to be used in the field as possible to prevent the need for a redesign. It is felt that this time can be around 100 to 160 nS while keeping the number of processing elements reasonable. The 160 nS time will allow a single processor to give up to 20 cubic inches per minute at 2.25 MHz and 45 degrees, if the A-scans start at the sample's surface. If two or three processing elements are built a significant processing rate (for laboratory conditions) can be achieved.

If possible, the interfacing should be identical to that to be used in the field. The control interface can be connected to the laboratory system just as if it was the SCC. Two additional parallel interfaces can then be added to the computer. Under software control these can be made to look like the scanner and envelope detector devices. The lab system will perform all the initialization functions of the SCC. It then feeds prescanned data (stored on disk) to the processor and reads and stores the processed A-scans. These can then be envelope detected by software and displayed with current display programs.

4. Display Techniques for Inspection and Analysis.

4.1. Introduction

A useful feature of any display system for nondestructive evaluation is the ability to animate a group of related images to make volumetric analysis of flaw size and location. Two examples are:

(1) The viewing of pseudo-color images of sequential planes, a process visually similar to destructively sectioning a sample image plane by image plane (WALKTHRU).

(2) The real-time rotation of a perspective view of a single image plane.

In the animation process we assume that all images are raster, and that they have been created prior to presentation here. These assumptions simplify the task of the host computer to one of selecting the next image to be displayed and transmitting it to the display

4.2. Hardware Requirements

Some general requirements for the display processor are presented here. First, an integrated frame buffer is needed; that is, a display which stores a single color number for each pixel. These numbers are used as addresses into a color lookup table (colormap) which maps input data into display colors. This is contrast to a segregated, or true-color, display which has an independent memory for each of the red, blue, and green components of an image.

In addition, the display controller should have the following capabilities:

(1) PROGRAMMABILITY - The user should be able to download firmware to customize the controller's handling of input data.

(2) CONCURRENT PROCESSES - The controller should be composed of synchronous independent processes communicating with each other (and the host) only as necessary. An example of this would be a separate module to monitor peripherals. This module would communicate with the display controller rather than being polled by the host every time a keyboard or joystick is needed.

More specific hardware requirements will be given in the next section.

4.3. Implementation

This section describes two display algorithms which could be used for either the WALKTHRU or the real-time rotation. The first of the algorithms encodes two images concurrently into a single block of memory. The second uses a double buffer memory. The interactive control of the image sequence will also be discussed.

In each of the algorithms we assume that the images have a maximum resolution of 512*512 pixels and n bits per pixel. The color map must contain less than 2^n values, however. We will also assume that standard firmware operating in the display controller will handle primary scan

and line-to-line update of image memory, so pictures can be transferred in a continuous stream. Each algorithm requires storage of two images at all times.

4.3.1. Single Memory

The implementation of a single memory animation system is based on the concept of m independent channels within each memory word. Any single channel or combination of channels can be enabled/disabled by means of a SUBCHANNELS MASK, an m bit word which is AND-ed to data as it is written to, or read from, memory. In addition to the mask, logic should be available to perform other logical functions between image memory contents and incoming data.

If $m > 2n$, the SUBCHANNELS MASKS could be used to partition the memory planes to simultaneously hold two n bit images. A possible configuration of such a scheme would require:

(1) 512x512x2n Image Memory; Colormap 2^{2n} entries long; An *n* bit shifter for incoming data which is selectively enabled/disabled; Two independent subchannel masks, one for the incoming data (the WSM), and one for the display (the RSM).

Image 0 will reside in the lower *n* bits of memory. Thus, to access Image 0 for display, the RSM is set to $2^n - 1$. The WSM is set to the same to write a new Image 0 into memory.

Image 1 resides in the upper *n* bits. It is accessed as above by setting the appropriate mask to the value ($2^{2n}-(2^n)$). When a new image 1 is to be written into memory, the *n* bit shifter should be enabled so that image will be placed in the correct planes.

The color map for Image 0 is placed in locations [1: (2^n-1)] of the lookup table, inclusive. The corresponding map for Image 1 begins at location 2^n , with a new entry every 2^n places. For a five-bit image, the map would be placed consecutively in locations 32,64,96,...($2^{2n}-2^n$).

Once an initial Image 0 is in memory, the display controller operates in two cycles as follows:

Cycle 0
Display Image 0
Update Image 0

1. Send Joystick status to Host
2. Set RSM to (2^n-1)
3. Set WSM to ($2^{2n}-2^n$)
4. Enable *n bit shifter*
5. Read Image 1 from Host

Cycle 1
Display Image 1
Update Image 0

6. Send Joystick status to Host

7. Set RSM to ($2^{2n}-2^n$)
8. Set WSM to (2^n-1)
9. Disable shifter
10. Read Image 0 from host
11. Go to 1

Image 1 is always updated while Image 0 is being viewed, and *vice versa*. The speed at which this algorithm can proceed is almost entirely determined by the rate at which images can be transferred from the host. Experimentally, four images/sec has been found to be the most pleasing refresh rate.

4.3.2. Double Memory

The above algorithm can be modified to use a separate memory block for each image. The benefits of added memory are considerably simpler firmware and the need to keep only one color map present. Specifically, we would need the following configuration:

(1) 1024 x 512 x n Image Memory;

(2) Lookup table 2^n word long;

(3) Four registers controlling memory access; the READ ORIGIN (RO) containing the line and column of the first pixel of the display, and the WRITE ORIGIN (WO) containing the line and column of the first pixel of the incoming image. No subchannels mask or shifter is needed.

Part of the smooth animation effect is achieved by positioning the images in memory at the same location relative to some origin. Image 0 is placed relative to (0,0); Image 1 is placed relative to (512,0). As a result, the WO and RO are identical except for the value of the uppermost bit of the line counter in each. Furthermore, if that bit is set to 1 in the RO, it must be set to 0 in the WO. A single flip-flop can be used to control the RO and WO; simply toggle the flip-flop to alternate between the two images.

The firmware operates in a single cycle:

1. Send Joystick status to Host;
2. Toggle flip-flop;
3. Read next Image from Host;
4. Repeat.

Clearly, the host does not have to know whether it is transmitting Image 0 or Image 1. Its function is to decide on the next image to transfer based on the status of the joystick. Details on this process are given in the following section.

4.3.3. Selection of Next Image

The only difference in the operation of the two algorithms above in their application to WALKTHRU as opposed to real-time rotation is in the selection of the next image to be displayed. (There is no difference from the viewpoint of the display controller.) Interactive control over this selection can be maintained using the joystick. During WALKTHRU the host must maintain links to (at least) three files: 1) the image currently displayed; 2) the previous images in the sequence; 3) the next image in the sequence if forward motion is continued. At the start of over 1/4 second refresh cycle, the status of the joystick is sent to the host. If the stick is displaced in a certain direction (UP or RIGHT), the next image in the sequence is transmitted. If it is moved in the opposite direction (DOWN or LEFT) the previous image is transmitted. If the stick is in its home position the currently displayed image is re-transmitted. It is important that a file be transmitted during each cycle. The file selection process for real-time rotation is more involved than for WALKTHRU, as (at least) nine file links must be maintained. In addition to the current image, links must exist to the images which would result if we were to rotate:

1. Clockwise in the plane;
2. Anti-clockwise in the plane;
3. Up
4. Down
5. Clockwise and up
6. Clockwise and Down
7. Anti-clockwise and up
8. Anti-clockwise and down

Joystick motion to the RIGHT and LEFT would control azimuth rotation, while movements UP and DOWN would control rotation in those directions.

4.4. Summary

We have presented two algorithms for interactive real-time display of raster image sequences. The algorithms rely on placing most of the intelligence of the system inside the display terminal, making it a stand-alone image processor, communicating with the host only when input data is needed. The hardware requirements given in this description can be obtained from a number of vendors and are features we believe will be offered as standard or optional on many more commercial systems in the future.

5. A Comparative Study of Different Edge Operators Applied to Ultrasonic Images

5.1. Introduction

Several edge operators have been developed during the past few years. Some of them have been invented because of the need to process a certain class of images. No single operator is perfect for all different areas of application. Hence, there may be new operators to come in the next few years. A number of investigators have compared these operators both qualitatively and quantitatively [1,2,3,4]. It is not feasible to consider all possible operators for a comparative study. The major constraint is the diversity of these operators which prevents one from devising a common set of tests to compare them. However, many of these operators are similar and exhibit similar behaviors. Four operators are studied in this report. At one end, Robert's cross gradient operator has been included, and at the other end, the operators based on Hummel's work [5]. While the former is very trivial and simple, the latter is computationally complex. Moreover, because of the nature of the research, attention has been focused primarily on ultrasonic images. These images are obtained by ultrasonic scanning of steel specimens, which are monochromatic and generally low-contrast images with few detail. It is essential that operators do not fail in detection of flaws in these metal samples. A flaw could be due to a hole, a crack, a slack inclusion, etc. In what follows, a concise description of each operator is given. Different tests and experiments applied are explained. Finally, the conclusion of this study has been given along with some remarks about CPU time for each operator.

5.2. Edge Operators

Digital images to be processed are represented by a matrix F whose elements will be referenced by $f(i,j)$, where $i=1,2,\ldots,$ number of rows and $j=1,2,\ldots,$ number of columns.

5.2.1. Robert's Gradient

This operator is supposedly the simplest operator that is capable of detecting edges in any direction. It is defined in the following two forms:

$$g(i,j) = \text{value of gradient at point } (i,j) =$$

$$\equiv \sqrt{([f(i,j)-f(i+1,j+1)]^2+[f(i,j+1)-f(i+1,j)]^2}$$

or:

$$g(i,j) \equiv |f(i,j)-f(i+1,j+1)| + |f(i,j+1)-f(i+1,j)|$$

The second form is computationally simpler but is slightly worse than the first one. The first form was chosen for computer implementation and the gradient value calculated is actually $1/\sqrt{2}$ times the value given by the above formula. (This weighting factor guarantees that there would be no overlow in the value of the gradient. That kind of restriction should be imposed because of display purposes. In an 8-bit display system, the minimum and maximum values that each pixel can take are 0 and 255, respectively. It certainly is desirable to display the gradient image without any overflows, because that may result in a wrong shade.) The final step is thresholding of the gradient image according to the following scheme:

$$b(i,j) = \text{binary image (edge map)} \equiv \begin{cases} 255 & \text{if} \quad g(i,j) \geq a \\ 0 & \text{else} \end{cases}$$

where a is the threshold value. As it will turn out, the most difficult problem with these edge detectors is to find the optimum threshold value so that neither any of the edges are missed nor

any artifacts are introduced by detecting spurious edges.

5.2.2. Sobel Operator

Suppose that $(2n+1)$-neighborhood of the point $f(i,j)$ is shown by $B_n(i,j)$. Then Sobel gradient is defined as follows:

$$g(i,j) \equiv [\frac{1}{20}[(B_1(i,j),W_1)^2+(B_1(i,j),W_2)^2]]^{\frac{1}{2}}$$

where $B_1(i,j)$ represents a 3×3 square neighborhood of point (i,j), and the dot product of two matrices is defined to be:

$$(B_n(i,j),W) = \sum_{k=-n}^{n}\sum_{l=-n}^{n} f(i+k,j+l)w(k+n,l+n)$$

where W is a $(2n+1)\times(2n+1)$ matrix. W_1 and W_2 are the following matrices:

$$W_1 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad W_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Remark: The 1/20 factor included in the above formula is to avoid from overflows in the gradient image.

The final step, as usual, is to threshold the gradient image G according to the scheme described before. The threshold selection problem remains as difficult as before and needs careful treatment.

5.2.3. Frei & Chen Edge Detector

This operator is capable of looking for edges, lines, or both [6]. Given the fact that each 1-neighborhood of an image pixel $f(i,j)$ can be regarded as a vector in the 9-dimensional space, a suitable orthonormal basis has been found to span this space. Four of these basis vectors span an edge subspace, the other four span a line subspace, and the last vector has been

added to complete the basis. Then it is natural to expect that each neighborhood that covers a segment of an edge to have a significant projection on this subspace. In other words, the angle between the neighborhood and the subspace would be small. The normalized basis vectors are given below:

edge subspace:

$$W_1 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad W_2 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_3 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \quad W_4 = \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

line supspace:

$$W_5 = \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad W_6 = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_7 = \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad W_8 = \frac{1}{6} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

average:

$$W_9 = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Then the gradient image is defined as follows:

$$g(i,j) \equiv [\sum_{k=l}^{m} (B_1(i,j), W_k)^2]^{\frac{1}{2}}$$

where the following values for l and m should be used:

[a] $l=1$, $m=4$ edge detection

[b] $l=5$, $m=8$ line detection

[c] $l=1$, $m=8$ edge and line detection.

In this study, only the first case (i.e., edge detection) has been considered. The main distinction between this edge detector and the others is the thresholding scheme used. Here, instead of applying the threshold to the magnitude of the gradient vector, the angle between the image vector (neighborhood) and the feature (edge, line, or both) subspace is thresholded. Hence, the binary image is formed as in the following:

$$b(i,j)=\begin{cases} 255 & \text{if } \cos\theta \equiv \dfrac{g(i,j)}{\sqrt{B_1(i,j),B_1(i,j)}} \geq \cos\theta_0 \\ 0 & \text{else} \end{cases}$$

It has been claimed that the above thresholding scheme is superior to the standard way of thresholding the magnitude of the gradient vector. On some occasions, the results were not satisfactory, which made it reasonable to include the standard thresholding scheme in addition to the new scheme in this study. There is one minor modification to the thresholding scheme that would enhance the performance of this operator. The following measure for thresholding would make the detector insensitive to dc level of a step-like subimage. Then only the step height would be important. The new measure would be:

$$\cos\theta' \equiv \frac{g(i,j)}{\sqrt{(B_1(i,j)-\mu,B_1(i,j)-\mu)}}$$

where μ is the mean value of the nine pixels involved. Let us consider the neighborhood $B_1(i,j)$ and add a constant c ($c>0$) to all the pixels forming the neighborhood. Then letting this new neighborhood be represented by $A_1(i,j)$, values of $g(i,j)$ and $\cos\theta'$ for both neighborhoods will be the same. On the other hand $\cos\theta$ for $B_1(i,j)$ is larger than $\cos\theta$ for $A_1(i,j)$ because the denominator is smaller for the first one. The net effect of using $\cos\theta$ for thresholding is that, for the same step height, the detector is biased toward detecting dark edges (assuming 0 and 255 representing black and white, respectively) rather than bright ones. Therefore,

cos θ' was used instead of cos θ. It is apparent that cos θ' does not exist if all the nine pixels have the same value. In such an occasion, there is no edge or line present at all.

5.2.4. Hummel Edge Detector

This method [5] is an extension to the one originally suggested by Hueckel [7,8,9] and subsequently simplified by Ma' ro' and Va' ssy [10]. This is a general feature detector which can look for any class of patterns by finding the best fit between the local image data and a set of target patterns. These patterns are specified by a number of parameters. A common way of finding the best matching pattern is to minimize the mean squared error between the image data and a target pattern. This minimization problem could be eased by projecting both the image data and the target patterns on a finite dimensional orthonormal basis. Assuming that the subspace spanned by this basis would preserve most of the feature information, then it would be reasonable to approximate the problem by that of finding the set of target pattern parameters which would minimize the mean squared error between the projections. One useful observation is that the value of this second mean squared error is always less than that of the first one because it is a projection of the first one. It is desirable to specify how good the extent of the match between the optimum target pattern and the image data should be. This can be achieved by setting an upper bound (threshold) on the second mean squared error. Sometimes the match is such that this condition is satisfied although the actual mean squared error (the first one) has a high value. This situation arises because a finite-dimensional subspace is used instead of a complete infinite-dimensional one. The avoid such false detections, another upper bound will be put on the actual mean squared error. Hence, a neighborhood of image data will be accepted as a legitimate edge region only when both mean squared errors are less than corresponding thresholds. One may argue that only one threshold (the one on actual mean squared error) would be enough. The advantage of including the other threshold is that it

terminates the search for the optimum edge at an earlier stage. In practice, the inclusion of this threshold will cause some problems, although in continuous domain, the value of the first one should always be smaller than that of the second one. One major question that should be answered at this point is the matter of choosing an appropriate set of basis function, $\phi_i(x,y), i=1,2,\cdots$. It is desirable to make a cho ;e which would minimize

$$E \;\| S(x,y) - \sum_{i=1}^{n} a_i \phi_i(x,y) \|^2$$

where

$$a_i = \int\int_D S(x,y)\phi_i(x,y)\,dx\,dy$$

and D is the unit disk. The solution to this problem is called the Karhunen-Loeve basis functions and satisfy the following integral equation:

$$\int\int R(x,y,x',y')\phi_i(x',y')\,dx'\,dy' = \lambda_i \phi_i(x,y) \qquad \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots > 0$$

where

$$R(x,y,x',y') = E[S(x,y)S(x',y')]$$
$$= \int_{p_1}\int_{p_2}\cdots\int_{p_k} S_{p_1 p_2 \cdots p_k}(x,y) S_{p_1 p_2 \cdots p_k}(x',y') f(p_1,p_2,\ldots,p_k)\,dp_1\,dp_2\cdots dp_k$$

$S_{p_1 p_2 \cdots p_k}(x,y)$ represent the k-parameter family of target patterns defined over D. These parameters evidently are random variables and f is their joint probability density function, (pdf). The author has then derived the following basis functions for the case where edge orientation is a uniform random variable:

$$\phi_{2n-1}(\theta) = \cos(2n-1)\theta$$

$$\phi_{2n}(\theta) = \sin(2n)\theta \qquad n = 1,2,3\cdots$$

ϕ's are functions of only angular displacement defined over D. D is approximated by a 9×9 window (as in [8]) where three pixels are dropped from each corner. Hence, there are 69 pixels forming a neighborhood. Only the first four basis functions are retained, and the corresponding coefficients of local expansions are found by summations over 69 pixels instead of integration

over D. Then the equation

$$\sum_{i=1}^{4} c_i \phi_i(\theta_0)=0$$

is solved for θ_0 which is the optimum edge angle. This equation can have a maximum of three solutions in $[0,\pi)$. For each solution, there is one more associated solution which is π radians larger (the difference between two such solutions is the positions of +1s and -1s in the ideal step model. The edge orientation is the same for both). The above equation is obtained by setting the derivative of the mean squared error in the four-dimensional space equal to zero. To make sure that we are dealing with a real minimum as opposed to a maximum or a saddlepoint, the second derivative is calculated for each root. Only those roots are retained which make this second derivative positive. Then the mean squared error is calculated and if its minimum value over the remaining roots is larger than a threshold, the program terminates the search and moves to the next pixel. Otherwise, it calculates the actual mean squared error between the optimal step-edge model and the normalized input data (with zero mean and unit mean squared). If the minimum value of such errors over the remaining roots is larger than a second threshold, again, the search is aborted. Otherwise, the root which results in the least mean squared error is taken to be the optimum edge angle, and the corresponding pixels in the 9×9 neighborhood are turned on (=255). The program performs an exhaustive search for all the pixels of the image and is rather time-consuming.

5.2.5. The Experiments

There are a few parameters which are to be considered when we study the performance of an edge detector. The ones that are commonly used are: *i*) sensitivity of the detector to the noise present in image, *ii*) its sensitivity to threshold value (most of them have some kind of thresholding operation), *iii*) directional preference (that is, some of these detectors detect edges

along certain orientations better than other orientations), and *iv*) edge thickness. Not all of these parameters are easy to measure. It is easier to compare different edge detectors qualitatively rather than quantitatively, as far as possible. We have tried to devise reasonable experiments that would work uniformly for different operators and measure the above characteristics. It was not possible to include Hummel's detector in some of these experiments because of the following reasons:

[a] It needs two threshold values (in its current form) as opposed to the others which need just one;

[b] It is not possible to make a threshold sensitivity analysis on this operator using the same experiment that is used for the others. The reason is that it considers overlapping 9×9 regions, and when it detects an edge, it turns on a number of pixels and not just one. This will become more clear after the experiment is described;

[c] It is a very slow operator compared to the others, and it is hard to work with because it has two thresholds.

Different experiments are described below:

5.2.5.1. Noise Sensitivity

The noise-free image is processed with a selected edge operator, and a prespecified percentage of pixels are turned on in the binary image. If *a priori* knowledge is available of class of images that are to be processed, then it is not too difficult to make such an assumption about the amount of edge information in an image. For the case of ultrasonic images considered in this study, a reasonable figure is 3-10 percent. In order to achieve this goal of having an edge map with this prespecified percentage of edge pixels, the gradient (or whatever function that is to be used for thresholding) is measured for all pixels of the image. Then, several passes

through these values are made so that only those pixels with top gradient values, say the top 5 percent, are selected as edge points. The threshold value found in this way is recorded. Evidently, it is not possible to achieve such a goal with the Hummel edge detector. The edge map resulting from the noise free image in the above fashion is stored somewhere. Then, different amounts of zero-mean Gaussian white noise is added to the image, and the resulting image is linearly stretched to cover the whole range of (0,255). Such a linear transformation is done all the time even if there is no additive noise because there is a chance of having negative values which will cause display problems. In this group of experiments, noise with five different variances was added to each image. The signal-to-noise ration(SNR) was varied from 20dB to 0dB. Edge maps were obtained keeping the percentage of edge points constant. Again, threshold values were recorded and have been plotted for three different samples; as Test A (see Figures 5-1, 5-5, and 5-9). The distance between two edge maps was defined to be the number of positions where the two images have different values divided by total number of pixels in the image. We note that edge maps are binary images which take only the values of 0 and 255. This way of defining the distance is equivalent to the way that Hamming distance is defined for block codes. The edge maps were compared, and the distance between each one and the noise-free edge map was plotted versus SNR. This is Test B (see Figures 5-2, 5-6, and 5-10). In both these tests the percent of edge pixels was set at 3 percent. Moreover, the plot of threshold value versus SNR was drawn. The thresholds are all normalized to have a maximum value of one. This can be done by dividing the actual threshold value by 255 because the maximum value of gradient functions is 255 in all different detectors. The next part of the experiment was to process noisy images with the threshold set at the value, which was used for the noise-free image (obviously, the percentage of edge pixels is not kept constant here). Finally, another plot of distance versus SNR was drawn, but this time threshold value rather than edge amount is kept constant. This is Test C (see Figures 5-3, 5-7, and 5-11).

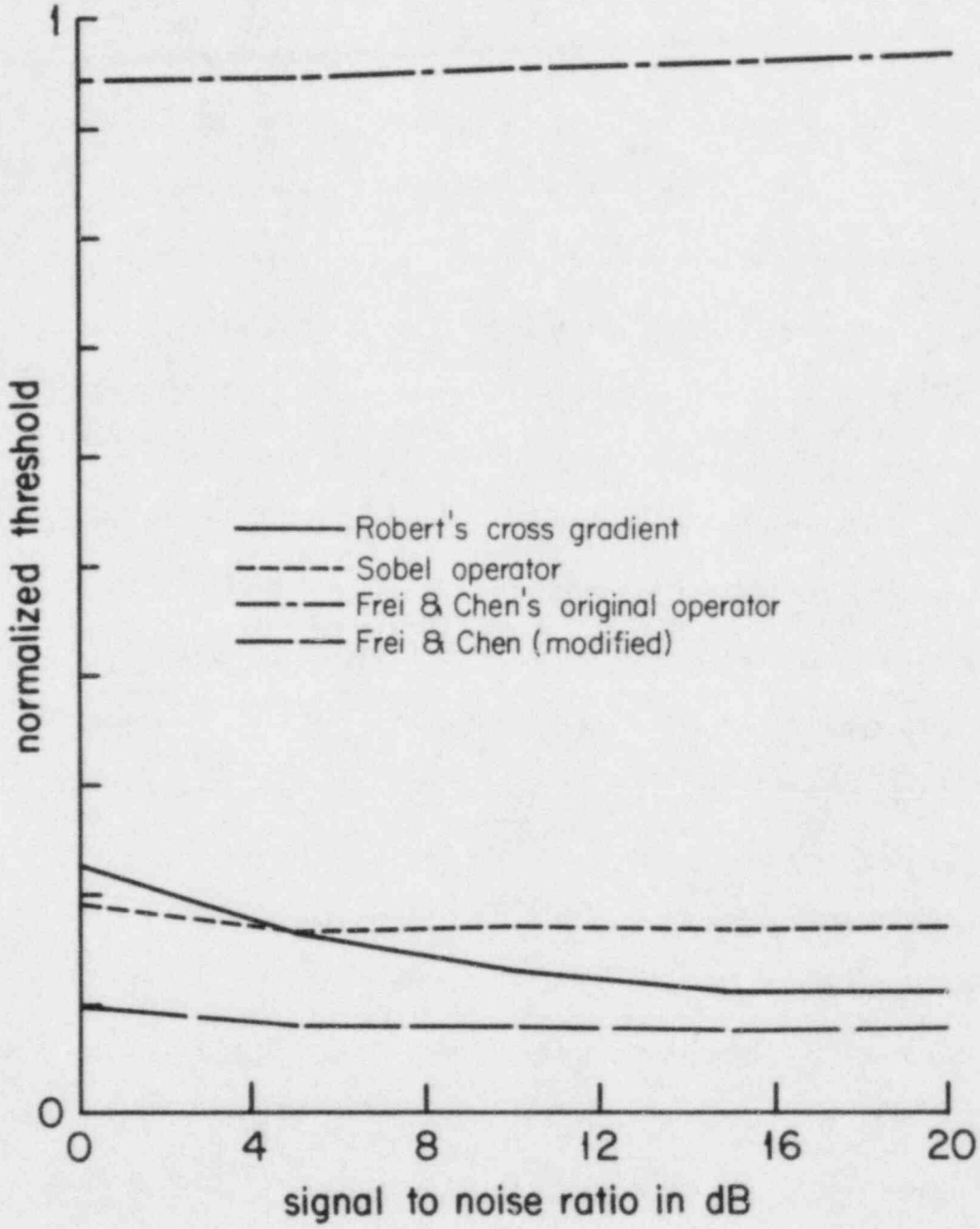


FIGURE 5-1 Sample 1, Test A

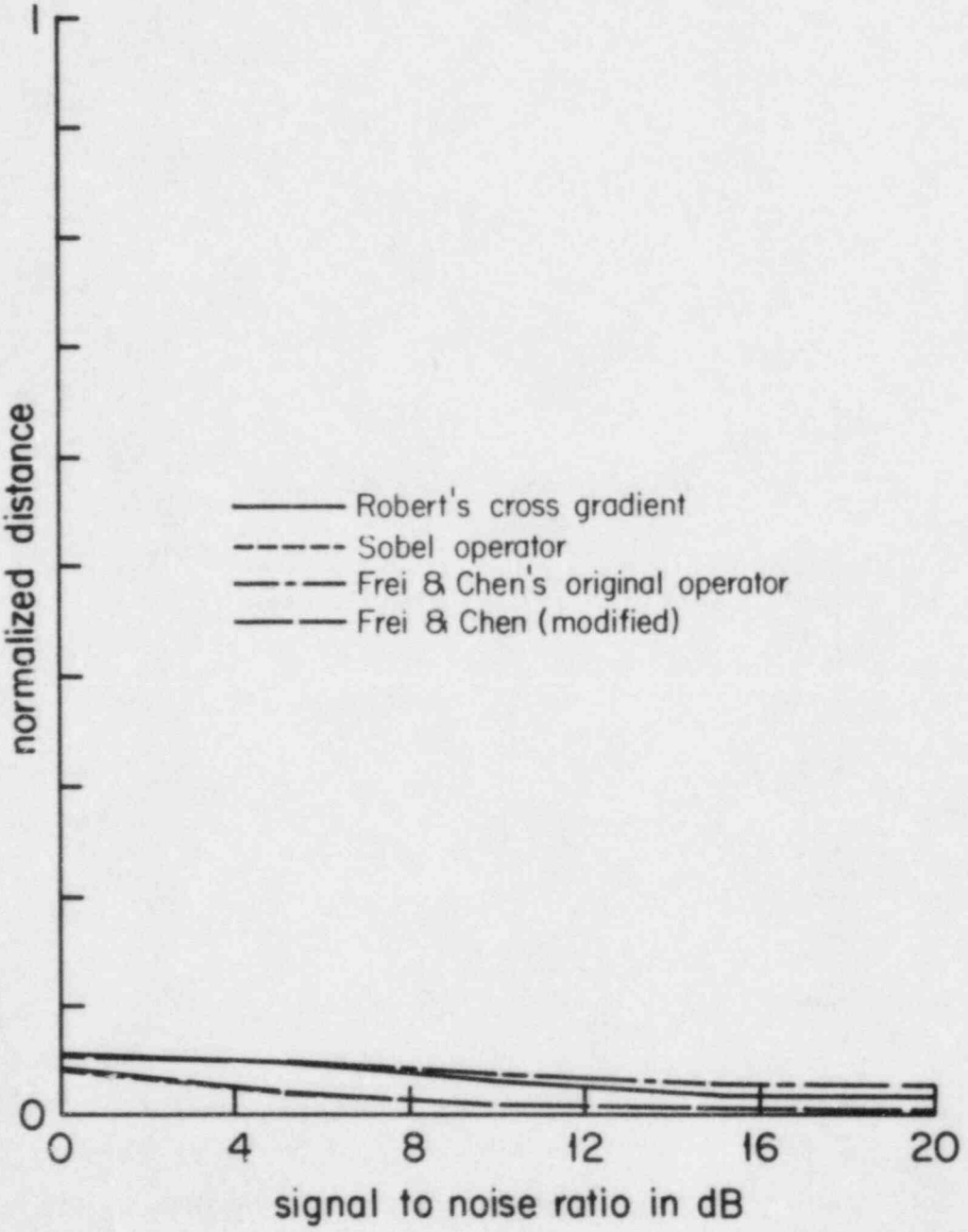


FIGURE 5-2 Sample 1, Test B

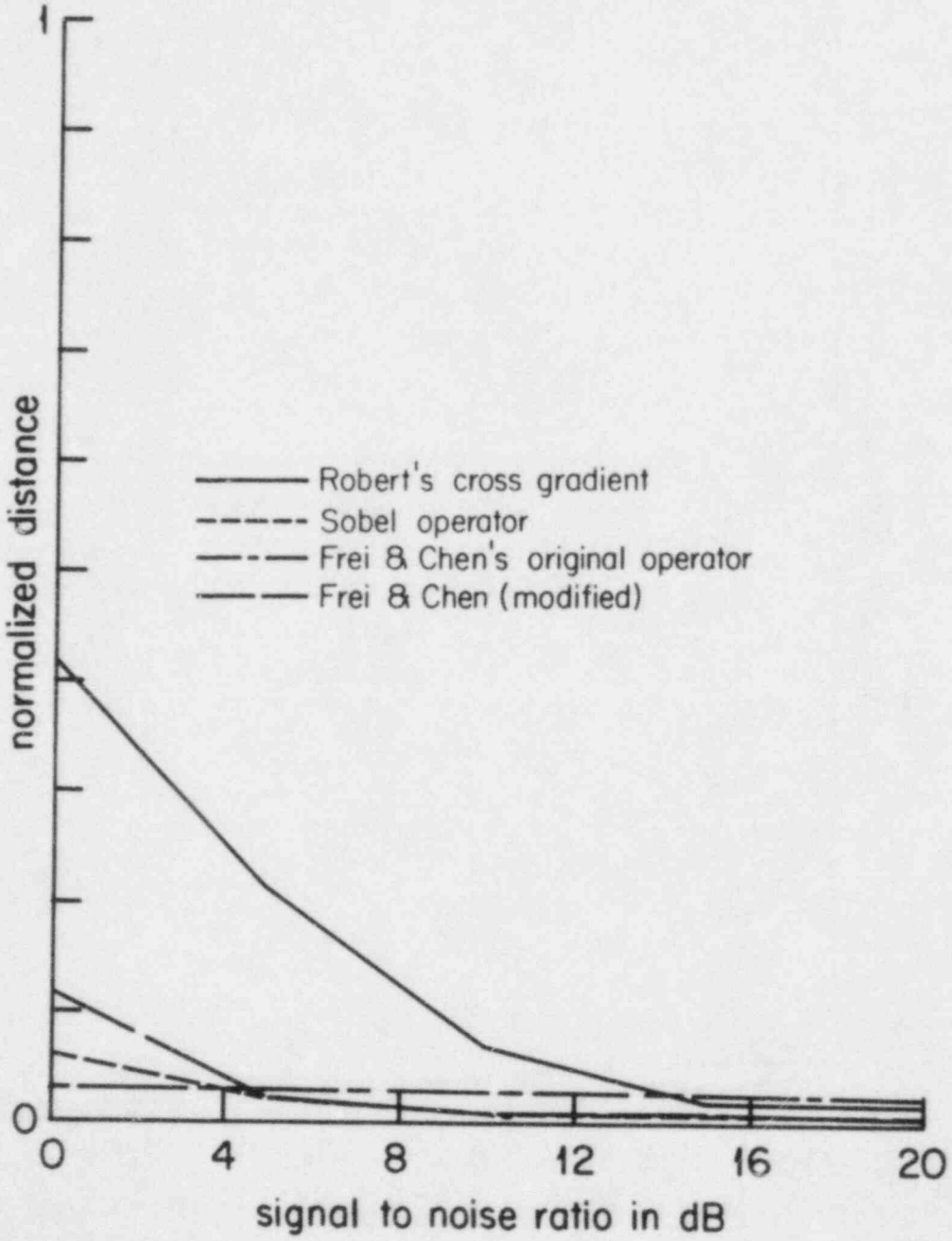


FIGURE 5-3 Sample 1, Test C

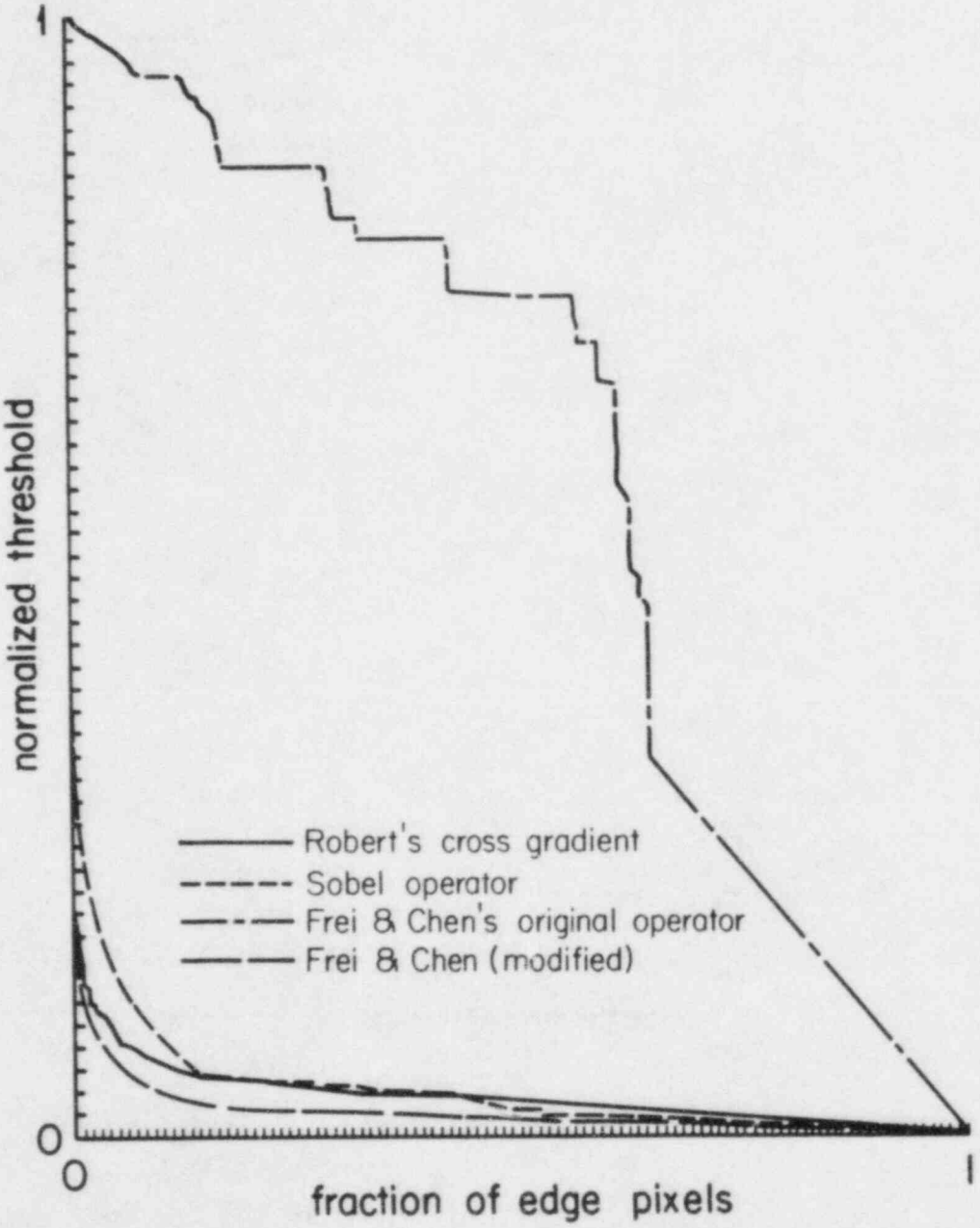


FIGURE 5-4 Sample 1, Test D

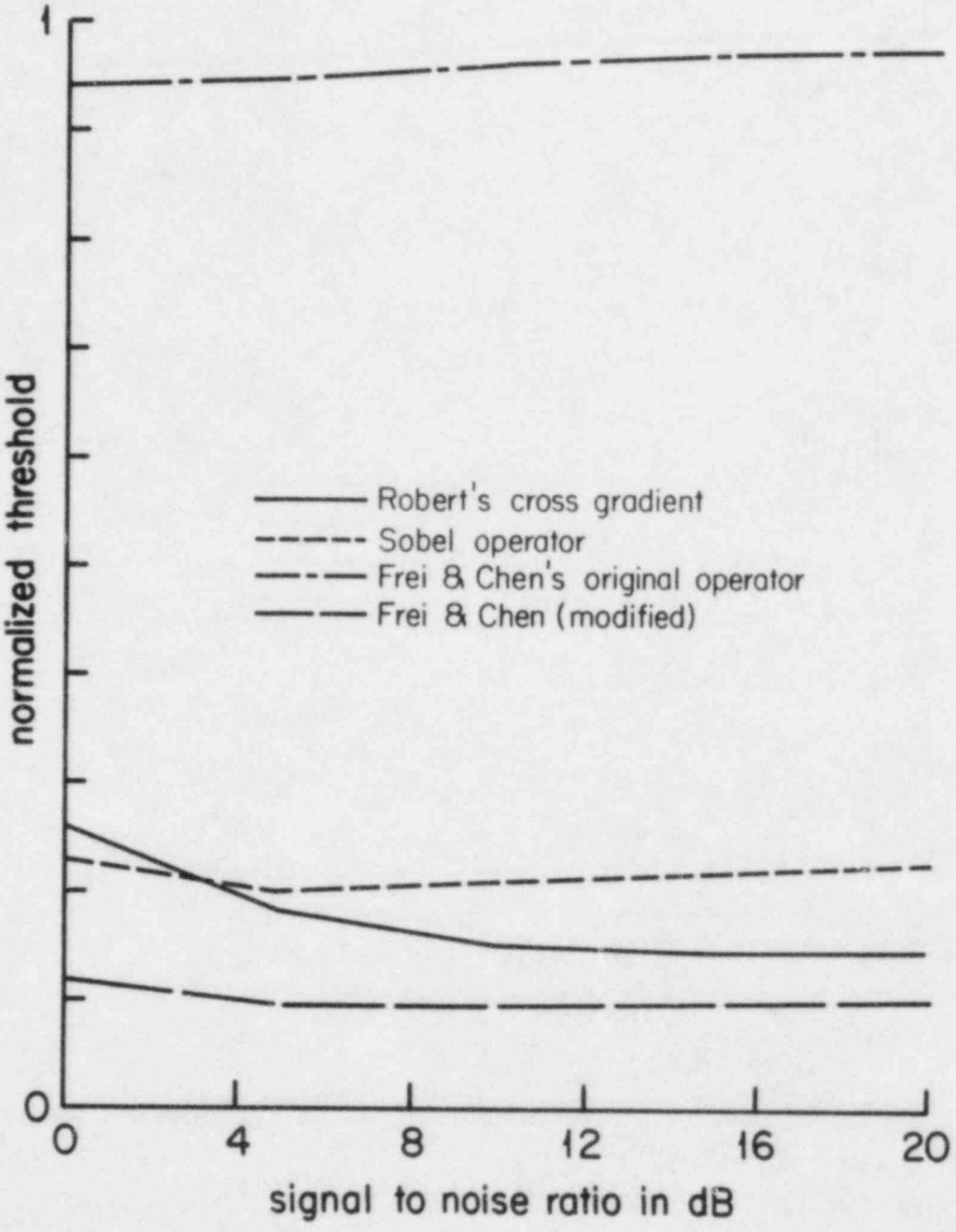


FIGURE 5-5 Sample 2, Test A

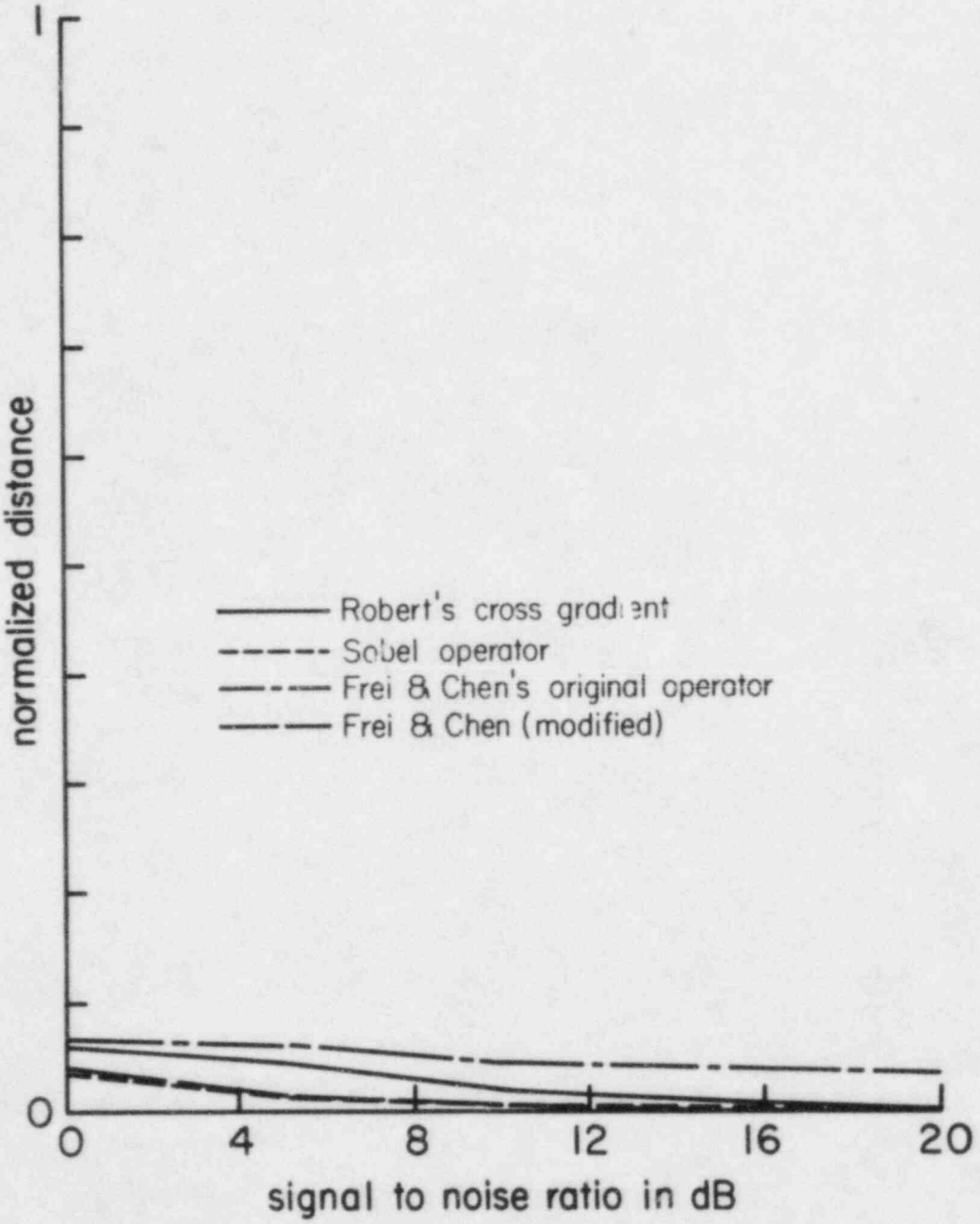


FIGURE 5-6 Sample 2, Test B

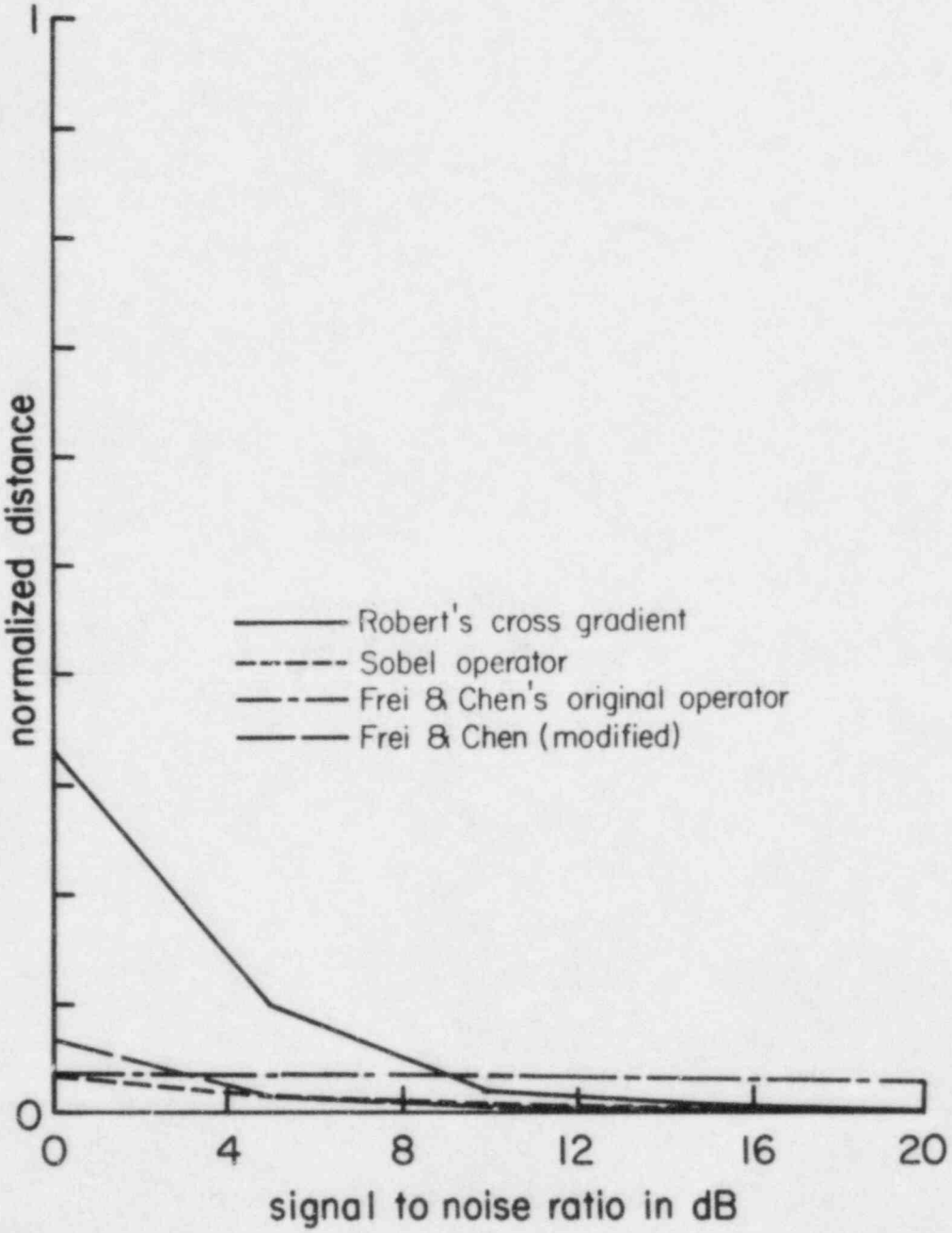


FIGURE 5-7 Sample 2, Test C

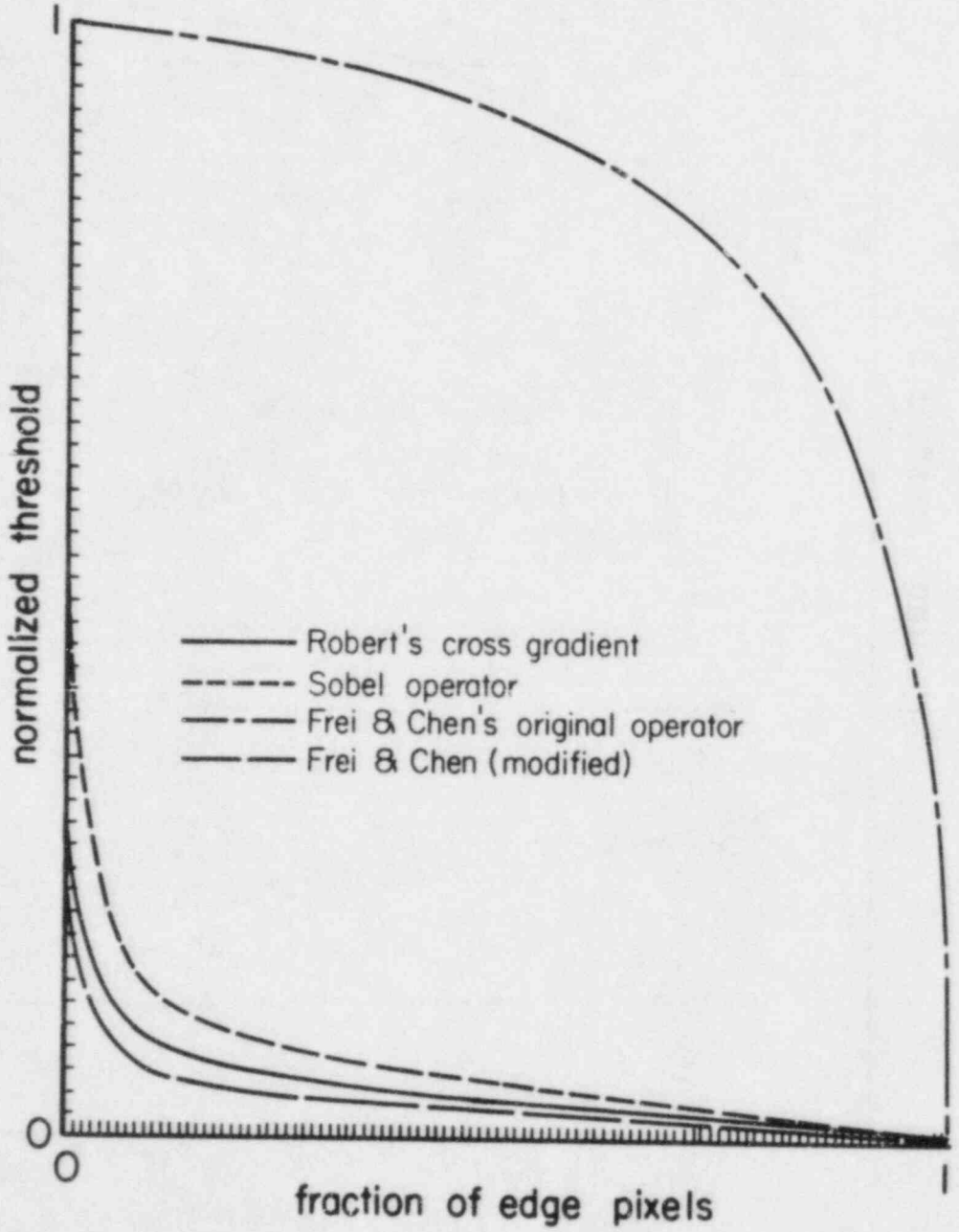


FIGURE 5-8 Sample 2, Test D

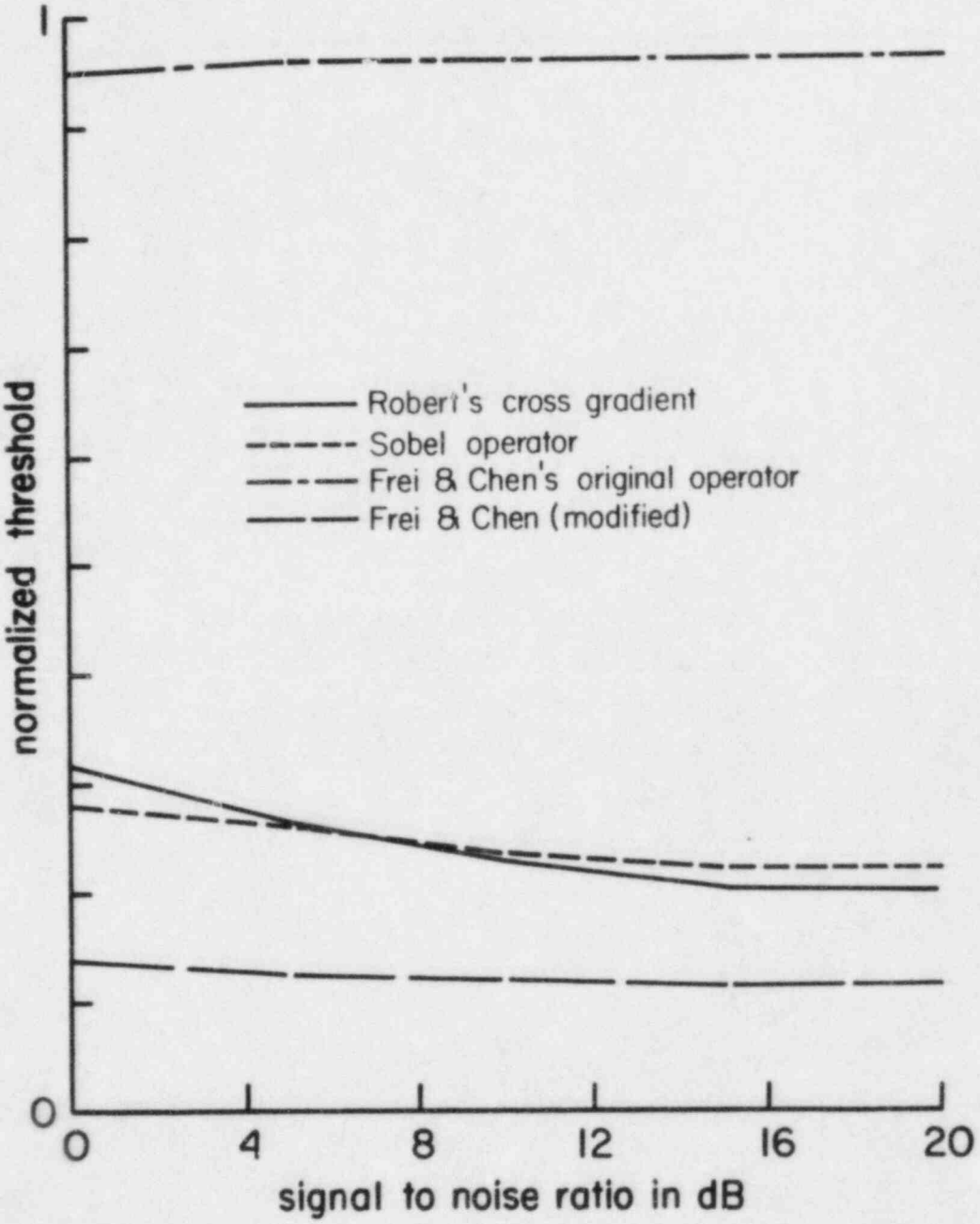


FIGURE 5-9 Sample 3, Test A

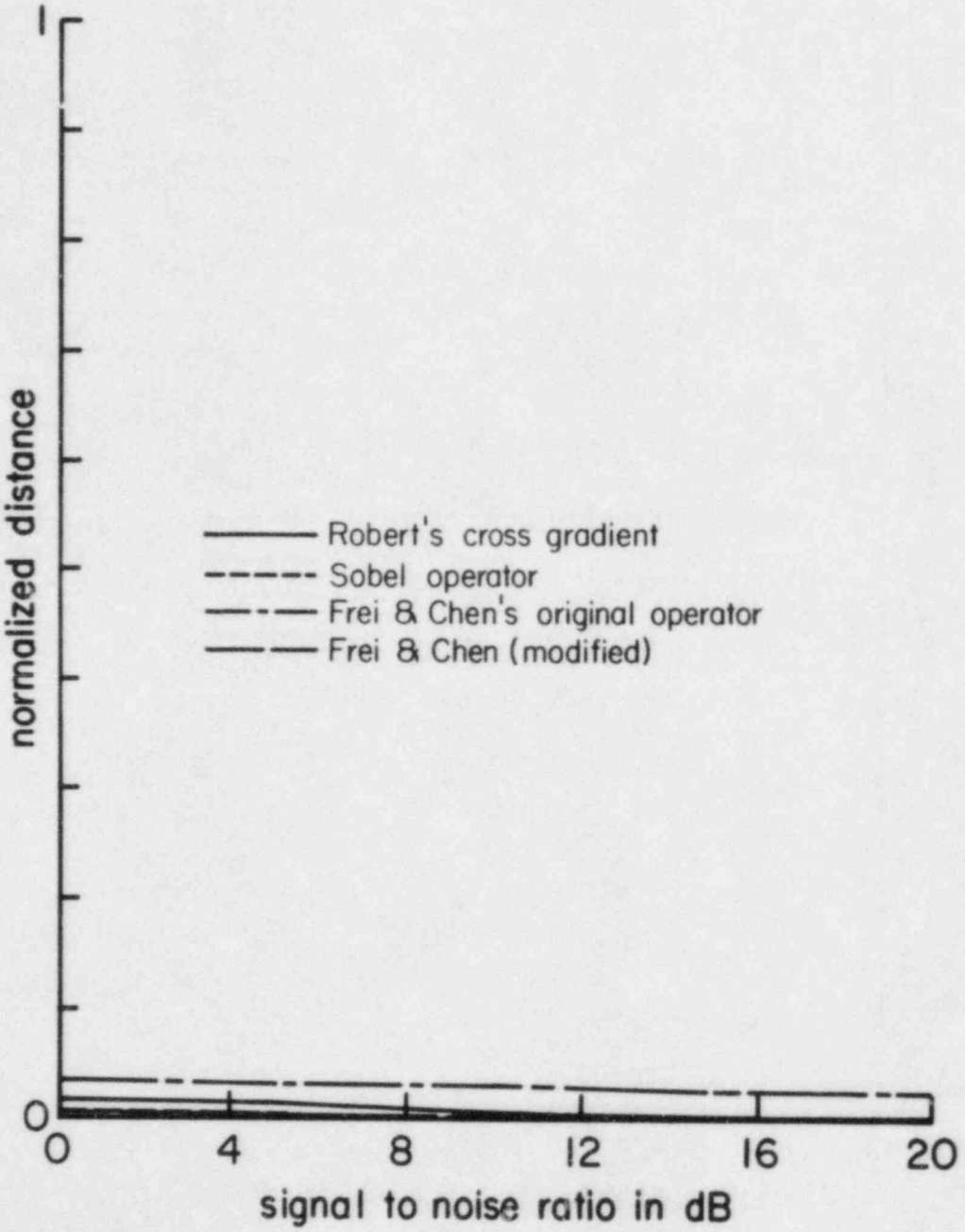


FIGURE 5-10 Sample 3, Test B

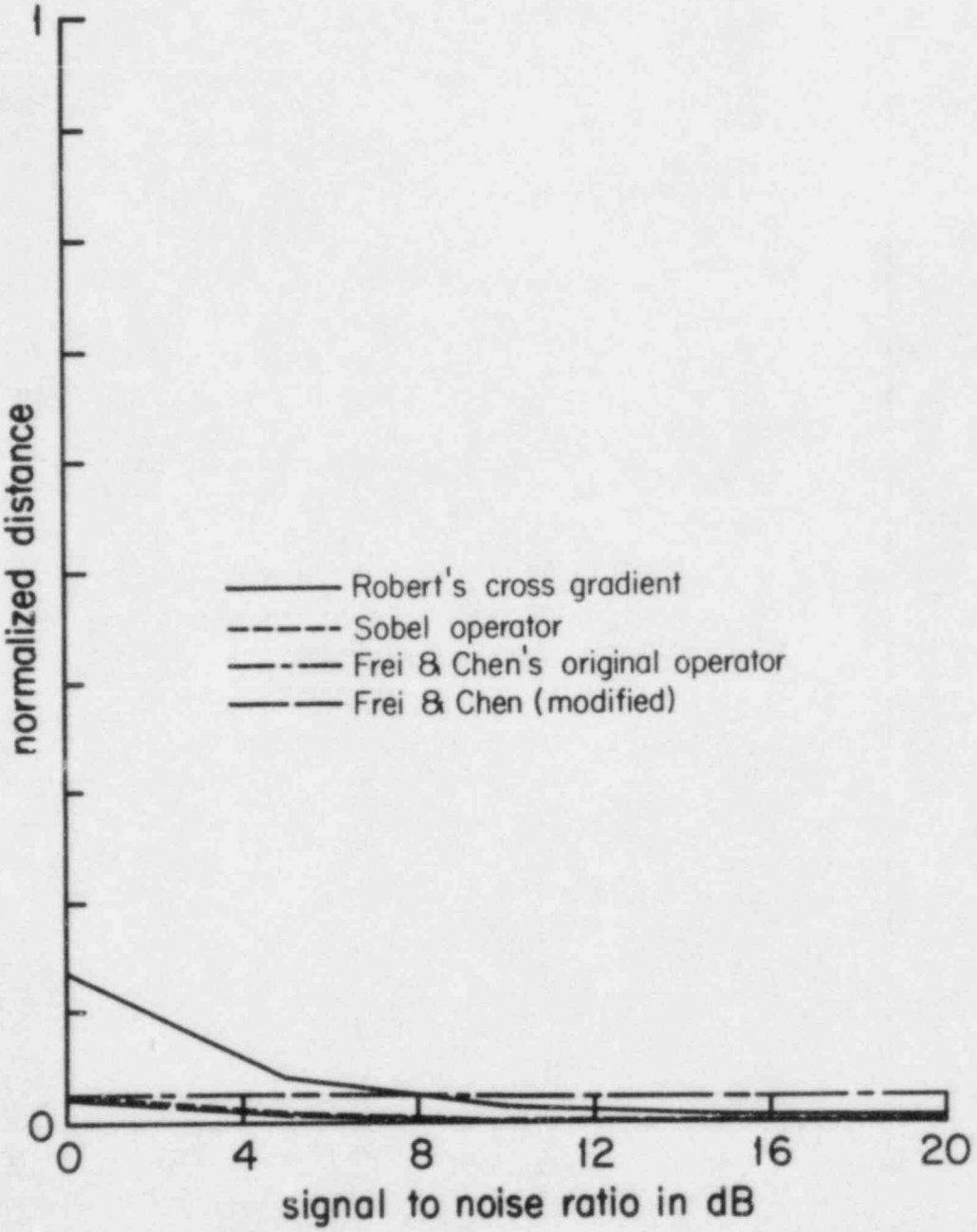


FIGURE 5-11 Sample 3, Test C

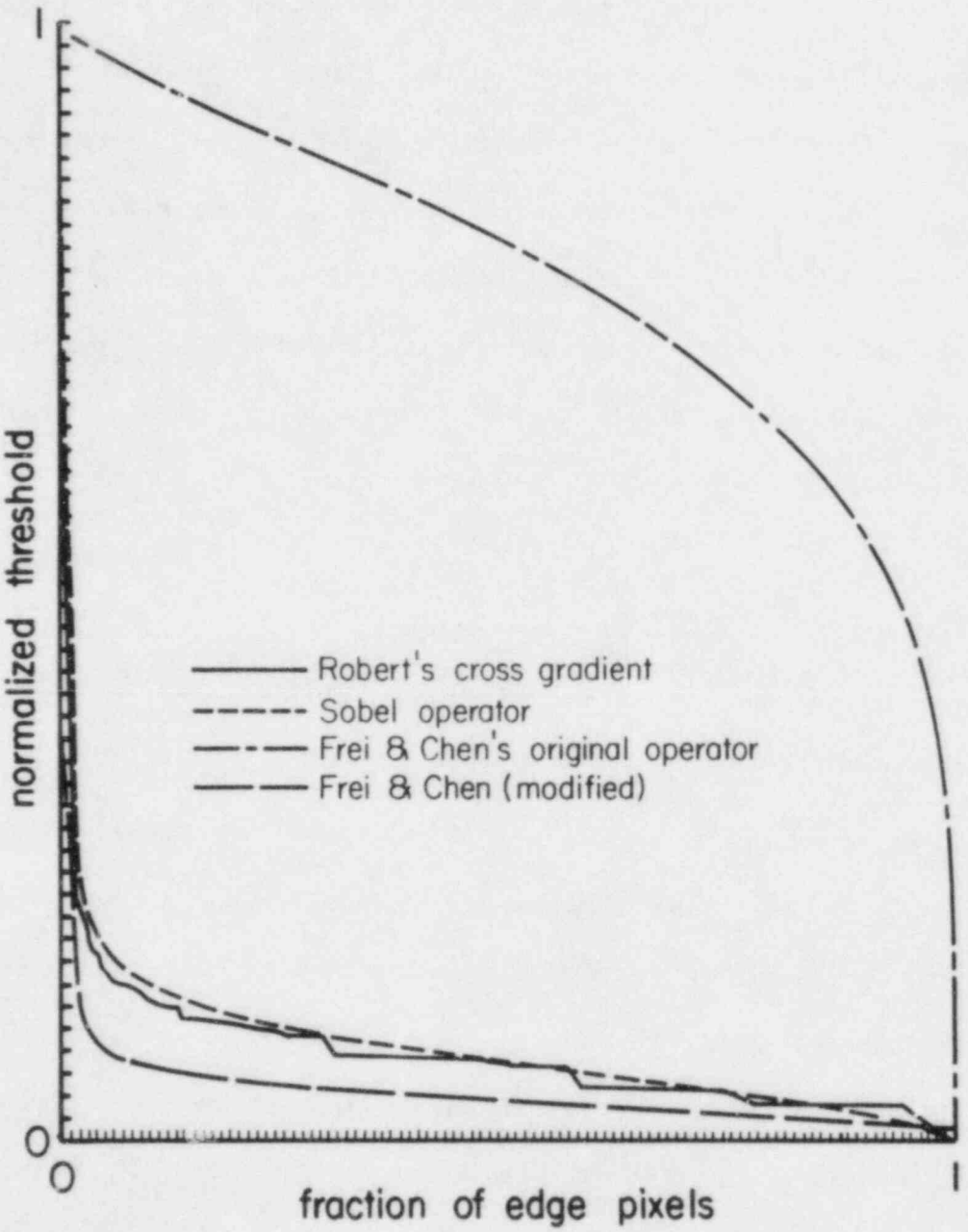


FIGURE 5-12 Sample 3, Test D

5.2.5.2. Threshold Sensitivity Analysis

The number of edge points in the edge map is directly related to the threshold value. It is desirable to have an edge operator which is robust with respect to changes in the threshold value. That is, the binary edge map should not change considerably as the threshold is changed in the range of threshold values used or in the range of the required amount of edge pixels desired. The experiment set up to measure this parameter was to pick an edge operator and an image and then obtain the gradient image (or whatever is to be thresholded). This image is sorted, and a plot of percentage of edge points versus threshold value is drawn. To make the sorting process as fast as possible, the "quick sort" algorithm was used; this is Test D (see Figures 5-4, 5-8, and 5-12).

5.2.5.3. Directional Preference

There were no measurements of this parameter. However, two artificial images were generated for this experiment and the next one. They are shown in Figures 5-13 and 5-14.

Intuitively, these two images are appropriate for measurement of directional preference and even edge thickness. They contain edges along different orientations. Different edge detectors were run with these two images and results were qualitatively compared. It was concluded that all except the Hummel edge detector are more sensitive to horizontal and vertical edge orientations. This is not unexpected as all of them have masks which are designed to detect edges along these two orientations. However, the gradient operator defined by Robert does not have this property as shown in Figures 5-15 and 5-16, where d is the difference between pixels shown by + and −. It is observed that detection of a diagonal edge requires a lower value of threshold. Hummel edge detector is not quite isotropic either. This is due to the fact that the number of basis functions has been limited to four. It can be improved by including more basis functions.

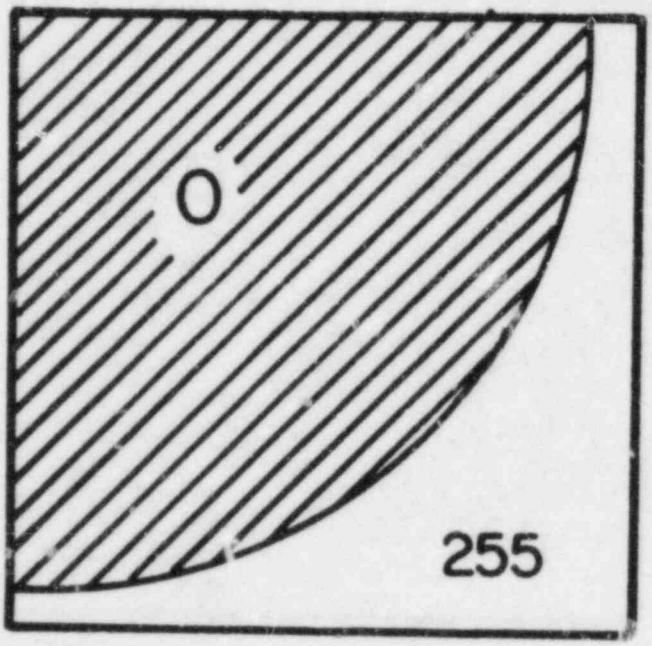


FIGURE 5-13 Test Image for Experiments 3 & 4

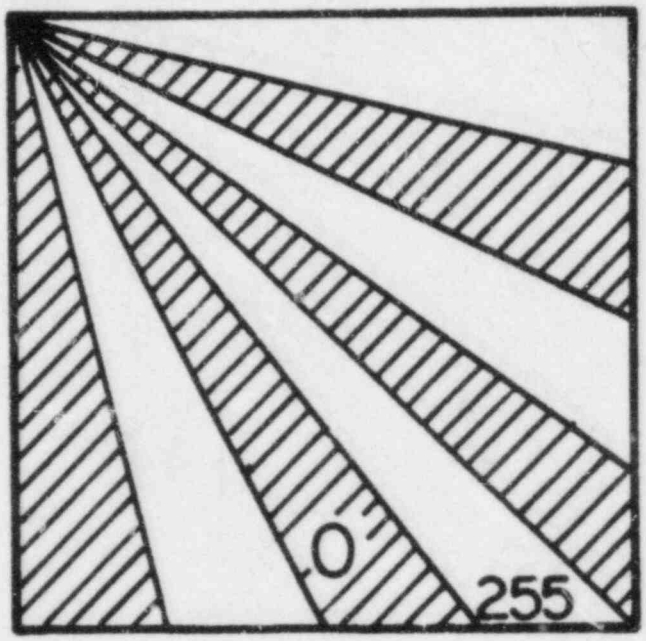


FIGURE 5-14 Test Image for Experiments 3 & 4

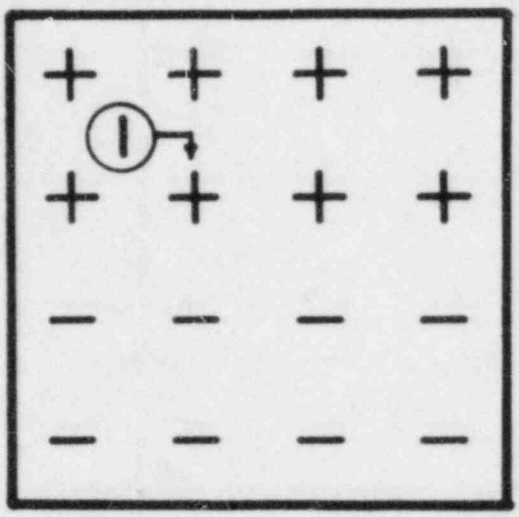


FIGURE 5-15 Horizontal Edge Gradient Value at Point (1) = d $\sqrt{2}$

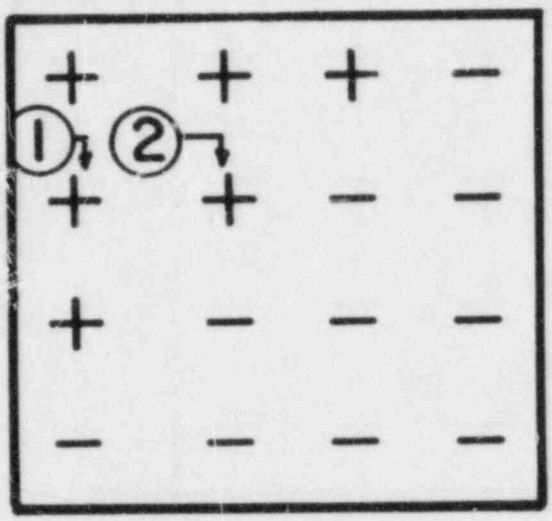


FIGURE 5-16 Diagonal Edge Gradient Value at Points (1) & (2) = d

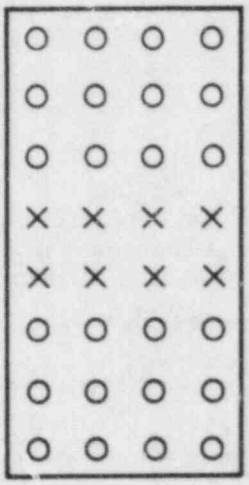


FIGURE 5-17 Subimage for Edge Detection

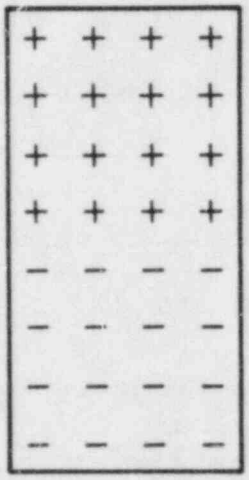


FIGURE 5-18 Values of Sobel Gradient

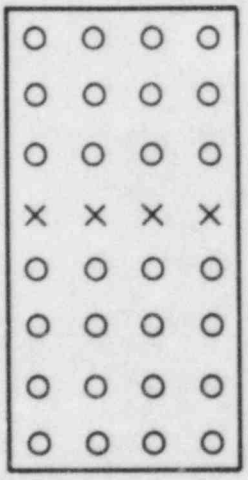


FIGURE 5-19 Values of Robert Gradient

5.2.5.4. Edge Thickness

Again, it is not that easy to define and measure edge thickness, especially because it is a function of the threshold value. Nevertheless, it is not difficult to observe that edges become thicker as the neighborhood size is increased (This is a disadvantage of Hummel edge detector. It is not possible to use a smaller neighborhood because it should be a reasonable approximation to the unit disk). In Figure 5-17 a subimage containing a horizontal step-like edge has been shown. Two different edge operators have been applied to this subimage. In Figures 5-18 and 5-19, X's and O's denote the positions where the gradient values are non-zero and zero, respectively. It is concluded that a thicker edge would be resulted in the case of Sobel edge detector if an appropriate threshold value is not used.

5.3. Conclusions

By looking at the results of the experiments performed and the resulting graphs, a few conclusions can be drawn:

1. In terms of threshold sensitivity, it is desirable to have little change in resulting binary image (edge detected image) when threshold is varied within desired range of operation. Modified Frei & Chen operator has had the best performance. Robert's cross and Sobel operators are slightly worse, but Frei & Chen operator is unacceptable.

2. Noise sensitivity was measured by running three sub-experiments. Initially, the percentage of edge pixels in the binary image was kept fixed and then variations of threshold value used was drawn vs. noise power. Secondly, variations of the binary image vs. noise power was plotted while percentage of edge pixels was held fixed. Finally, keeping the threshold value at that obtained in the first sub-experiment, variations in binary image was plotted against noise power. In all these experiments, it is desirable to have small variation in the

dependent variable of the plot as the independent variable is varied within some operating range.

In the first one, Modified Frei & Chen and Sobel operators turned to be good and quite comparable. Robert's cross was slightly worse, and, once again, Frei & Chen operator did a poor job. In the second sub-experiment, Frei & Chen operator was surprisingly the best. Modified Frei & Chen and Sobel operators performed well and identically, and Robert's cross was somewhat worse than these two. In the last sub-experiment, Frei & Chen operator was again the best among the four. Sobel, Modified Frei & Chen, and Robert's cross had reasonable performances with degradation in the order given, i.e., Sobel operator was the best among them.

3. Very little can be said about directional preference and edge thickness. Frei & Chen operator did not have a reliable performance. The other three were not noticeably different as far as these two features and concerned.

One last factor which has to be taken into account is CPU times required to run each of these operators. The following table shows the CPU times (in seconds) required by a VAX 11/780 computer to process a 256×256 image. It is obvious that processing time increases linearly with the total number of pixels in the image.

Hummel's edge detector takes orders of magnitude of time more than the others to run, and does not have a good performance either. (That operator is suitable for cases where some object on a textured background is to be detected. It really does not fit the kind of application considered here).

Taking all these factors into account, it is concluded that Sobel's edge operator is the most appropriate one for the class of images considered here. It is simple enough to be implemented (even in hardware), and yet it has a good and reliable performance. The negative statements

Edge Operator	CPU seconds
Robert's cross	37
Sobel	45
Modified Frei & Chen	60
Frei & Chen	99

Table 1. Processing Time for Different Edge Operators

about other operators is an indication of the fact that it is not necessary to use an operator any fancier that Sobel operator for this kind of application.

6. Flaw Detection and Classification

6.1. Introduction

One goal of the work performed in the Ultrasonic Imaging Laboratory is to accurately detect and classify flaws in metals through the use of ultrasonic inspection. Most of the effort is devoted to improving means of implementing the Synthetic Aperture Focusing Technique (SAFT) and of interpreting the images created from collections of SAFT-processed A-scans [11]. Although a great deal of progress has been made, there are some difficulties inherent in the SAFT processing (i.e., the requirements for large amounts of data storage and large numbers of calculations and the loss of some of the information contained in raw A-scans). For these reasons, it was decided to test how well (that is, how quickly and how accurately) the detection and classification tasks could be performed with non-synthetic aperture processed scans. This section discusses work that has been done to find solutions using single A-scans. It includes sections on the models developed for A-scan signals, the detection and estimation problems specified in terms of the models, a discussion of how to obtain functions required by the models and possibilities for future work.

6.2. The Detection and Classification Problems

The flaw detection problem is essentially one of picking out the part of an A-scan that is due to the presence of a large reflector. It is difficult because the reflected signals often are buried in noise which is large compared to the signals. Classification involves the estimation of certain parameters which provide enough information about the reflector to specify its nature. Although the two tasks may be thought of as being separate, there is a reasonable approach to stating the problems so that they are related, and in fact, so that the solution of the

classification (estimation) problem is a necessary prelude to a solution of the detection problem. Before describing this approach, we must first model the important components of the A-scan and relate them to the physical conditions under which they are produced. The scan from a single location is denoted $z(t)$ (with transducer location as an implicit argument) and is represented as

$$z(t) = s(t) + n_g(t) + n_a(t) \tag{1}$$

where $s(t)$ is the part of the A-scan due to the reflected and scattered field from the object to be detected (the reflector), $n_g(t)$ is the grain noise due to backscattering of the ultrasonic pulse by the small, randomly located inhomogeneities present in metals, and $n_a(t)$ is the ambient (electronic) noise. Each of these components will now be discussed.

6.2.1. The signal $s(t)$

Assume that a coordinate system is given. Let X be the transducer surface (i.e., $X = (\vec{r} \mid \vec{r}$ is a transducer surface point)). Let $\tilde{g}(t;\vec{r}_x^{\,1}), \vec{r}_x^{\,1} \in X$, be the amplitude of the normal displacement vector for surface point $\vec{r}_x^{\,1}$ at time t. In response to this displacement, a pressure field is created in the material. If $p_f(t;\vec{r})$ is the component of the field at a point $\vec{r}$ due to the forward propagating waves, it can be represented as the integral over X of the output of a linear, time-invarient filter between $\vec{r}_x^{\,1}$ and $\vec{r}$ when the input is $\tilde{g}(t;\vec{r}_x^{\,1})$. That is,

$$p_f(t;\vec{r}) = \int_X \int_{-\infty}^{\infty} h_{FX}(t-\tau;\vec{r},\vec{r}_x^{\,1})\tilde{g}(\tau;\vec{r}_x^{\,1})\,d\tau\,d\vec{r}_x^{\,1} \tag{2}$$

Now, let S denote the unshadowed part of the reflector surface, so that $S = (\vec{r} \mid \vec{r}$ is an unshadowed reflector surface point), and let $h_{ss}(t;\vec{r}_s,\vec{r}_s^{\,1})$ be the impulse response of a filter corresponding to the reflector, relating the incident and reflected waves at different surface points. Then $p_B(t;\vec{r}_s)$, the part of the field at a reflector surface point $\vec{r}_s$ due to the wave reflected and scattered at the surface, is

$$p_B(t;\vec{r}_s) = \int_S \int_{-\infty}^{\infty} h_{ss}(t-\tau;\vec{r}_s,\vec{r}_s^{\,1}) p_f(\tau;\vec{r}_s^{\,1})\, d\tau d\vec{r}_s^{\,1} \tag{3}$$

Similarly, if $h_{XB}(t;\vec{r}_x,\vec{r}_s)$ is the impulse response of a filter relating the amplitude of the normal displacement of the transducer surface at $\vec{r}_x$ to the value of the reflected field at $\vec{r}_s$ and if $h_X(t;\vec{r}_x)$ is the impulse response of the transducer in the receiving mode to a displacement at $\vec{r}_x$,

$$s(t) = \int_X \int_{-\infty}^{\infty} h_X(t-\tau;\vec{r}_x) p_x(\tau;\vec{r}_x)\, d\tau d\vec{r}_x \tag{4}$$

where $p_x(\tau;\vec{r}_x)$ is the amplitude of the displacement of the surface at $\vec{r}_x$ at time τ and is given by

$$p_x(t;\vec{r}_x) = \int_S \int_{-\infty}^{\infty} h_{XB}(t-\tau;\vec{r}_x,\vec{r}_s) p_B(\tau;\vec{r}_s)\, d\tau d\vec{r}_s \tag{5}$$

So, the model for the signal due to a reflector when the transducer output is $\tilde{g}$ is given by

$$s(t) = \int_X \int_{-\infty}^{\infty} h_X(t-\tau;\vec{r}_x) \int_S \int_{-\infty}^{\infty} h_{XB}(\tau-u;\vec{r}_x,\vec{r}_s) \int_S \int_{-\infty}^{\infty} h_{ss}(u-v;\vec{r}_s,\vec{r}_s^{\,1}) \tag{6}$$

$$\int_X \int_{-\infty}^{\infty} h_{FX}(v-w;\vec{r}_s^{\,1},\vec{r}_x^{\,1}) \tilde{g}(w;\vec{r}_x^{\,1})\, dw d\vec{r}_x^{\,1}\, dv d\vec{r}_s^{\,1}\, du d\vec{r}_s\, d\tau d\vec{r}_x \tag{7}$$

Several reasonable assumptions may be made to simplify this expression. They include:

[a] At a given value of t, h_x and $\tilde{g}$ are constant across X; that is,

$$\begin{aligned} h_x(\vec{t};r_x) &= h_x(t), \quad \vec{r}_x \in X \\ &= 0, \qquad \text{elsewhere} \end{aligned} \tag{8}$$

and

$$\begin{aligned} \tilde{g}(t;\vec{r}_x) &= \tilde{g}(t), \quad \vec{r}_x \in X \\ &= 0, \qquad \text{elsewhere} \end{aligned} \tag{9}$$

Define

$$g(t) = \int_{-\infty}^{\infty} h_x(t-\tau)\tilde{g}(\tau)d\tau$$

$$h_F(t;\vec{r}_s^{\,1}) = \int_X h_{FX}(t;\vec{r}_s^{\,1},\vec{r}_x^{\,1})d\vec{r}_x^{\,1}$$

and

$$h_B(t;\vec{r}_s^{\,1}) = \int_X h_{XB}(t;\vec{r}_x,\vec{r}_s)d\vec{r}_x$$

Then, by integrating over $\vec{r}_x$ and $\vec{r}_x^{\,1}$, interchanging the order of integration and changing the variables of integration, the signal can be written as

$$s(t) = \int_{-\infty}^{\infty}\int_S h_B(t-\tau;\vec{r}_s)\int_{-\infty}^{\infty}\int_S h_{ss}(\tau-u;\vec{r}_s,\vec{r}_s^{\,1})\int_{-\infty}^{\infty} h_F(u-v;\vec{r}_s^{\,1})g(v)dv d\vec{r}_s^{\,1}du d\vec{r}_s\, d\tau \qquad (10)$$

[b] The reflected wave at a surface point depends only on the wave impinging on that point (this is consistent with reflection and diffraction models of the reflector return waves). Then $h_{ss}(t;\vec{r}_s,\vec{r}_s^{\,1})$ can be written as $h_{ss}(t;\vec{r}_s)\delta(|\vec{r}_s-\vec{r}_s^{\,1}|)$. Define $h_{FB}(t;\vec{r}_s) = \int_{-\infty}^{\infty} h_B(t-\tau;\vec{r}_s)h_F(\tau;\vec{r}_s)d\tau$. Then, by integrating over $\vec{r}_s^{\,1}$ and interchanging the order of integrations,

$$s(t) = \int_{-\infty}^{\infty} h_{ss}(t-\tau;\vec{r}_s)\int_{-\infty}^{\infty} h_{FB}(\tau-u;\vec{r}_s)g(u)du d\vec{r}_s\, d\tau \qquad (11)$$

[c] $h_{FB}(t;\vec{r}) = 0$ for $\vec{r}$ not in C, where C is the "come of insonification." Then the integration over S can actually be performed over $E = S \cap C$. Let $p(t;\vec{r}) = \int_{-\infty}^{\infty} h_{FB}(t-\tau;\vec{r})g(\tau)d\tau$; because of the random nature of the transmission paths to and from a point $\vec{r}$, $p(t;\vec{r})$ is actually a random field. We assume that, for a given $\vec{r}$, the statistics of $p(t;\vec{r})$ depend only on the average distance from $\vec{r}$ to the transducer surface; that is, if $D(\vec{r}) = \int_X |\vec{r}-\vec{r}_x|\,d\vec{r}_x$, quantities such as the mean and autocorrelation functions of the process $p(\cdot;\vec{r})$ are functions of $D(\vec{r})$. Further, in those materials which

have a well-determined structure with only small irregularities which have negligible effect on the propagating wave (as is often the case with aluminum, for example), $p(t;\vec{r})$ is deterministic and can be written as $p(t;D(\vec{r}))$ for $\vec{r} \in C$. For most of these materials, h_{FB} is not very sensitive to small change in $D(\vec{r})$, and inspection is often performed at a distance from the transducer that is large compared to the change in distance across the reflector surface. So, the assumption can usually be made that over E, $D(\vec{r}_e) \equiv a$ constant r, the range or depth of the reflector. Define $h_e(t) = \int_E h_{ee}(t;\vec{r}_e)d\vec{r}_e$; then, noting the dependence on the reflector response explicitly in its argument, the signal term can be represented as

$$s(t;h_e) = \int_{-\infty}^{\infty} h_e(t-\tau) \int_{-\infty}^{\infty} h_{FB}(\tau-u)g(u)du\,d\tau \tag{12}$$

when the material is deterministic (case I) and as

$$s(t;h_{ee}) = \int_{-\infty}^{\infty} \int_E h_{ee}(t-\tau;\vec{r}_e) \int_{-\infty}^{\infty} h_{FB}(\tau-u;\vec{r}_e)g(u)du\,d\vec{r}_e\,d\tau \tag{13}$$

when the material is random (case II).

In the equations, $g(u)$ is known; it is equal to the front surface response. The function h_{FB} is the impulse response of a filter representing the forward and back transmission path through the metal. Under case I, it is a deterministic function and can be represented as the Inverse Fourier transform of $e^{-\gamma(f,r)}$ where $\gamma(f,r) = \alpha(f,r) + j\beta(f,r)$, with $\alpha(f,r)$ being the frequency- and depth-dependent attenuation coefficient and $\beta(f,r)$ the phase shift which includes the dispersive effects of frequency- and depth-varying sound speeds. In this case, $s(t)$ is a deterministic function given r and the reflector response. Methods for calculating this function are discussed in Appendix C. Under case II, $p(t;\vec{r})$ as a sort of multipath process; it is assumed to be Gaussian, so that in this case, $s(t)$ is also a Gaussian random process. The functions $h_{ee}(t;\vec{r})$ and its integrated version $h_e(t)$ characterize the reflector, and the estimate for

their values provides information available to help solve the classification problem. This information is in the form of parameters that specify the reflector response and, it is hoped, specify the important physical features of the reflector.

One aspect of the classification problem which is itself a difficult task is to determine what physically constitutes the difference between the classes of reflectors in which we are interested (such as cracks and voids) and those about which we do not care (such as geometrical reflectors), and how these differences translate into differences in the parameterized impulse response. The geometrical features which are most troublesome are associated with the back surface near weld boundaries [12,13]. The weld counterbore is a planar surface that poses problems because its location and angle are uncertain. The weld root bead is a rough planar reflector with an unknown surface shape. Both of these cause difficulties because their echo signals can be misinterpreted as being due to flaws. Unfortunately, most true flaws also originate near welds, and the most severe sources of failure occur at the back surface. In the base material, planar fatigue cracks are the main problems, while in austenitic welds, the most prevalent flaws are intergranular stress corrosion cracks (IGSCC). These flaws are more irregular than fatigue cracks because they follow grain boundaries, and thus cannot be accurately represented by single planar surfaces. Another class of serious defects includes voids [14]. These are "gaps" in the metal which, near welds, often are due to incomplete fusion of the weld material, either with the weld sidewalls or in the interior of the weld itself. These flaws may be approximately represented by cylinders or ellipsoids. Less serious defects that also have approximately an ellipsoidal shape are slag inclusions and porous regions.

Many papers have been published that deal with the relationship between an object's geometry and the field which it scatters. Several methods rely on low-frequency, weak-scatterer, or smooth-surface restrictions which do not apply to our case of interest. One method of

parameterizing the reflector response by a series of amplitudes and time delays is described for sonar targets in [15]. This leads to some reasonable detector implementations, but it is not certain whether these parameters provide enough information in a classifier for ultrasonic reflectors. Another method which is potentially useful is that of resonance theory [16]. In this theory, resonances occur in the reflector response that are related to the circumnavigation of the surface by creeping waves. As the physical descriptions indicate, the main difference between flaws and geometrical reflectors seems to be in the closure of the reflector surface. All of the major flaw types consist of closed or nearly closed surfaces. On the other hand, geometrical reflectors are large surfaces that are not so much enclosures of a small region of acoustic properties which differ from the medium's as they are open boundaries between two different media. It seems reasonable to expect stronger resonance terms in the flaw response than in responses of geometrical reflectors. Of course, many questions must be answered, including how variations in viewing angle and surface roughness affect the resonances, how easily the presence of resonance terms can be verified in noisy A-scans, and perhaps most importantly, how to incorporate resonance parameters in a detector/estimator formulation. More research into the possible applications of resonance theory to the problem should be performed.

6.2.2. The grain noise

This term is due to backscattering toward the transducer of the forward propagating ultasonic wave by small inhomogeneities in the metal. As such, it depends on the transducer output $g(t)$, and the component of the grain noise from a given region in the metal depends on the metal's properties in that region as well as the forward and back metal path filter for that location. In some cases, specifically those for which the first order Born approximation is valid (which implies that case I holds), the model for the backscattered signal derived in [17] can be used as outlined in Appendix A to specify the grain noise as

$$n_g(t) = \int_{R_0}^{R_1} A(r) \int_{-\infty}^{\infty} h_{FB}(t-\tau;r)g(\tau)d\tau dr$$

where $A(r)$ is a random process corresponding to an integration across the insonified region at depth r of a function of the random fluctuations in material properties (namely, compressibility and density). The situation is less clear in large grain materials. As a practical matter, the noise is assumed to be the sum of a large number of independent scattering terms, and the central limit theorem is invoked to justify assuming it to be Gaussian, with an empirically estimated autocorrelation function. Because the time dependence of the propagating wave continually changes with depth and because scattered waves from deeper in the material take longer to reach the transducer than do those from near the surface, the components of the grain noise at different times have different shapes, and the noise is non-stationary. As indicated in Appendix A, however, it often can be represented as nearly wide-sense stationary over intervals shorter than the entire scan length. An example of depth-varying estimated noise autocorrelation is given in Appendix C.

6.2.3. The ambient noise

This is the only component of the A-scan that varies with time of inspection. It is very nearly white noise, so it can be reduced by the simple procedure of averaging together several A-scans collected at one location. If N scans are averaged, the variance of the ambient noise in the averaged scan will be reduced from that of the single-inspection scan by a factor of N. In practice, N is on the order of 10 to 100, and the resulting $n_a(t)$ has much less bearing on the detection and estimation problem than does the grain noise.

6.2.4. Detection and estimation

With the A-scan components specified, the flaw detection problem can be stated as one of deciding between two hypotheses:

$$H_0 : \; z(t) = n(t)$$

$$H_1 : \; z(t) = n(t) + s(t;\alpha)$$

where $t \in T$, which is either a set of discrete points or an interval; $n(t) = n_g(t) + n_s(t)$ is mean-zero Gaussian noise with a known (either through calculation or empirical estimation) autocorrelation function $En(t)n(t) = R(t,\tau)$; $\alpha = (\alpha_1, \ldots, \alpha_m)$ is a set of parameters characterizing the reflector response; and $s(t;\alpha)$ under case I is a known function given α, while under case II, it is a Gaussian random process with an autocorrelation function $Q(t,\tau;\alpha) = Es(t;\alpha)s(\tau;\alpha)$ parameterized by α. In this form, the problem can be approached via the methods of maximum likelihood detection and estimation. The generalized likelihood ratio based on the observation of $z(t)$ for T consisting of N points is defined as

$$L(z_1, \ldots, z_N) = \sup_{\alpha \in A} \frac{f_1(z_1, \ldots, z_N;\alpha)}{f_0(z_1, \ldots, z_N)} \tag{14}$$

where A is the set of values over which α can vary, f_1 is the N-variate probability density function for z under H_1 and f_0 is the N-variate probability density function for z under H_0. The test for whether or not H_1 holds is to compare $L(z_1, \ldots, z_N)$ to a threshold, while the associated maximum likelihood estimate of α is that value which maximizes $f_1(z_1, \ldots, z_N;\alpha)$, and this estimate provides the basis for a solution of the classification problem. Theoretical extensions of the definition of the likelihood ratio to the case of continuous signals (which are sometimes useful for contributing to the design of filters for the A-scan prior to sampling) exist and are described for known signals in noise (case I) in [19] and for noise-in-noise problems (case II) in [20]. Some arguments for desirability of using maximum likelihood estimates of the parameters show that solutions exist under fairly general conditions, that the estimates are

invariant and often asymptotically efficient, and that they converge in probability to the true values as the integrated signal-to-noise ration increases [19,21]. Of course, the fact that solutions exist does not guarantee that they will be easy to find or practical to implement. Subject to some assumptions about the grain noise (mainly, the local wide-sense stationarity condition discussed in Appendix A), reasonable algorithms have been developed for detecting objects consisting of one- or two-point reflectors under case I (outlined in Appendix B, as is a solution to the problem of finding a point reflector under case II). Detection with the two-point reflector model is especially interesting because several classification and sizing algorithms are based on it [22, 23]. These detectors can be fairly easily generalized to include target models like those in [15], as long as the number of unknown parameters remains small. Solutions involving different parameterizations are now being investigated, along with means of ensuring adequate detector/estimator performance under the range of operating conditions found in practice.

6.3. Conclusion

As was stated earlier, the goal of this task was to describe means of detecting and classifying flaws using raw A-scan data. Most of the work done involved developing a model for the A-scan signals which would be broad enough to include most of the cases of interest while still allowing for reasonably implementable procedures. Using this model, algorithms for detection and classification (i.e., parameter estimation) have been described for some simple parameterizations. The evaluation of the performance of these algorithms on actual test specimens and the development of algorithms for other parameterizations of interest (such as in terms of resonant frequencies) remain as tasks to be completed in the future.

Two other possibilities for future work should be mentioned. First, the success of the detector and estimator depends in large part upon the accuracy of the model. In order to iso-

late the effects of interest (i.e., the flaw characteristics), the one developed here assumes that other function, such as the grain noise autocorrelation and the impulse response of the filter representing the effects of transmission through the material, are known. Some simple ways of estimating these functions from observed A-scans have been mentioned; more sophisticated techniques would be expected to give more accurate results. Better still would be the development of a more "phenomenological" approach, allowing the calculation of these functions from known material properties. This is especially true for substances like cast stainless steel, for which the assumptions most often employed in making empirical estimates (particularly the independence of the noise and signal) are not valid.

The second possibility is research into ways of accounting for uncertainty in the models. It is especially true in the case of ultrasonic inspection, in which the material characteristics can vary widely from spot to spot on the same specimen and in which the signal parameterizations are always necessarily incomplete, that any model will contain inaccuracies. The design of "robust" tests insensitive to small inaccuracies and of adaptive test making use of observations to refine the model would be valuable in helping to make non-SAFT detection and classification procedures more effective.

7. Development of SAFT Processor Prototype

7.1. Introduction

The synthetic aperture focusing technique (as applied to ultrasonic testing) *SAFT/UT) requires that raw data be collected over an arbitrarily selected aperture volume. The data samples are then processed to coherently sum contributions to a point in the volume from every other point in the aperture volume that could have received energy from the point being processed. Figure 7-1 illustrates the procedure. In Figure 7-1a, a focused transducer is shown with its focal point located at the surface of an inspected volume and insonifying a beam with included angle θ after refraction. As illustrated, a point on the axis of the beam will return energy to the transducer. After moving laterally, the beam sees the same point again but with a time shift due to the change in range from the transducer focal point to the object.

All points within the transducer beam that can see an artifact, will lie on a hyperbolic sheet. Figure 7-1b, shows that a digitized A-scan will relate to an A-scan lying at some other radius with a time displacement which can be resolved into an integer number of samples deeper than the central A-scan sample. Thus, a knowledge of the geometry can be used to create a table of control values which control the displacement of all contributions to the central A-scan. If the contributions are truly related to the reflection target in the central A-scan, they will add coherently and serve to provide a significant improvement in resolution and signal-to-noise ratio of the target. If they do not belong to the processed point they will not correlate and will average to a near zero contribution. An important feature of the synthetic aperture technique is that it results in an infinite focal depth for all points which can be processed.

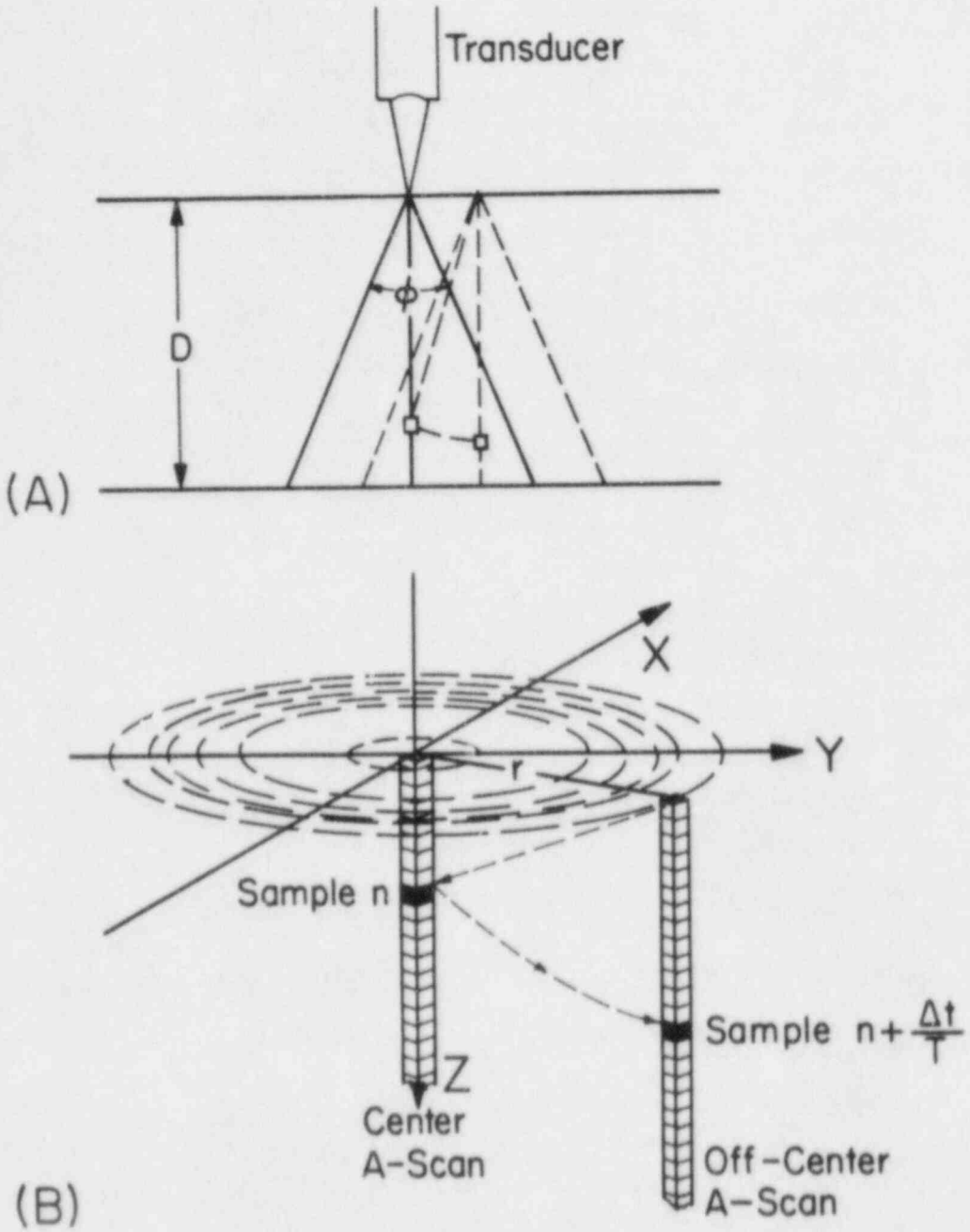
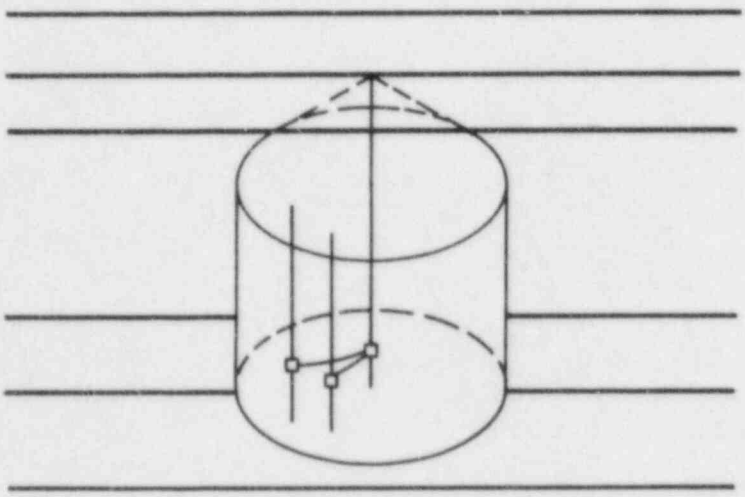
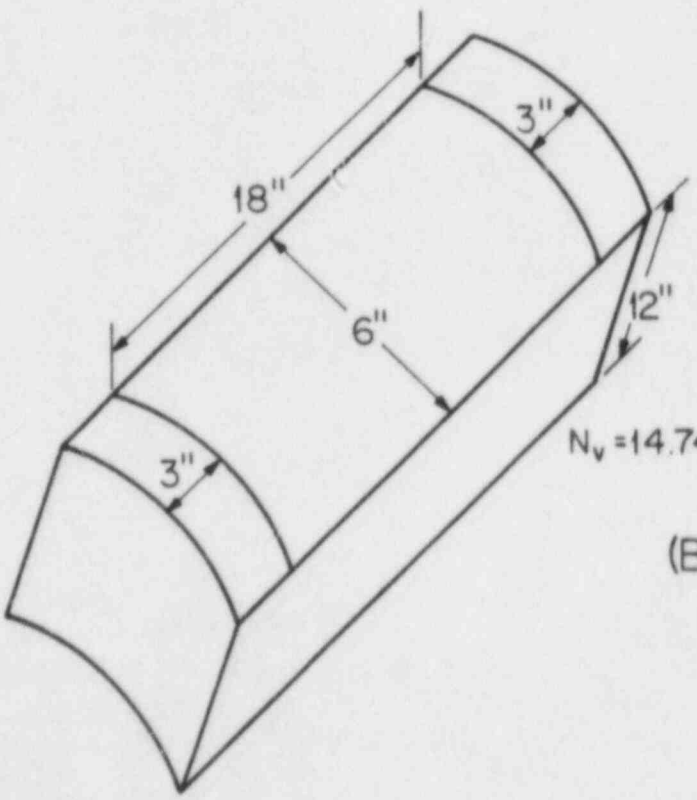


FIGURE 7-1 SAFT Correlation Procedure



(A)



$N_v = 14.74 \times 10^6$ points storage

(B)

FIGURE 7-2 SAFT Aperture Volume

Figure 7-2 illustrates that the aperture volume considered in the hardware design is arbitrarily limited due to consideration of practical limitations on processed points. It also shows many points do not fall within the beam solid near the entry plane of the beam and will be discarded since they do not contribute to the SAFT process. The resultant processed volume is thus a cone truncated at a arbitrary radius to a right cylinder.

In a practical situation, consider a reactor pressure vessel with wall thickness up to 12", and a girth weld that requires an 18" wide scan and a aperture radius of 3". The volume scanned out by this set of dimensions is represented by Figure 7-2b. As illustrated, an extra 3" band at each end of the scan is required to provide good data to the edge of the processed volume. Thus, a 24" scan across the face of the weld will result in a processed volume that is 18" long by 6" wide and 12" thick. With data points taken at every 0.1" in both X and Y dimension and digitized to approximately 1,000 samples in dept', a total of approximately 14.74 million points of raw data must be stored to represent this volume.

Figure 7-3 addresses the reason why a special hardware processor is called for. The basic kernel of processing action which must be repeated for each processed point and each step is shown. Addresses must be incremented to determine the next off-center A-scan. A shift value is determined either by calculation or table look-up. Finally the off-center A-scan. A shift value is determined either by calculation or table look-up. Finally the off-center point is added to the center point. This loop must be performed for each vertical sample of an A-scan for every shifted component of all off-center A-scans which can contribute. A maximum of close to 3,000 points can be involved in the processing of each single center A-scan point when processing in the full 6" diameter

CONVENTIONAL SOFTWARE
COMPUTATION KERNEL

DETERMINE NEXT OFF-CENTER
A-SCAN

DETERMINE SHIFT VALUE

FETCH OFF-CENTER VALUE

ADD OFF-CENTER POINT TO
CENTER POINT
(MEMORY ADD)

INNER LOOP PROCESSING TIME

	T/μs	in.^{3}/min. (estimate)
INTERDATA 7/32	14.75	0.24
PERKIN-ELMER 3240	6.94	0.5
DEC VAX 11/780	5.5	0.6
PRIME 750	4.5	0.8
FP5-120B*	0.5	7
CRAY-1	0.1	35
SAFT PROCESSOR	0.035	100

* must have a host mainframe

FIGURE 7-3 Steps in Basic Processing Kernel and Comparisons of Processing Rates of Various SAFT Implementations

aperture. Figure 7-3 illustrates the processing capability of a number of modern available central processor units compared with the performance goal of the SAFT processor. It can be seen that the VAX 11/780 can only process about 0.6 cubic inches per minute and the CRAY-1 (costing over 8 million dollars) can get up to approximately 35 cubic inches per minute, whereas the SAFT processor with inexpensive memory can reach the order of 100 cubic inches per minute, and, with somewhat faster and more expensive memory, can reach the 200 cubic inches which is the operating goal of the user.

The Fieldable Real-Time SAFT System

Figure 7-4 displays the major components of a typical fieldable SAFT real-time system. At the upper right of the figure a scanner system is shown consisting of a servo controlled scanner drive with a transducer or array of transducers. This assembly is driven mechanically under control of a microprocessor-based controller obeying a scan program transmitted to it from a host computer. Physically near the transducer elements are the electronics consisting of a pulser, receiver, (multiplexer if an array is used), and analog-to-digital and averaging facilities. The latter may be placed remotely at the far end of the cable connecting this equipment with the host. In the figure, the communication is indicated as digitized RF signals on a Unibus path. This could also be configured as a properly designed coaxial cable for the analog signal path with digitizing facilities located at the host computer end of the cable. A host computer depicted as a VAX 11/780 is typical of a central CPU facility capable of handling the overall supervisory and file management tasks of the fieldable system. The usual peripherals consisting of mass memory, printer, CRT terminal, and graphics terminal are shown connected through Unibus facilities. Also connected by Unibus is the SAFT processor hardware shown within the dotted enclosure. This consists of a process controller (presently

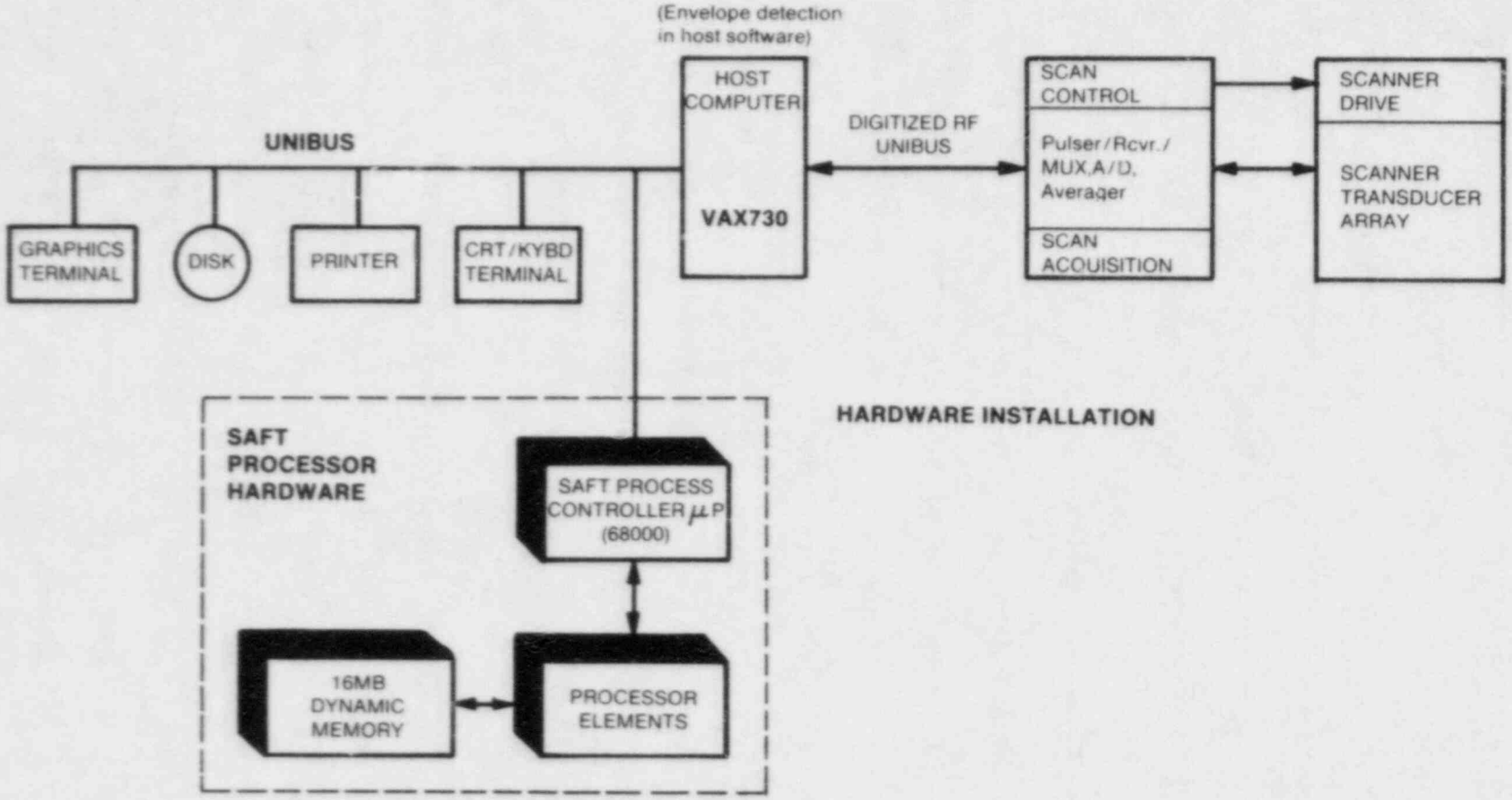


FIGURE 7-4 Typical Fieldable SAFT Real-Time System

designed around the Motorola 68000, 16 bit microprocessor) with appropriate program, data memory and peripherals. This controller manages communication between the SAFT hardware and the host. Shown connected to the controller are the actual processor elements comprised of a parallel processor controller board and a number of parallel processing elements. Typically, 16 processor elements are contemplated to handle the SAFT processing task with the required speed. Finally, a 16-megabyte dynamic memory is dedicated to the parallel processor assembly and holds 1 maximum sized aperture of raw data for processing.

Figure 7-5 illustrates the relationships of functions performed by the fieldable system. As indicated, scanning motion and the process of transmitting ultrasonic energy and receiving it and conditioning the data is accomplished by the scanner system. The scanning sequence is generated, stored, and communicated from the host computer to the scanner system. This does not preclude stand-alone control of the scanning sequence by a stored program within the scan controller.

Raw data is configured as formatted files by the host with communication to the SAFT processor managed by cooperation of the host and the SAFT processor. Tables of shift control words are generated withing the host by appropriate algorithms for a given geometry as an off-line operation and stored for communication to the processor by the host. The SAFT data processor performs the shift and addprocessing of each data point of all A-scans within the aperture. Fully processed data points are scaled back to an 8 bit byte size by the SAFT processor and communicated to the host for envelope detection and stored in a formatted processed and detected data file.

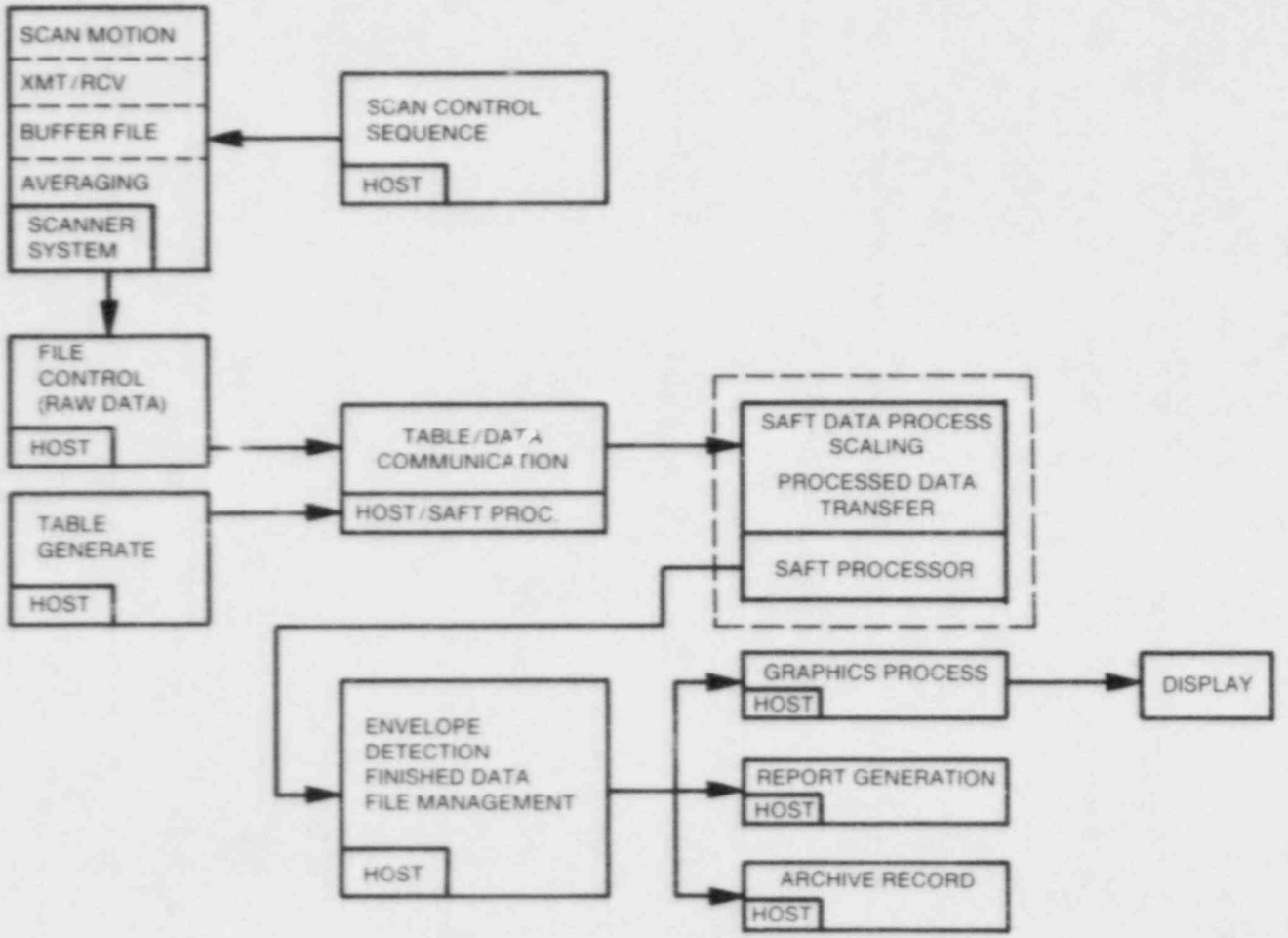


FIGURE 7-5 SAFT Function Relations

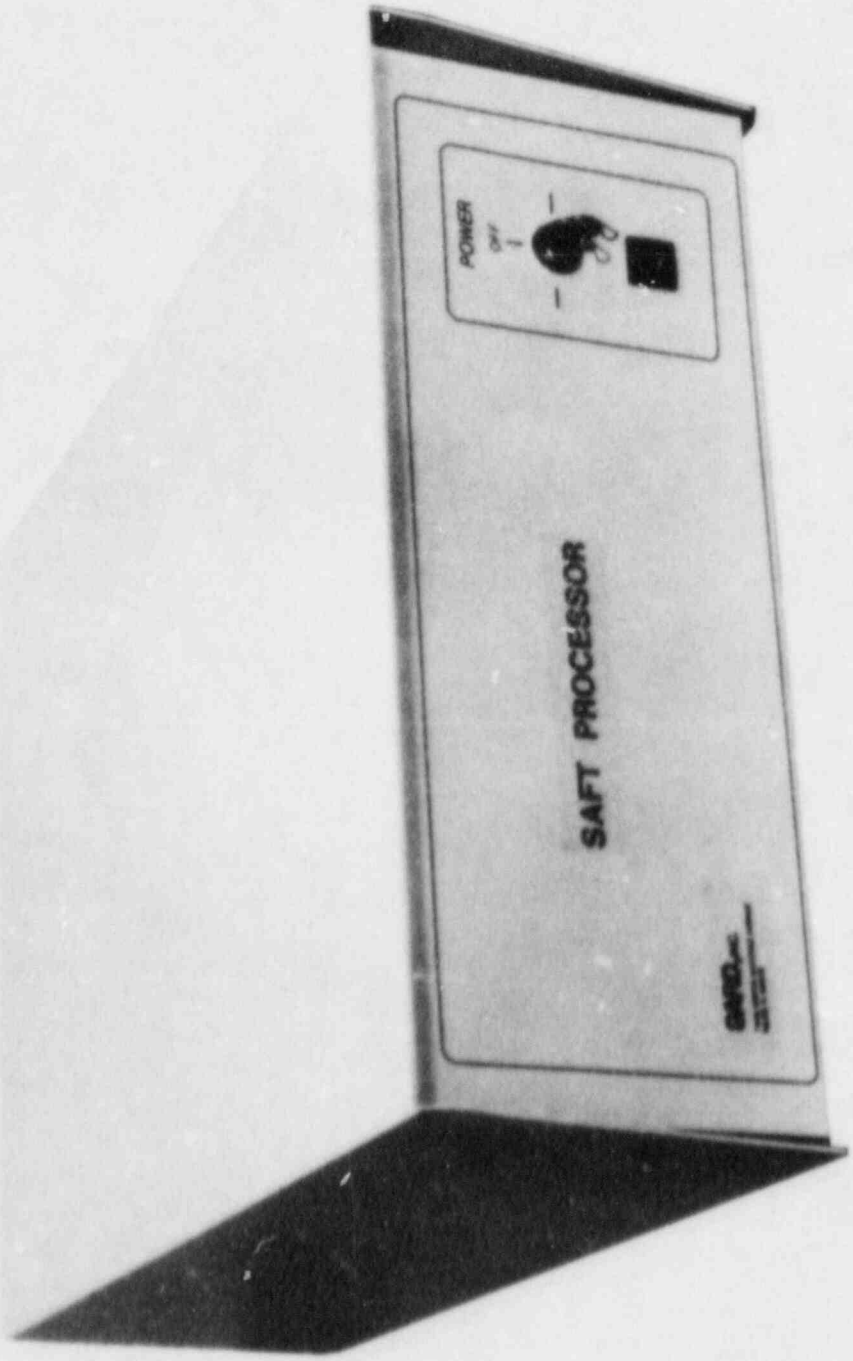


FIGURE 7-6 SAFT Processor Prototype

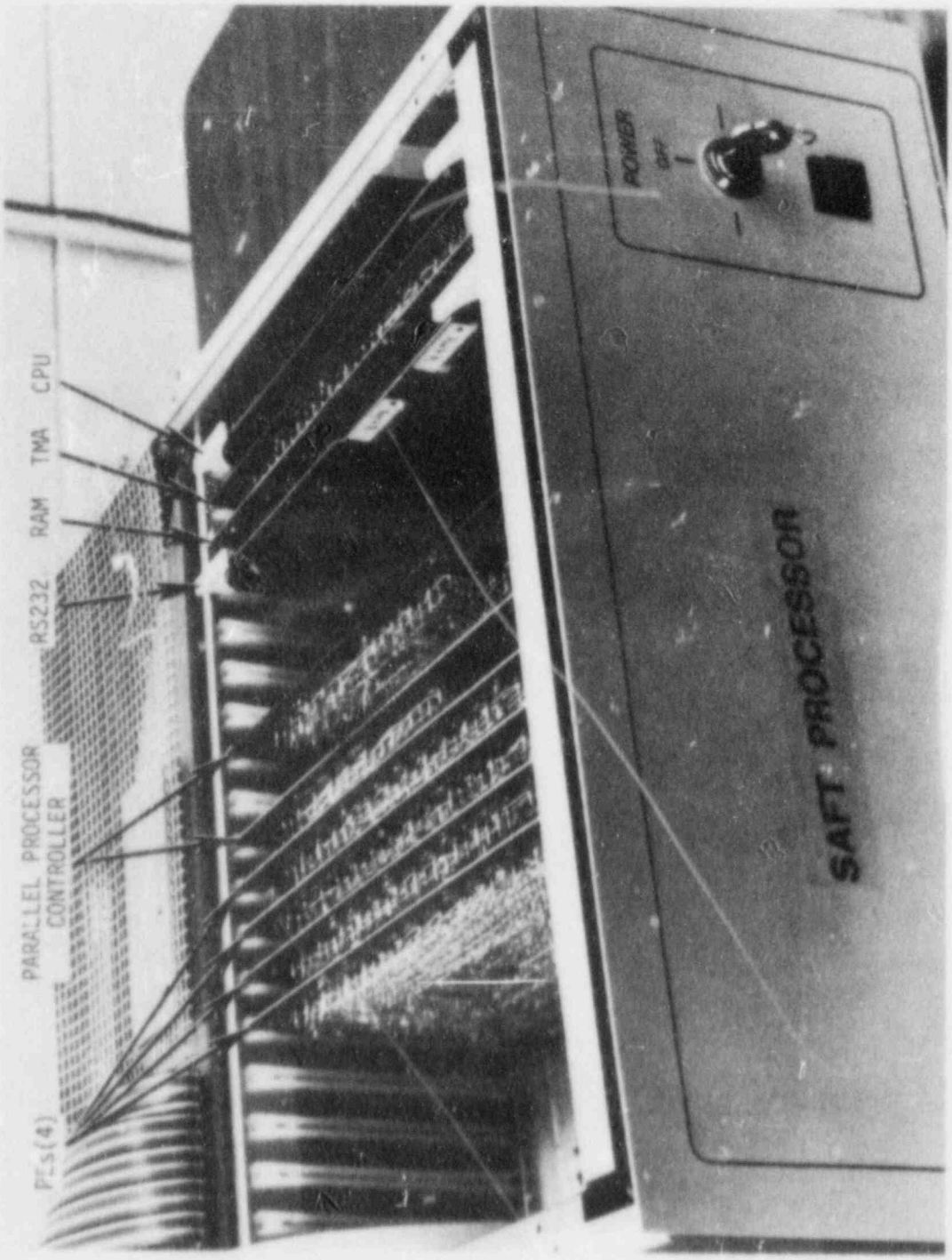


FIGURE 7-7 SAFT Processor Board Complement

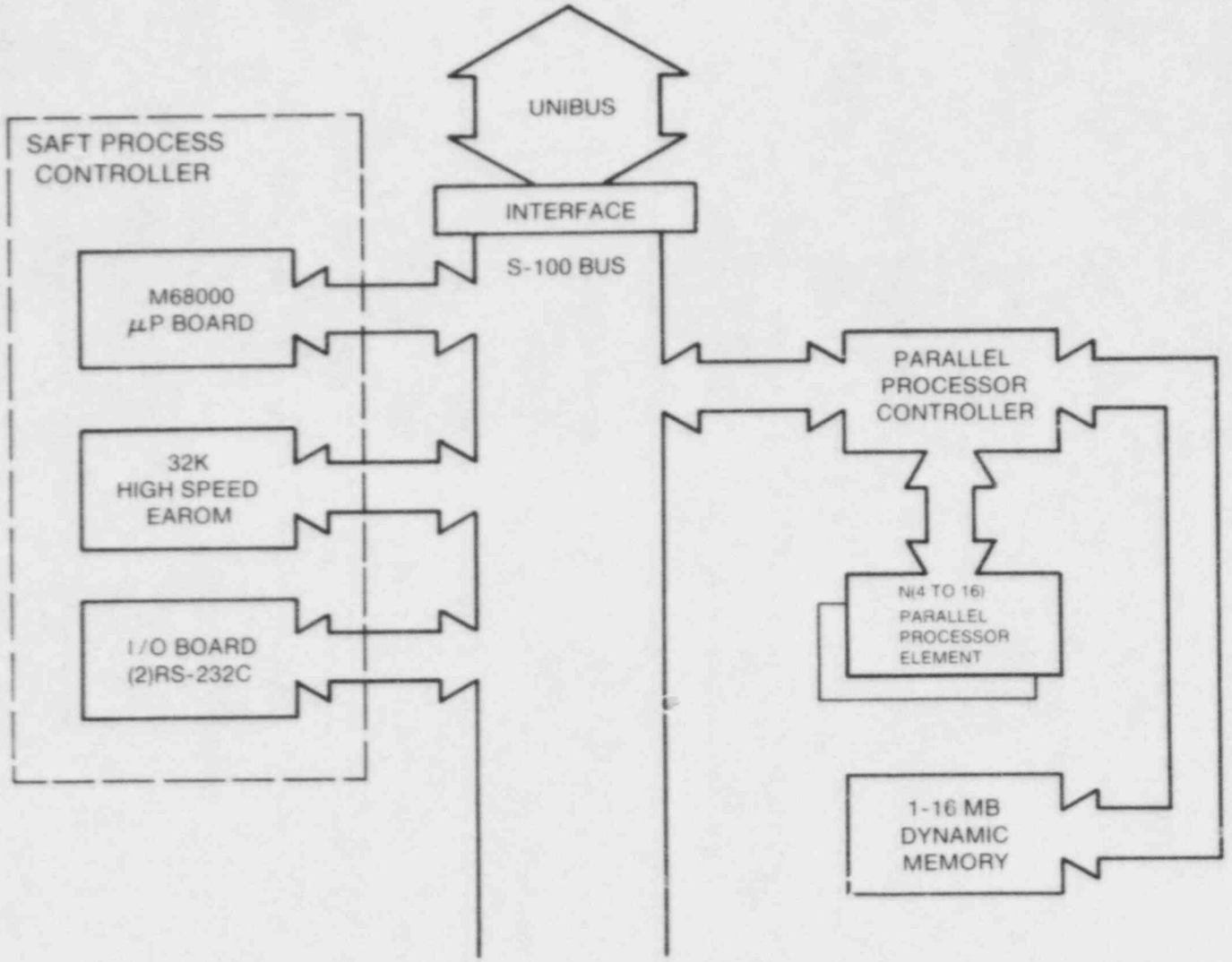


FIGURE 7-8 SAFT Processor Block Diagram

7.2. REAL-TIME SAFT PROCESSOR IMPLEMENTATION

The prototype SAFT processor developed on this program is shown in Figure 7-6. All system elements except the 1 megabyte aperture memory are housed in an 18-slot S-100 bus enclosure with integral power supply. The S-100 bus has been split between the two halves of the motherboard to form two independent buses; an S-100 bus for computer elements and a processing element (PE) bus. Figure 7-7 shows the board complement of the prototype.

A block diagram of the subsystem components of the SAFT processor is shown in Figure 7-8. Communication with the host computer (DEC VAX 11/730) utilized a Unibus mounted Direct Memory Access (DMA) module in the host and a GARD-developed Temporary Master Access (TMA) board on the SAFT processor's S-100 bus. In this manner, expedient data transfer was possible as neither processor was involved in the data exchange as is the case with time-consuming programmed I/O methods. Both the host and the SAFT processor memory were accessed allowing direct transfer of data from or to the host. On completion of a transfer, both buses are returned to their respective computer systems. Also residing on the S-100 bus is the M68000 CPU board, a 32K byte high speed EAROM, an RS-232C I/) board with 2 asynchronous serial ports, and the SAFT processing element controller/interface. This equipment group will handle communication between the parallel elements and the host CPU and also perform additional tasks such as scaling processed data. Each of the board functions will be individually described in the sections that follow.

7.2.2. Parallel Processor Controller

The task of timing and control of data and shift, control works to the parallel processing elements is accommodated by the parallel processor controller shown pictorially in Figure 7-9, and in the block diagram in Figure 7-10. This assembly occupies an address space on the S-100 bus thereby making it the communications link to the processor's CPU. It also originates a processing element (PE) bus allowing the CPU to directly address the PE's to retrieve processed data.

The address space for the Parallel Process Controller includes 128 contiguous words used by the CPU to write shift-add codes, write raw data to the 16 M-byte RAM, and read processed data. Adjacent with this address space are four 16-bit registers defining scanning parameters.

The basic process is controlled by 68000 firmware and software, thereby allowing a maximum flexibility in system configuration requiring a minimum of hardware modification. To achieve this goal, while maintaining system speed, the parallel processor controller uses a pair of duplicate registers ($X-Y$ proc 0 and $X-Y$ proc 1) and shift/add memories (S/A MEM 0 and S/A MEM 1); one is in use by the process controller to define the $x-y$ coordinates of the off-center A-scan being summed and the corresponding set of shift add codes to be used, the other is available to the CPU for loading those parameters for the next process cycle. In this way, an uninterrupted process can be achieved. To expedite loading, the 2-bit shift add codes are concatenated, 8 to each 16-bit word. A decommutator (mus) is used to break these down into 2-bit words for broadcast to the first processing element.

As new raw data is made available from the scanner (via the host computer), the CPU requests a process stop to allow access the raw data memory. At this time, the

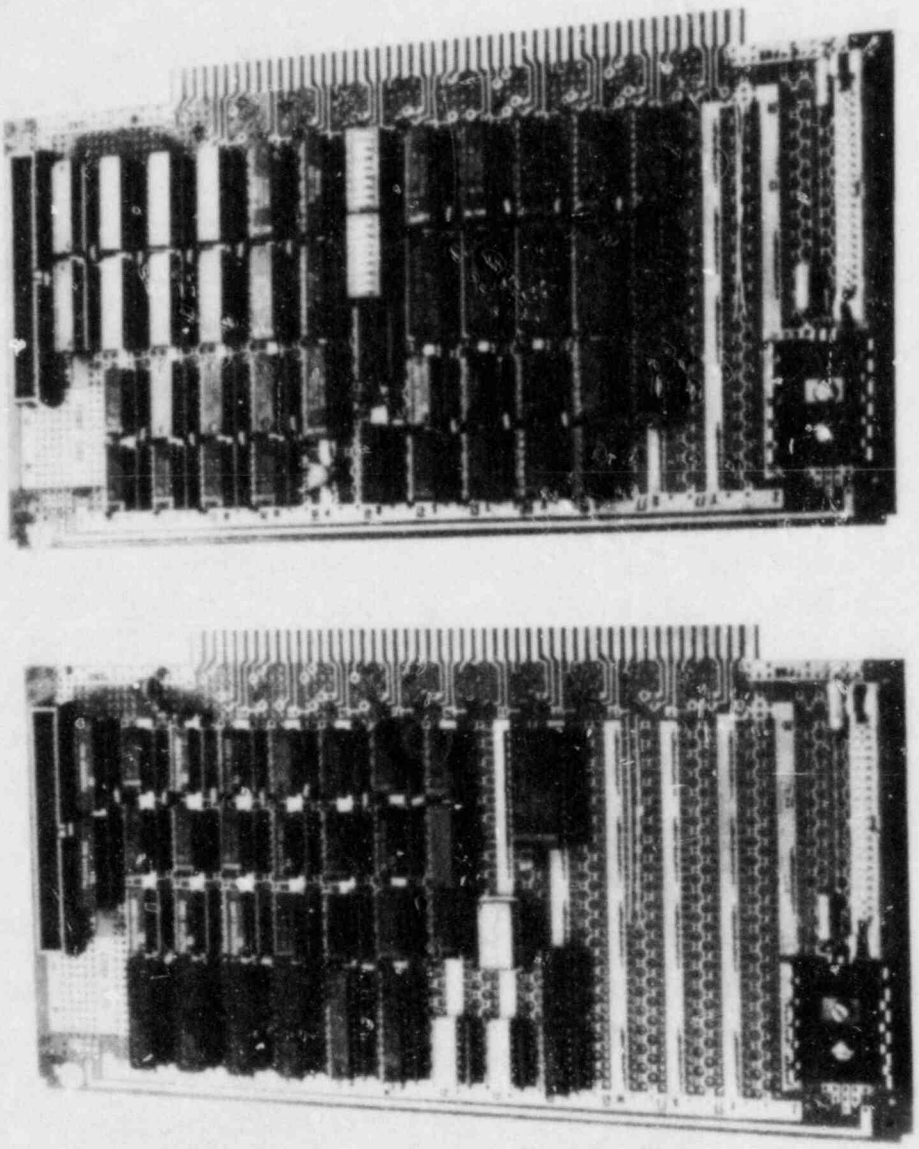


FIGURE 7-9 Parallel Processor Controller

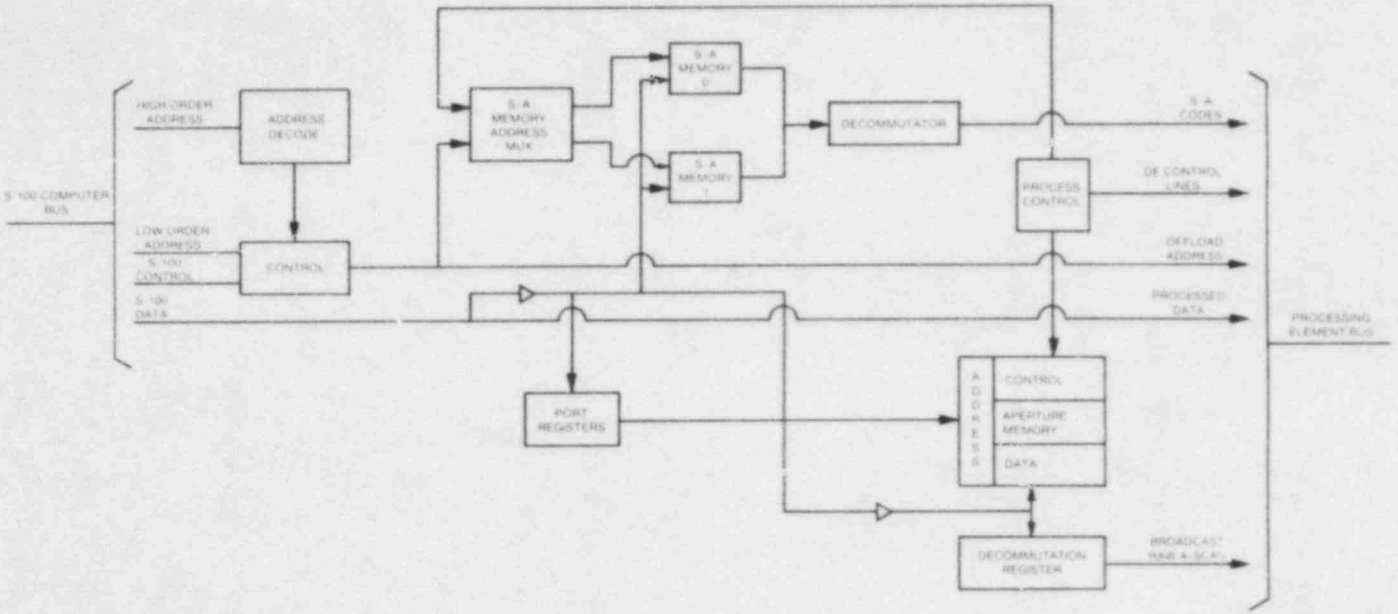


FIGURE 7-10 Parallel Process Controller Block Diagram

CPU sets the load $x-y$ register to define the partial address of the aperture memory to be loaded. An entry A-scan (up to 1024 bytes) will be loaded; this will be in the form of up to 512 16-bit words. When access is granted, the load address is applied to the aperture memory address port; the S-100 data lines are tied to the aperture memory's data lines. Thus, the raw data memory is tied in to CPU control for loading. On completion of the load, the CPU releases the process controller for continuation of SAFT processing.

The remaining function of the parallel process controller is off loading of processed SAFT data. The correlation accumulations are stored in the processing elements (PE). The details of the PE hardware will be discussed later. The process controller signals the CPU when new accumulations are available for off loading. As cited earlier, a word length of up to 20 bits will be required to contain the accumulation of 3000 8-bit words; hence each processed A-scan accumulation will be represented by a long word (32-bits). During the CPU read of the accumulated A-scans, the $x-z$ off load register contains a 4-bit code defining which of 16 PE's are to be off loaded as well as a 4-bit code providing the most significant four bits of the Z address. The buffered S-100 address bits A7-A1 form the remaining off load address. The S-100 data bus lines are extended to the PE bus allowing them to send data directly to the CPU.

During the normal processing cycle, data is read from the aperture memory. Each word contains two concatenated bytes or A-scan points. Two 16-bit registers are alternately loaded from the memory output lines. The memory read functions are performed one access time ahead of the broadcast time allowing for variance in data availability time due to the dynamic RAM refresh operations. Memory accesses occur every 1 μsecond on the average. The decommutation registers output the 8-bit A-scan points at uniform 500 nanosecond intervals compensating for access time variance. As a result, a

1024 point A-scan can be broadcast and accumulated in 512 μseconds.

The CPU - Parallel Process Controller operation is synchronized by a communications network consisting of a status register on the Parallel Process Controller and interrupts directed to the CPU. The status register informs the process controller of the request to load the aperture memory and of the availability of a new set of shift add codes. There are three interrupts to the CPU. The highest priority is a request to load a new set of shift-add codes; this occurs when the process controller begins a new process cycle using the previously loaded code set. The second is in response to a CPU request to load raw data and indicates that processing has been halted to allow this load to occur. The third tells the CPU that a new set of processed accumulations is ready to be off loaded. The software carefully uses these communications to keep the process under control at maximum speed. Of course, failure of the host to provide new raw data or to receive scaled processed data will temporarily halt the process. This will not affect the data integrity and processing will resume once the cause of data stoppage is rectified.

7.2.2. Processing Element

The processing element (PE) is shown in Figure 7-11; a block diagram showing the functional components of a typical PE is in Figure 7-12. This is one of 4 such elements in the prototype operating in pipeline parallel mode to achieve the increased processing rates. Each off-center A-scan is shifted and added to the on-center A-scan, represented by the particular processing element, through a sequence of up to 1024 control words sent serially to the processing element and delayed by a fixed length "FIFO" to the next processing element. The FIFO is in the form of a 1024$\times$2-bit RAM addressed by the Z-scan register. This is the Z component of the raw A-scan element being broadcast. As

cited in the previous section, A-scan points and their corresponding 2-bit shift-add codes are broadcast at a 2 mHz rate (500 nsecond period). The raw A-scan points are broadcast to all PE's, while the S/A words are sent only to the first PE. The second PE operates on the set of S/A commands used by the first PE during the previous scan of the Z values. The FIFO accomplishes the required delay by storing S/A codes transmitted to it by the parallel process controller during a given process scan. During the next scan, it will make the previous S/A word available to the next PE before writing the new S/A word from the previous PE. Since the same adjacent geometry transfers sequentially down the line, each group of delayed control words is serially delayed by an identical length "FIFO" and serves subsequent PE's down the line.

Similarly, the broadcast off center A-scan data passes through a buffer FIFO. The FIFO is implemented using an 8-bit by 64 word RAM. The 1024 byte A-scan is written in sequential addresses of this RAM overwriting up to 16 time. The read pointer of this RAM is controlled by the shift add words and allows multiple uses of a given raw data point. It will never be allowed to lag the loading pointer by more than 30 to 40 counts, hence the overwriting cited above will be eradicating previously used and/or unneeded data.

A center A-scan pointer (CAS) provides the Z address of the center A-scan element currently in process. This pointer is incremented under control of the shift add control word currently in use. During the process of a given point on the center A-scan, the current value of the point determined by the CAS is read from a $1K \times 20$-bit accumulation RAM into a 20-bit register. Conditional with an Add command in the Shift add control word, the output of this register is summed with the output of the broadcast data FIFO word (corresponding to the read pointer). The new partial sum is then writ-

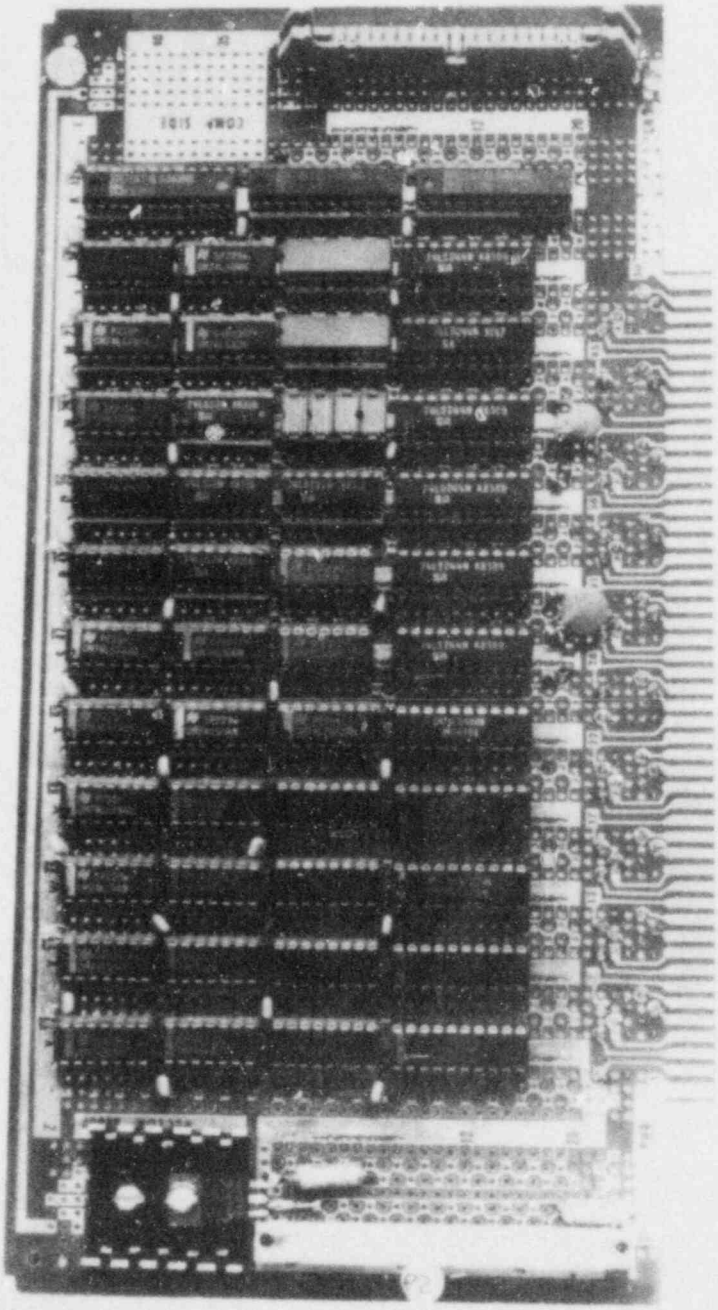


FIGURE 7-11 Processing Element

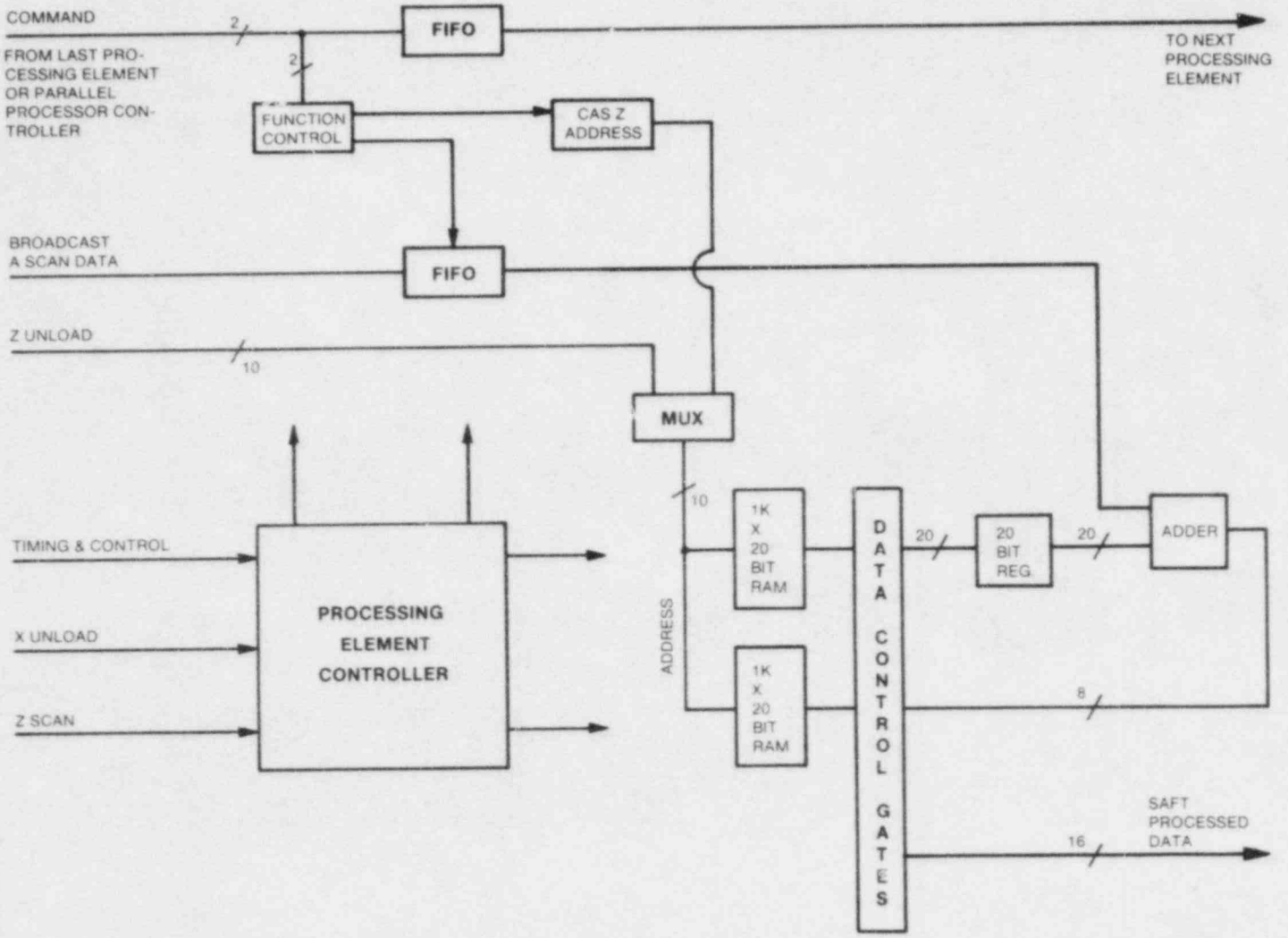


FIGURE 7-12 Processing Element, Block Diagram

ten back into the same RAM location.

Two such high-speed $1K \times 20$-bit RAMs are used in a toggle mode with one RAM operating as cited above with the other containing a fully processed A-scan ready to be off-loaded, scaled by the CPU, and transmitted back to the host computer. As indicated, up to a 20-bit sum is possible with the anticipated aperture size. The gating of the RAM data formats each 20-bit word as one 16-bit word followed by the 4 most significant bits in the following word. During this read mode, the selected output word is placed directly on the S-100 data lines for access by the CPU.

7.3. CONCLUSIONS

The prototype processor contains four processing elements along with all support elements discussed previously. Data points are broadcast from the parallel process controller at a 2 megasample rate. Thus, a 1024 point off-center A-scan is correlated with all on-center A-scan ideally in 512 μseconds. With process restart overhead, this time interval has been observed to be 540 μseconds.

The basic interval is used to determine the thruput rate possible for systems containing various number of processing elements. Table 7-1 summarizes the pertinent calculations. N is the number of PEs in the system which for implementation simplicity is restricted to powers of 2. The N_{oc} is the number of off-center A-scans in the aperture of the N continuous colinear on-center A-scans. The Rep's column shows the number of times the N_{PE}'s must be used to process the 192 on-center scans forming a plane of output data. The total column is the product of the previous two and is proportional to the total amount computation per unit volume required. The thruput rate is inversely proportional to Total and process speed and is specified in cubic inches per minute. As

TABLE 7-1 PROCESSOR PERFORMANCE vs N_{PE}

N_{PE}	N_{OC}	REPETITIONS	TOTAL	THRUPUT (IN^3/MIN)	THRUPUT/PE
1	3019	192	579,648	4.41	4.41
2	3082	96	295,872	8.63	4.32
4	3208	48	153,984	16.6	4.15
8	3460	24	83,040	30.7	3.84
16	3964	12	47,568	53.7	3.35
32	4972	6	29,832	85.7	2.68
64	6988	3	20,964	122.	1.91

shown in the table the four-PE prototype system developed on this program provides a thruput rate of up to 16.6 cubic inches per minute. It demonstrates the expected increase over a single PE system, which itself, operates nearly an order of magnitude faster than most conventional computer systems performing the same task. The last column in Table 7-1 gives the average thruput rate per PE. As can be seen, when N exceed 16, the effectiveness of all PEs is reduced significantly. Overall system speed does increase, but only with a large increase in hardware accompanied by increased enclosure volume and power supply demands, power consumption and cooling requirements, and in spite of the high reliability of solid state electronics, a corresponding decrease in MTBF. It is for these reasons that an optimal value of N=16 be chosen. With 16 PEs, the SAFT processor can operate with an input volume rate of 71.6 cubic inches/minute resulting in 53.7 cubic inches per minute of processed data.

The present prototype design will support a 16-PE system although present enclosure restrictions prohibit expansion past 8 PE's with power supply capacity being even more restrictive. Similarly, the aperture volume has been reduced for the prototype system in accordance with the recommendations in the U of M Report No. UIL-TR-3. This was done chiefly to reduce the aperture memory requirements from 16 M byte to a more realistic prototype size of 1 M byte. This also sizes the SAFT processor prototype consistent with the data file sizes currently available for the prototype evaluation.

By completion of the program, SAFT processor operation was realized and design performance, in terms of data thruput rate, was achieved. This was accomplished using software emulation of all host computer functions by the SAFT processor's own 68000 CPU. Thus the SAFT prototype fulfills the requirements of this program.

Checkout of the SAFT prototype, first on simulated and then on actual data, remains to be accomplished. While data interchange between the host and the SAFT processor has been achieve, complete checkout of host software needs to be performed before such an evaluation can be started.

RECOMMENDATIONS:

Based on the completed evaluation of the SAFT processor prototype in terms of thruput rate, and its anticipated successful performance evaluation on ideal A-scan data, it is recommended that the prototype be used in an extended evaluation on various types of SAFT processing. These evaluations will be within the scaled down aperture limits of the prototype and will demonstrate thruput rates consistent with a 4-element processor.

GARD anticipates a successful evaluation which will be followed by a review of prototype performance by both PNL and GARD personnel prior to full scale SAFT processor design and fabrication. At this point the current prototype design will be value engineered and incorporate changes deemed necessary. Possible changes include: incorporating separate input and output ports allowing the processor to be "in" the data stream and not "on" it, allowance for control of system configuration via host system commands, and determination of optimal aperture memory configuration. System packaging will be addressed to take into account SAFT processor integration with other SAFT imaging system components (i.e., host computer, scanner, display, etc.).

8. Conclusions

It is our conclusion, based on our more than three years of research, that a real-time, field-implementable SAFT system is very viable and holds great promise for real-time quantitative NDE of reactor vessels and piping components. This, coupled with the enhanced display techniques for inspection, makes an inherently volumetric approach taken by SAFT even more attractive. It is our feeling that our initial results with edge detectors for automatic sizing of flaws shows great promise and deserves more experimental testing. We also believe that it would be worthwhile to develop better flaw detection techniques so that a computation intensive process like SAFT can be used with even greater success in areas that show the existence of the potential flaw. While it is possible to do real-time SAFT inspection on a complete vessel, it might be more profitable to develop a less expensive system to work on smaller areas as detected by a good front-end flaw detection system.

REFERENCES

1. L. S. Davis, "A Survey of Edge Detection Techniques," *Computer Graphics and Image Processing*, Vol.4, 1975, pp.248-270.

2. J. R. Fram and E. S. Deutsch, "On the Quantitative Evaluation of Edge Detection Schemes and Their Comparison with Human Performance," *IEEE Transactions on Computers*, Vol.C-24, No.6, June 1975, pp.616-628.

3. I. E. Abdou and W. K. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Processings of the IEEE*, Vol.67, No.5, May 1979, pp.753-763.

4. G. B. Shaw, "Local and Regional Edge Detectors: Some Comparisons," *Computer Graphics and Image Processing*, Vol.9, 1979, pp.135-149.

5. R. A. Hummel, "Feature Detection Using Basis Functions," *Computer Graphics and Image Processing*, Vol.9, 1975, pp.40-55.

6. W. Frei & C. C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," *IEEE Transactions on Computers*, Vol.C-26, No.10, Oct. 1977, pp.988-998.

7. M. H. Hueckel, "An Operator Which Locates Edges in Digitized Pictures," *Journal of Association for Computing Machinery*, Vol.18, No.1, Jan. 1971, pp.113-125.

8. M. H. Hueckel, "A Local Visual Operator Which Recognizes Edges and Lines," *Journal of Association for Computing Machinery*, Vol.20, No.4, Oct. 1973, pp.634-647.

9. M. H. Hueckel, Erratum to "A Local Visual Operator Which Recognizes Edges and Lines," *Journal of Association for Computing Machinery*, Vol.21, No.2, 1974, pp.350.

10. L. Me'rö and Z. Vassy, "A Simplified and Fast Version of the Hueckel Operator for Finding Optimal Edges in Pictures," *Proceedings of the 4th International Conference on Artificial Intelligence*, Tbilisi, USSR, Sept. 1975, pp.650-655.

11. Frederick, J.R., et.al, "Improved Ultrasonic Nondestructive Testing of Pressure Vessels - Annual Progress Report, Aug. 1, 1975 - July 31, 1976", NRC Report No. NUREG-0007-2.

12. *Proceedings of NDE Experts Workshop on Austenitic Pipe Inspection* (G.J. Dau, ed.), EPRI Special Report SR-30, Feb. 1976.

13. Shankar, R., et.al, "Development of Adaptive Learning Networks for Pipe Inspection", EPRI NP-688, Proj. 770 Final Report, March 1978.

14. Fortunko, C.M., "Ultrasonic Detection and Sizing of Two-Dimensional Weld Defects in the Long Wavelength Limit", Proc. 1980 IEEE Ultrasonics Symposium, pp.862-867.

15. Altes, R.A., "Sonar for Generalized Target Description and its Similarity to Animal Echolocation Systems," JASA, vol.59, pp.97-105, Jan. 1976.

16. Flax, L., G.C. Gaunaurd, and H. Uberall, "Theory of Resonance Scattering", in *Physical Acoustics, Vol.XV* (W.P. Mason and R.N. Thurston, eds.), Academic Press, New York, 1981, pp.191-294.

17. Norton, S.J., and M. Linzer, "Ultrasonic Reflectivity Imaging in Three Dimensions: Exact Inverse Scattering Solutions for Plane, Cylindrical, and Spherical Apertures", IEEE Trans. Biomed. Engin., vol.BME-28, pp.202-220, Feb. 1981.

18. Szilard, J., and G. Scruton, "Revealing the Grain Structure of Metals by Ultrasonics", Ultrasonics, pp.114-120, May 1973.

19. Kelly, E.J., I.S. Reed, and W.L. Root, "The Detection of Radar Echoes in Noise, I and II", J. Soc. Indust. Appl. Math., vol.8, pp.309-341, June 1960, and vol.8, pp.481-505, Sept. 1960.

20. Root, W.L., "Stability in Signal Detection Problems", Proc. Symp. Appl. Math., vol.16, pp.247-263, 1964.

21. Selin, I., *Detection Theory*, Princeton University Press, Princeton, NJ, 1965.

22. Proegler, H., and R. Keck, "Defect Distinction by Evaluating the Phase of Ultrasonic Pulses", Report of Betriebsforschungs institut, Fed. Rep. Germany.

23. Mucciardi, A.N., and G.J. Dau, "Detection and Sizing of Feedwater Nozzle Cracks", IPRI Report RP1125-1(c), Jan. 1979.

24. Varadan, V.K., "Multiple Scattering of Acoustic, Electromagnetic, and" Elastic Waves", in *Acoustic, Electromagnetic, and Elastic Wave Scattering - Focus on the T-matrix Approach* (V.K. Varadan and V.V. Varadan, eds.), Pergamon Press, New York, 1980, pp.103-134.

25. Krautkrämer, J. and H. Krautkrämer, *Ultrasonic Testing of Materials* (2nd edition), Springer-Verlag, Berlin, 1977.

Appendix A: Grain Noise

A.1. Noise in small grained materials

Reference [17] shows that the backscattered signal resulting from spherical wave insonification of a scattering region R with compressibility $K_0 + K(\vec{r})$ and density $\rho_0 + \rho(\vec{r})$, where K_0 and ρ_0 are the mean values and when $E\,|K(\vec{r})|^2 \ll K_0^2$ and $E\,|\rho(\vec{r})|^2 \ll \rho_0^2$, can be written as

$$\tilde{\psi}(\vec{r}_0,k) = k^2\tilde{\rho}(k)\int_R f(\vec{r})g_k^2(\vec{r}\mid\vec{r}_0)d\vec{r} \tag{A-1}$$

where $\tilde{\psi}(\vec{r}_0,k)$ = the Fourier transform of the backscattered signal; $k^2\tilde{\rho}(k)$ = the Fourier transform of the source output; $f(\vec{r}) = \frac{K(\vec{r})}{K_0} - \frac{\rho(\vec{r})}{\rho_0}$ and $g_k(\vec{r}\mid\vec{r}_0) = \frac{1}{4\pi|\vec{r}-\vec{r}_0|}\,e^{jk|\vec{r}-\vec{r}_0|}$. With the change of notation

$$f = -\frac{ck}{2\pi} \qquad Y(f) = \tilde{\psi}(\vec{r}_0,k)$$
$$r = |\vec{r}-\vec{r}_0| \qquad X(f) = k^2\tilde{\rho}(k)$$
$$b^2 = 16\pi^2 \ .$$

Equation (A-1) becomes

$$Y(f) = X(f)\int_{R_0}^{R_1} A(r)e^{-\frac{j4\pi fr}{c}}\,dr \tag{A-2}$$

where $A(r) = \frac{2}{b^2r^2}\int_{-\pi}^{\pi}\int_{0}^{\frac{\alpha}{2}} f(r,\theta,\phi)\sin\theta\,d\theta\,d\phi$ for α = beamwidth and $[R_0,R_1]$ the range of radii which corresponds to the scattering region. For small beamwidths and relatively large values of r, the wave is approximately planar; in that case,

$$A(r) = \frac{2}{b^2r^2}\int_{a(r)}\int f(r,z,y)\,dz\,dy \tag{A-3}$$

where, for our case, r is now a variable measuring vertical depth and $a(r)$ is the inter-

section of the cone of insonification with the plane $z = r$. Equation (A-2) does not include the effects of frequency- and depth-dependent attenuation and dispersion; they can be included by replacing the term $e^{-\frac{j4\pi f r}{c}}$ by $e^{-\gamma(f,r)}$, where $\gamma(f,r)$ is the same as that discussed in Section 6.2.1. Then

$$Y(f) = X(f) \int_{R_0}^{R_1} A(r) e^{-\gamma(f,r)} dr \quad .$$

Let $n_g(t) = \int_{-\infty}^{\infty} Y(f) e^{j2\pi f t} df$; then the grain noise is given by

$$n_g(t) = \int_{R_0}^{R_1} A(r) p(t,r) dr \tag{A-4}$$

where $p(t,r) = \int_{-\infty}^{\infty} X(f) e^{-\gamma(f,r)} e^{j2\pi f t} df$ and $A(r)$ is the random process specified by equation (A-3).

A.2. Wide-sense stationarity of grain noise

From equation (A-3) $A(r) = \frac{2}{b^2 r^2} \int_{a(r)} \int f(r,x,y) dx dy$. The $\frac{1}{r^2}$ term comes from the beam-spread as the wave propagates through the scattering region. It can be interpreted as implying that the insonification energy impinging on a given area decreases with rate r^2 as the depth of the area increases. Note, however, that the area of $a(r)$, the intersection of the cone of insonification with the plan $z = r$, grows with a rate r^2. Consider the autocorrelation of $A(r)$:

$$\begin{aligned} E\, A(r)A(\rho) &\propto \frac{1}{r^2\rho^2} \int_{a(r)} \int \int_{a(\rho)} \int E[f(r,x,y) f(\rho,u,v)] dx dy du dv \\ &= \frac{1}{r^2\rho^2} \int_{a(r)} \int \int_{a(\rho)} \int R_f(r,\rho;x,u;y,v) dx dy du dv \quad . \end{aligned} \tag{A-5}$$

Assume that R_f is separable in the vertical and lateral directions, and that the grains

are uniformly scattered laterally, so that equation (A-5) may be written as

$$\frac{R_0(r,\rho)}{r^2\rho^2} \int_{a(r)} \int \int_{a(\rho)} \int R_1(z-u;y-v)\,dz\,dy\,du\,dv \quad .$$

At one extreme, if the grains are considered to be independent point scatterers, this becomes

$$\frac{\sigma^2(r)}{r^4}\,\delta(r-\rho) \int_{a(r)} \int \int_{a(r)} \int \delta(|z-u|)\,\delta(|y-v|)\,dz\,dy\,du\,dv$$

$$\frac{\sigma^2(r)}{r^2}\,\delta(r-\rho)$$

so that there is a $\frac{1}{r^2}$ dependence (which is seen, for example, in underwater reverberation noise). At the other extreme, if $f(r,z,y)$ is constant in z and y, then the area terms cancel the $\frac{1}{r^2\rho^2}$ term, and

$$E\ A(r)A(\rho) \propto R_0(r,\rho) \quad .$$

If $R_0(r,\rho) = R_0(r-\rho)$, this implies that $A(r)$ is wide-sense stationary, which is in fact nearer to the observed results in ultrasonic testing [18]. Even if the grain properties do vary with depth, they do not change rapidly, so at worst, the process $A(r)$ will be assumed to be locally wide-sense stationary, with an autocorrelation function

$$E\ A(r)A(\rho) = g_0(r+\rho)g_1(r-\rho) \quad .$$

(For example, in the case of independent scatters, $g_0(r+\rho) = \frac{4}{(r+\rho)^2}\sigma^2(\frac{r+\rho}{2})$ and $g_1(r-\rho) = \delta(r-\rho)$). We assume that $g_1(r-\rho) = 0$ for $|r-\rho| > r_0$, where r_0 is a constant which depends on the grain sizes. Then, the autocorrelation function for $n_g(t)$ is, from equation (A-4)

$$R_n(t,s) = E\ n_g(t)n_g(s)$$

$$= E \int_{R_0}^{R_1}\int_{R_0}^{R_1} A(r)A(\rho)p(t,r)p(s,\rho)d\rho dr$$

$$= \int_{R_0}^{R_1}\int_{R_0}^{R_1} g_0(r+\rho)g_1(r-\rho)p(t,r)p(s,\rho)d\rho dr$$

$$= \sum_{k=0}^{k-1} \int_{R_0+kR}^{R_0+(k+1)R}\int_{r-r_0}^{r+r_0} g_0(r+\rho)g_1(r-\rho)p(t,r)p(s,\rho)d\rho dr$$

where $R_1 = R_0 + KR$. Assume that R can be chosen so that for $r \in [R_0+kR, R_0+(k+1)R]$, $\rho \in [r-r_0, r+r_0]$:

$$p(t,r) \approx p_k(t-\epsilon r)$$

$$p(s,\rho) \approx p_k(s-\epsilon\rho)$$

$$g_0(r+\rho) \approx \text{a constant } g_k \quad .$$

That is, R is a depth interval small enough so that $A(r)$ is nearly wide-sense stationary over the interval and so that the shape of the impulse response of the forward and back metal path filter does not appreciably change. Then

$$R_n(t,s) \approx \sum_k g_k \int_{R_0+kR}^{R_0+(k+1)R}\int_{r-r_0}^{r+r_0} g_1(r-\rho)p_k(t-\epsilon r)p_k(s-\epsilon\rho)d\rho dr \quad .$$

Let $t_1 = \epsilon r$, $t_2 = \epsilon\rho$, $t_0 = \epsilon r_0$, $T_0 = \epsilon R$, $T = \epsilon R$, and $g_1(z) = \frac{1}{\epsilon^2} g(\frac{z}{\epsilon})$. Then

$$R_n(t,s) = \sum_k g_k \int_{T_0+kT}^{T_0+(k+1)T}\int_{t_1+t_0}^{t_1-t_0} g(t_1-t_2)p_k(t-t_1)p_k(s-t_2)dt_2dt_1 \quad .$$

Let $u = t-t_1$, $v = s-t_2$, and $w = t-s$; then we have

$$R_n(t,s) = \sum_k g_k \int_{t-T_0-(k+1)T}^{t-t_0-kT}\int_{(u-w)-t_0}^{(u-w)+t_0} g(v-(u-w))p_k(u)p_k(v)dvdu \quad .$$

Assume that $p_k(u) = 0$ for u not in $[-T_k^1, T_k^1]$ and that $T > 2T_k$; define

$$I_k = [T_0+kT+T_k^1, T_0+(k+1)T-T_k^1] \quad .$$

Then, for $t \in I_k$,

$$R_n(t,s) = g_k \int_{-T_k^1}^{T_k^1} p_k(u) \int_{(u-w)-t_0}^{(u-w)+t_0} g(v-u-w)p_k(v)dvdu \quad .$$

The value of the inner integral is a function only of $(u-w)$ (t_0 is a constant); call it $\widetilde{g}_k(u-w)$. Then

$$R_n(t,s) = g_k \int_{-T_k^1}^{T_k^1} p_k(u)\widetilde{g}_k(u-w)du$$

$$= h_k(w) = h_k(t-s) \quad . \tag{A-6}$$

So, under the essential assumption that the propagating signal is short compared to the range over which its shape does not change by very much, the noise autocorrelation can be represented by a series of wide-sense stationary functions for different depth intervals.

Appendix B: Detection Algorithms

B.1. Known Signal (case I)

The A-scan, $z(t)$, is equal either to $n(t)$, which is known Gaussian noise with autocorrelation function $R(t,r)$, or to $n(t)+s(t;r,\alpha)$, where

$$s(t;r,\alpha) = \int_{-\infty}^{\infty} h_s(t-r;\alpha)p(r;r)\,dr$$

and $p(r;r)$ is known. Assume that (see Appendix A):

(a) The shape of $p(r;r)$ is approximately the same for all r in a range M_i; the entire sample depth is covered by a series of regions $\{M_i\}$, $i = 1, \ldots, M$. The rough estimate of target range is then equal to the estimate of which region the reflector is in; i.e., to an estimate of i.

(b) Over a time interval equivalent to the depth interval M_i, the autocorrelation function of $n(t)$ is nearly wide-sense stationary.

(c) There are N_i samples of the A-scan corresponding to depths in M_i; call the column vector of A-scan samples from that region z_i, and let $p_i(r)$ be the column vector of the sampled signal for that region delayed by r (that is, $p_i(tau)$ is the vector of samples of $p(t-r;r_i)$ where r_i is the depth at which M_i begins).

(I) Suppose that $h_s(t;\alpha) = f_0\delta(t-t_0)$ $(\alpha = \{f_0,t_0\})$; that is, suppose the object can be represented as a point reflector. Then, from equation 14 in Section (6.2.4):

$$L_i(Z_i) = \frac{\sup_{f_0,t_0,i} \frac{1}{(\sqrt{2\pi})^{N_i} |R_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(z_i - f_0 p_i(t_0))^T R_i^{-1}(z_i - f_0 p_i(t_0))\}}{\frac{1}{(\sqrt{2\pi})^{N_i} |R_i|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(z_i^T R_i^{-1} z_i\}}$$

where R_i is the autocorrelation matrix for the noise in the region M_i, $|R_i|$ is its deter-

minant, and the superscript T denotes transpose. Detection consists of comparing $L_i(Z_i)$ to a threshold. It is well known that decisions based on monotone functions of the likelihood ratio are equivalent to those based on the likelihood ratio itself, so, by taking logarithms, the problem can be stated as one of finding

$$\sup_{f_0 t_0 i} f_0 z_i^T R_i^{-1} p_i(t_0) - \frac{1}{2} f_0^2 \; p_i^T(t_0) R_i^{-1} p_i(t_0) \; . \tag{B-1}$$

Let $q_i(t_0) = R_i^{-1} p_i(t_0)$; note that because R_i is wide-sense stationary, $p_i^T(t_0) q_i(t_0)$ is a constant (say, c_i) for t_0 in the interval corresponding to M_i minus one signal period. Then we must find

$$\sup_{f_0 t_0 i} f_0 z_i^T q_i(t_0) - \frac{1}{2} f_0^2 \; c_i \; . \tag{B-2}$$

Suppose the maximum likelihood estimates of t_0 and i are known; then $\hat{f}_0$, the maximum likelihood estimate of f_0, solves the equation

$$z_i^T q_i(t_0) = \hat{f}_0 c_i$$

or

$$\hat{f}_0 = \frac{1}{c_i} \; z_i^T q_i(t_0) \; . \tag{B-3}$$

Substitute this back into (B-2); we must find

$$\sup_{t_0 i} \hat{f}_0 z_i^T q_i(t_0) - \frac{1}{2} \hat{f}_0^2 \; c_i$$

$$= \sup_{t_0 i} \frac{1}{c_i} \; [z_i^T q_i(t_0)]^2 - \frac{1}{2c_i} \; [z_i^T q_i(t_0)]^2$$

$$= \sup_{t_0 i} \frac{1}{2c_i} \; [z_i^T q_i(t_0)]^2 \; .$$

The detection and estimation algorithm is thus given by:

(a) For each i, precalculate $q_i(0)$;

(b) For each $m_i \in \{0, 1, \ldots, N_i - 1\}$, $i = 1, \ldots, M$, correlate z_i with $q_i(m_i)$; call the resulting values $e_i(m_i)$;

(c) For each m_i, calculate $\frac{1}{c_i}\, e_i^2(m_i)$;

(d) Compare the peak value of $\frac{1}{c_i}\, e_i^2(m_i)$ over all m_i to a predetermined threshold.

Detection is accomplished if the peak value exceeds the threshold; the estimate of i is given by the region in which the peak occurs, the estimate of t_0 is the time coordinate of the peak (m_i), and the estimate of f_0 is given by equation (B-3).

(II) Generalize the model of (I) to $h_\alpha(t;\alpha) = \sum_{k=1}^{K} f_k\, \delta(t - t_k)$; that is, suppose the target can be represented by a set of point reflectors. Then, in analogy to (B-1), we must find

$$\sup_{\{f_k\},\{t_k\},i} z_i^T R_i^{-1}\Big(\sum_{k=1}^{K} f_k\, p_i(t_k)\Big) - \frac{1}{2}\Big[\sum_{k=1}^{K} f_k\, p_i(t_k)\Big]^T\; R_i^{-1}\Big[\sum_{k=1}^{K} f_k\, p_i(t_k)\Big] \quad .$$

Let $q_i(t_k) = R_i^{-1} p_i(t_k)$, again, $p_i^T(t_k) q_i(t_k)$ is a constant for all values of t_k, and we will call it c^i. So, the quantity to be found is

$$\sup_{\{f_k\},\{t_k\},i} \sum_k f_k\; z_i^T q_i(t_k)\Big) - \frac{1}{2}\sum_k [f_k\, c^i + \sum_{j\neq k} f_j\, p_i^T(t_k) q_i(t_j)] \quad . \tag{B-4}$$

Suppose the estimates of i and the $\{t_k\}$ are given, let $c_{kj}^i = p_i^T(t_k) q_i(t_j)$ (note that $c_{kj}^i = c_{jk}^i$), and let $a_k^i = z_i^T q_i(t_k)$. Then $\hat{f}_k$ solves the equation

$$a_k^i - \sum_{j\neq k} \hat{f}_j\; c_{kj}^i - \hat{f}_k\; c^i = 0 \quad .$$

So:

$$c^i \hat{f}_k + \sum_{j\neq k} \hat{f}_j\; c_{kj}^i = a_k^i \qquad k = 1, \ldots, K \quad . \tag{B-5}$$

Let A_i be the vector $[a_1^i, a_2^i, \ldots, a_k^i]^T$, $\hat{f}$ the vector $[\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_k]^T$, and C_i the $K \times K$ symmetric matrix with main diagonal $[c^i, \ldots, c^i]$ and j^{th} sub- or super-diagonal $[c_{1,j}^i, c_{2,j+1}^i, \ldots, c_{k-j+1,K}^i]$. Then equation (B-5) can be restated as the matrix equation $\hat{f} = C_i^{-1} A_i$. As in part (I), substitute back into (B-4), we get

$$\sup_{\{t_k\},i} \hat{f}^T A_i - \frac{1}{2} \hat{f}^T C_i \hat{f}$$

$$= \sup_{\{t_k\},i} A_i^T C_i^{-1} A_i - \frac{1}{2} A_i^T C_i^{-1} C_i C_i^{-1} A_i \qquad \text{(B-6)}$$

$$= \sup_{\{t_k\},i} \frac{1}{2} A_i^T C_i^{-1} A_i \quad .$$

If K is large, this seems to be impractical (unless the separations of the t_k's from one another are known). It corresponds to correlating z_i with q_i, and for all combinations of K points from the output calculating the quadratic (B-6), which involves the inverse of the matrix of signal correlations at the K points chosen. The estimate of i is the region in which (B-6) occurs, and the estimates of the $\{t_k\}$ are those values which give the peak. When $K = 2$, a reasonable algorithm exists; (B-6) reduces to

$$\sup_{t_1,t_2,i} (c^{i^2} - c_{12}^{i^2})^{-1} [(a_1^{i^2} + a_2^{i^2}) c^i - 2 a_1^i a_2^i c_{12}^i] \quad .$$

Let $t_2 - t_1 = \tau$, denote c_{12}^i as $c^i(\tau)$, and substitute back in the values of a^i; then we must find

$$\sup_{t_1,\tau,i} (c^{i^2} - c^{i^2}(\tau))^{-1} \{[z_i^T q_i(t_1)]^2 c^i + [z_i^T q_i(t_1+\tau)]^2 c^i - 2 z_i^T q_i(t_1) z_i^T q_i(t_1+\tau) c^i(\tau)] \quad .$$

Let $\alpha_i(\tau) = \dfrac{c^i}{(c^{i^2} - c^{i^2}(\tau))}$ and $\beta_i(\tau) = -2 \dfrac{c^i(\tau)}{c^i}$.

Then, for a given i and τ, the detector is implemented as shown in Figure B-1. The detection and estimation algorithm is as follows:

(a) Precompute $q_i(0)$ and the functions $\alpha_i(\tau)$ and $\beta_i(\tau)$ for $i = 1, \ldots, M$;

(b) For each value of i, find the peak value over τ of the output $p_0(\tau, i)$ of the receiver shown in Figure B-1; call this value $p_1(i)$;

(c) Find the largest value of $p_1(i)$; call it p_2.

Detection is accomplished if p_2 exceeds a threshold; $\hat{i}$ is the value of i for which $p_1(i) = p_2$, $\hat{\tau}$ is the value of τ for which $p_0(\tau, \hat{i}) = p_2$, and $\hat{\tau}_1$ is the value of τ_1 at which

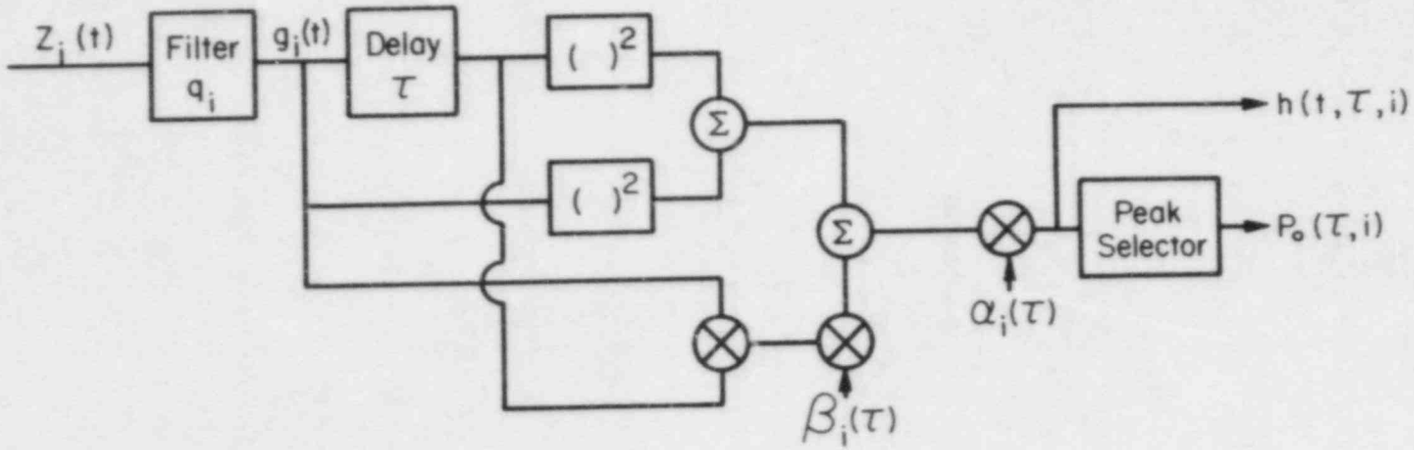


FIGURE B-1 Receiver for Two-Point Reflection Model

$h(t_1,\vec{r},\iota) = p_2$.

Note that in both (I) and (II), the main calculations which cannot be performed ahead of time are the correlations; for each value of ι, they require roughly N_ι^2 multiplications. Thus, the algorithms of (I) and (II) each require on the order of $\sum_{\iota=1}^{M} N_\iota^2$ operations.

B.2. Noise-in-Noise (case II)

From equation (13) in Section 6.2.1., $s(t;\alpha) = \int_{-\infty}^{\infty} \int_E h_{ss}(t-\tau;\vec{r})p(\tau;\vec{r})d\vec{r}d\tau$. In this case, $p(t;\vec{r})$ is a random field and has an autocorrelation function $\tilde{U}(t,\tau;\vec{r},\vec{\rho}) = Ep(t;\vec{r})p(\tau;\vec{\rho})$, which implies that

$$Q(t_1, t_2;\alpha) \triangleq E\ s(t_1;\alpha)s(t_2;\alpha)$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_E\int_E \tilde{U}(\tau,\mu;\vec{r},\vec{\rho})h_{ss}(t_1-\tau;\vec{r})h_{ss}(t_2-\mu;\vec{\rho})d\vec{\rho}d\vec{r}d\mu d\tau\ .$$

Assume that $\tilde{U}(t,\tau;\vec{r},\vec{\rho}) = 0$ for $\left|\frac{\vec{r}}{|\vec{r}|} - \frac{\vec{\rho}}{|\vec{\rho}|}\right| > \epsilon$ (i.e., that if the angle between the vectors $\vec{r}$ and $\vec{\rho}$ is larger than some fixed size, the processes at $\vec{r}$ and $\vec{\rho}$ are independent, and that for $r = |\vec{r}|$, $\tilde{U}(t,\tau;\vec{r},\vec{r})$ can be written as $U(t,\tau;r)$ (i.e., that the value of the autocorrelation function of the process at a point depends only on the depth of the point). This function is assumed to be known.

Consider a single point reflector at a position $\vec{r}_0$ with time delay v_0, so that $h_{ss}(t;\vec{r}) = f_0\delta(t-v_0)\delta(|\vec{r}-\vec{r}_0|)$ $(\alpha = \{f_0, v_0, \vec{r}_0\})$; then

$$Q(t_1,t_2;\alpha) = f_0^2 \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_E \delta(t_1-\tau-\upsilon_0)\delta(t_2-\mu-\upsilon_0)\delta(|\vec{r}-\vec{r}_0|)$$

$$\int_E \tilde{U}(\tau,\mu;\vec{r},\vec{\rho})\delta(|\vec{\rho}-\vec{r}_0|)d\vec{\rho}\,d\vec{r}\,d\mu\,d\tau$$

$$= f_0^2 U(t_1-\upsilon_0, t_2-\upsilon_0; |\vec{r}_0|) \ .$$

Assume that the same conditions on the noise autocorrelation mentioned in section B.1 hold, and let $U_i(\upsilon_0)$ denote the $N_i \times N_i$ autocorrelation matrix for the signal in the i^{th} region, where $[U_i(\upsilon_0)]_{m,n} = U(t_m-\upsilon_o, t_n-\upsilon_0;r_i)$. Then the generalized likelihood ratio is

$$L_i(Z_i) = \frac{\sup\limits_{f_0\upsilon_0 i} \dfrac{1}{(\sqrt{2\pi})^{N_i}\,|R_i+f_0^2U_i(\upsilon_0)|^{\frac{1}{2}}}\ \exp\{-\frac{1}{2}(z_i^T[R_i+f_0^2U_i(\upsilon_0)]^{-1}z_i\}}{\dfrac{1}{(\sqrt{2\pi})^{N_i}\,|R_i|^{\frac{1}{2}}}\ \exp\{-\frac{1}{2}(z_i^TR_i^{-1}z_i\}} \ .$$

Its logarithm is

$$\sup_{f_0\upsilon_0 i} \log \frac{|R_i|^{\frac{1}{2}}}{|R_i+f_0^2U_i(\upsilon_0)|^{\frac{1}{2}}} + \frac{1}{2}\ z_i^T\{R_i^{-1}-[R_i+f_0^2U_i(\upsilon_0)]^{-1}\}z_i \ .$$

So, the quantity we must find is

$$\sup_{f_0\upsilon_0 i} \frac{1}{2}\ Z_i^T\{R_i^{-1}-[R_i+f_0^2U_i(\upsilon_0)]^{-1}\}Z_i - \log|R_i+f_0^2U_i(\upsilon_0)|^{\frac{1}{2}} + \log|R_i|^{\frac{1}{2}} \quad \text{(B-7)}$$

Getting an explicit solution for this is not easy. We can simplify it by using a "large signal" approximation, however; that is, over a set of $K \times K$ points centered on (υ_0,υ_0) in M_i, where $K < N_i$, assume that $f_0^2U(\upsilon_0) + R_i \approx f_0^2U(\upsilon_0)$, where the matrices are now $K \times K$, and that the $K \times K$ matrix $U_i = U_i(\upsilon_0)$ is the same for all υ_0 in the interval being considered. Now consider the K point detection problem, and let $z_i(\upsilon_0) = [z_{i,\upsilon_0}, z_{i,\upsilon_0+1}, \ldots, z_{i,\upsilon_0+K-1}]$. Then the solution of (B-7) can be approximated

by the solution of

$$\sup_{f_0 v_0 i} z_i^T(v_0)R_i^{-1}z_i(v_0) - \frac{1}{f_0^2}\, z_i^T(v_0)U_i^{-1}z_i(v_0) - 2\log |f_0^2\, U_i|^{\frac{1}{2}} + 2\log |R_i|^{\frac{1}{2}} \tag{B-8}$$

since $(R_i + f_0^2 U_i)^{-1} \approx f_0^2 U_i^{-1}$ and $|R_i + f_0^2 U_i|^{\frac{1}{2}} \approx |f_0^2 U_i|^{\frac{1}{2}}$. Note that $|f_0^2 U_i| = f_0^{2K}\,|U_i|$ and assume that the estimates of v_0 and i are known. Then we are left with the problems of finding

$$\sup_{f_0} -\frac{1}{f_0^2}\, z_i^T(v_0)U_i^{-1}z_i(v_0) - 2K \log f_0 \,. \tag{B-9}$$

By setting the derivative of (B-9) with respect to f_0 equal to zero, the maximum likelihood estimate of f_0^2 is seen to be

$$\hat{f}_0^2 = \frac{z_i^T(v_0)U_i^{-1}z_i(v_0)}{K} \tag{B-10}$$

Substitute back into (B-8) to get

$$\sup_{v_0 i} z_i^T(v_0)R_i^{-1}z_i(v_0) - \frac{K}{z_i^T(v_0)U_i^{-1}z_i(v_0)}\, z_i^T(v_0)U_i^{-1}z_i(v_0)$$

$$- K \log \frac{z_i^T(v_0)U_i^{-1}z_i(v_0)}{K} - \log |U_i| + \log |R_i|$$

$$= \sup_{v_0 i} z_i^T(v_0)R_i^{-1}z_i(v_0) - K - K \log z_i^T(v_0)U_i^{-1}z_i(v_0) + K \log K - \log |U_i| + \log |R_i| \,.$$

So, the quantity which must be found is

$$\sup_{v_0 i} z_i^T(v_0)R_i^{-1}z_i(v_0) - K \log [z_i^T(v_0)U_i^{-1}z_i(v_0)] + \log \frac{|U_i|}{|R_i|} \,.$$

The detection procedure is thus accomplished by finding in each depth interval the value of v_0 at which the above expression is maximized, and comparing the overall peak to a threshold. The estimates of i and v_0 are the values giving the maximum, and $\hat{f}_0^2$ is given by equation (B-10). Assuming that there are approximately N_i possible values of v_0 in M_i, the total number of multiplications per A-scan is on the order of $2K^2 \sum_{i=1}^{M} N_i$.

Appendix C: System Functions

The detection and classification procedures discussed in this report assume that the following functions are known

(1) $g(t)$, the front surface response;

(2) the form of the reflector response, specified by a finite set of parameters with unknown values;

(3) $h_{FB}(t;r)$, the impulse response of the forward-and-back metal path filter to a depth r (attention is restricted here to case I problems; under case II what is assumed known is the autocorrelation function of $p(t;\vec{r}) = \int_{-\infty}^{\infty} h_{FB}(t-\tau;\vec{r})q(\tau)d\tau$); and

(4) $R(t,\tau)$, the autocorrelation function of $n(t)$.

The observation of $g(t)$ is usually easily accomplished, and the parameterization of the reflector response was discussed in Section B.1. Simple ways of estimating the functions $h_{FB}(t;r)$ and $R(t,\tau)$ are discussed in this appendix.

B.3.1. Metal Path Impulse Response ($h_{FB}(t;r)$)

The function $h_{FB}(t;r)$ depends on the material properties in the insonified region to a depth r, and at least theoretically it could be calculated from knowledge of those properties. Considering its transform, the quantity $\gamma(f,r)$ discussed in section B.1 is, for a given f and r, the attenuation and phase shift coefficient of a plane wave of frequency f at depth r; the calculation of this quantity is discussed in [14]. As might be expected, the calculation is difficult, and the solutions given in [14] depend on some stringent conditions imposed on the material characteristics. For our purposes, it

seemed to be more practical to experimentally determine the function. One way to do this is to assume that $\gamma(f,r) = \gamma(f) \times r$, and use the front and back surface reflections to find $\gamma(f)$. For the test set-up shown in Figure C-1, which involves an unfocused transducer at normal incidence, the Fourier-transformed back surface response can be written as

$$Z(f) = aG(f)e^{-\gamma(f)D} + N(f)$$

where $G(f)$ = the Fourier transform of $g(t)$; $N(f)$ = the Fourier transform of the received noise signal; $D = R_1 - R_0$, the distance between the front and back surfaces; and a is a factor which takes into account beam spread losses and reflection coefficients. Over the range of frequencies for which $|G(f)|$ is non-negligible (which is the only range we are interested in) the linear unbiased minimum variance estimate of $e^{-\gamma(f)D}$ is just $Z(f)/aG(f)$.

This procedure was used on a 52-mm thick stainless steel test sample. Using the formulas indicated in Figures C-1 and C-2 (which can be found in [15]), the beam spread factor with $h_0 = 10"$ was found to be 0.62 while the reduction in back surface response amplitude due to reflection and transmission coefficients was found to correspond to a factor of -0.126. Thus, for our sample, $a = -0.0781$. Using this value, $e^{-\gamma(f)D}$ was then derived from the back surface response. For any given depth r, the function $p(t,r)$, which represents the reflection from an ideal reflector at a depth r, is just $F^{-1}\{G(f)e^{-\gamma(f)(r-R_0)}\}$; graphs of this function for various depths using the results from our test sample are included in Figure C-3.

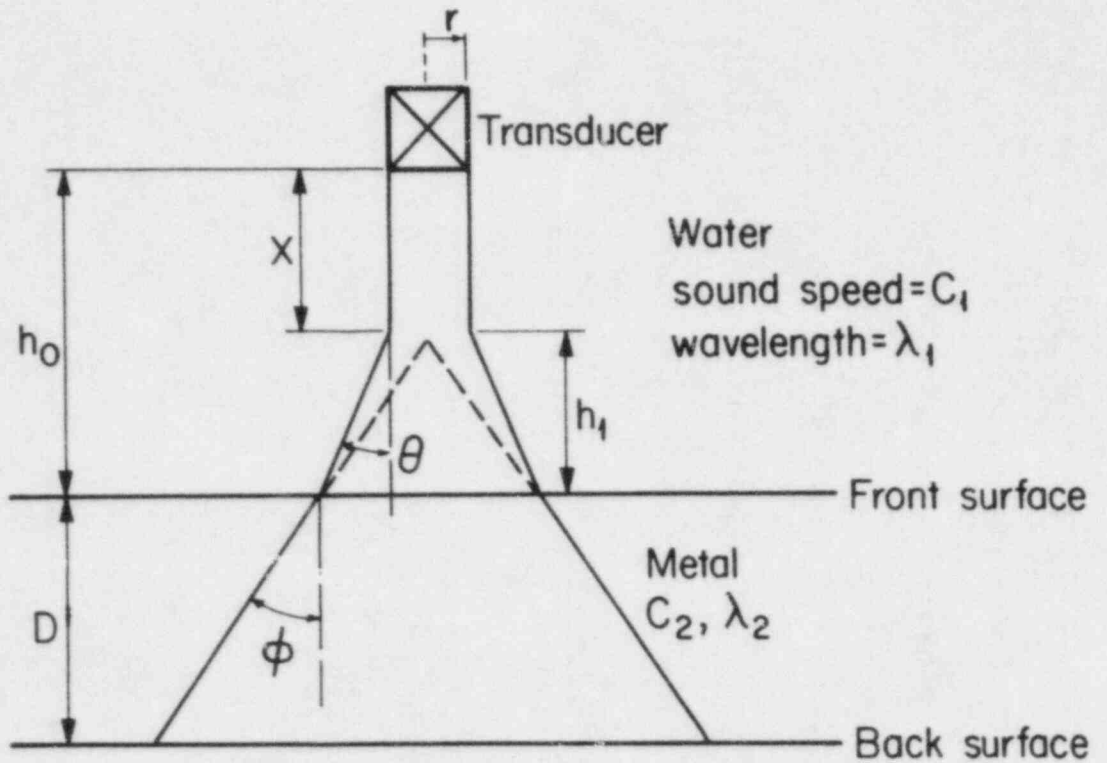
B.2.2. Grain Noise Autocorrelation (R(t,r))

As indicated in Appendix A, the grain noise in many materials can be assumed to be approximately wide=sense stationary on intervals over which the shape of the

propagating wave does not appreciably change. For our test sample, this range was estimated to be about 12 mm. A very simple autocorrelation function estimator was used on these intervals. Assume that the i^{th} interval M_i contains m_i points and that the autocorrelation fucntion is nearly equal to zero after k_i points ($m_i > 2k_i$). Pick P_i and N_i so that $M_i = [P_i - k_i - 1, P_i + N_i + k_i - 1]$ (i.e., so that the interval $[P_i, P_i + N_i - 1]$ lies in the middle of M_i). Then the estimate is formed by finding the individual section autocorrelations for each of several (say M) scans taken from non-flawed regions and averaging them. Thus, the estimated autocorrelation $\hat{R}_i$ is given by

$$\hat{R}_i(k) = \frac{1}{2N_i M} \sum_{l=1}^{M} \sum_{j=P_i}^{P_i+N_i-1} z_l(j)[z_l(j+k) + z_l(j-k)] \qquad k = 0, 1, \ldots, k_i \ .$$

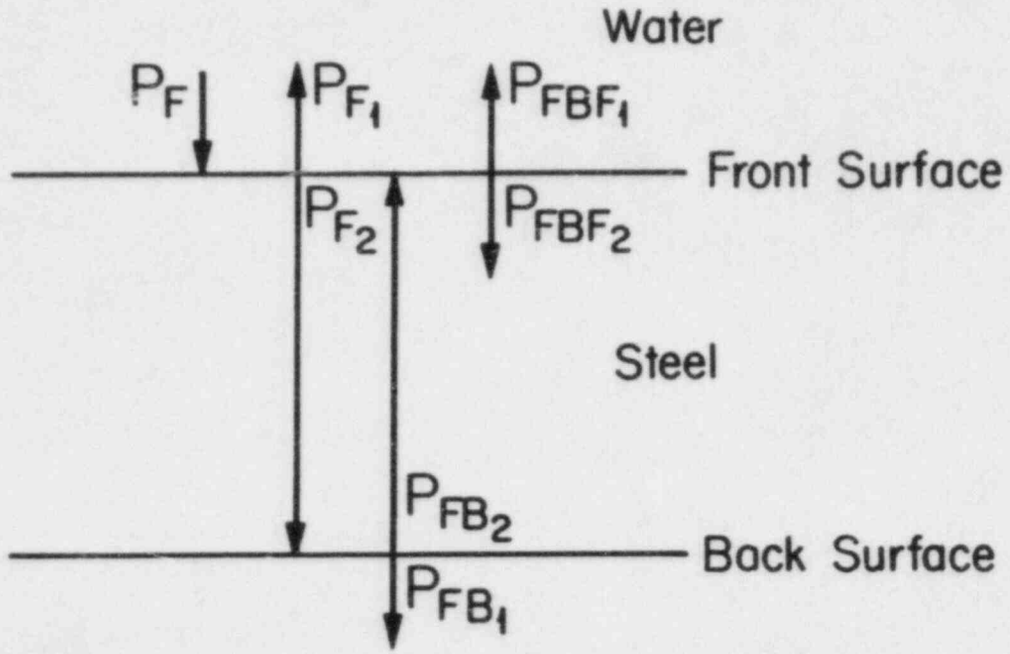
The expected value of $\hat{R}_i(k)$ is $R_i(k)$, and, assuming that the scans are far enough apart that the noise is independent from one scna to another, the variance of the estimate is proportional to $\frac{1}{M}$. Examples of estimates for autocorrelations from two different depths are given in Figure C-4.



$$\theta = \sin^{-1}\left(\frac{.61\lambda_1}{r}\right) \qquad X = \frac{4r^2 - \lambda_1^2}{4\lambda_1} \qquad \phi = \sin^{-1}\left(\frac{C_2}{C_1}\sin\theta\right)$$

$$h_1 = \left[r + \frac{h_0 - x}{\tan(90° - \theta)}\right]\tan(90° - \theta) \qquad \text{Beam-spread loss: } \frac{h_1}{D + h_1}$$

FIGURE C-1 Beam-Spread Loss Calculation



$P_{F_1}/P_F = .935$ $P_{F_2}/P_F = 1.935$ $P_{FB_2}/P_{F_2} = -.935$

$P_{FBF_1}/P_{FB_2} = .065$ Reflection Loss $= P_{FBF_1}/P_{F_1} = -.126$

FIGURE C-2 Reflection Loss Calculation

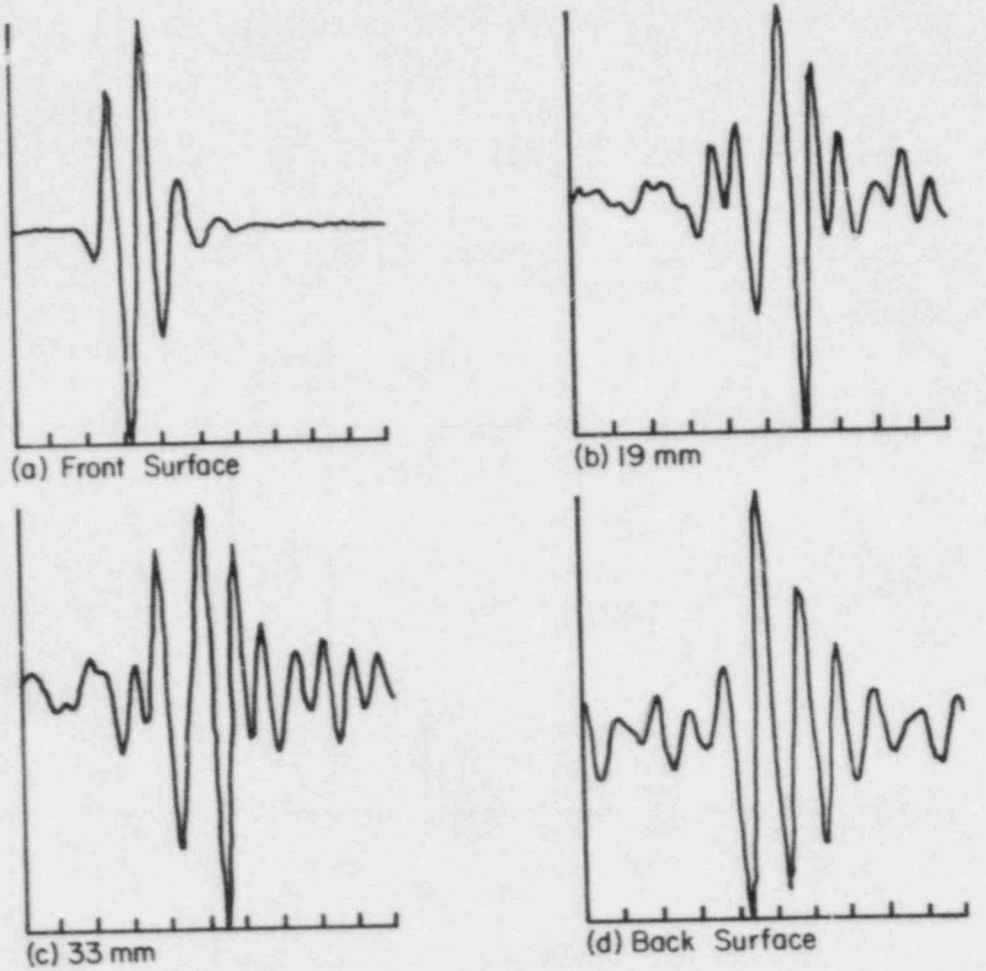


FIGURE C-3 Waveform Estimates

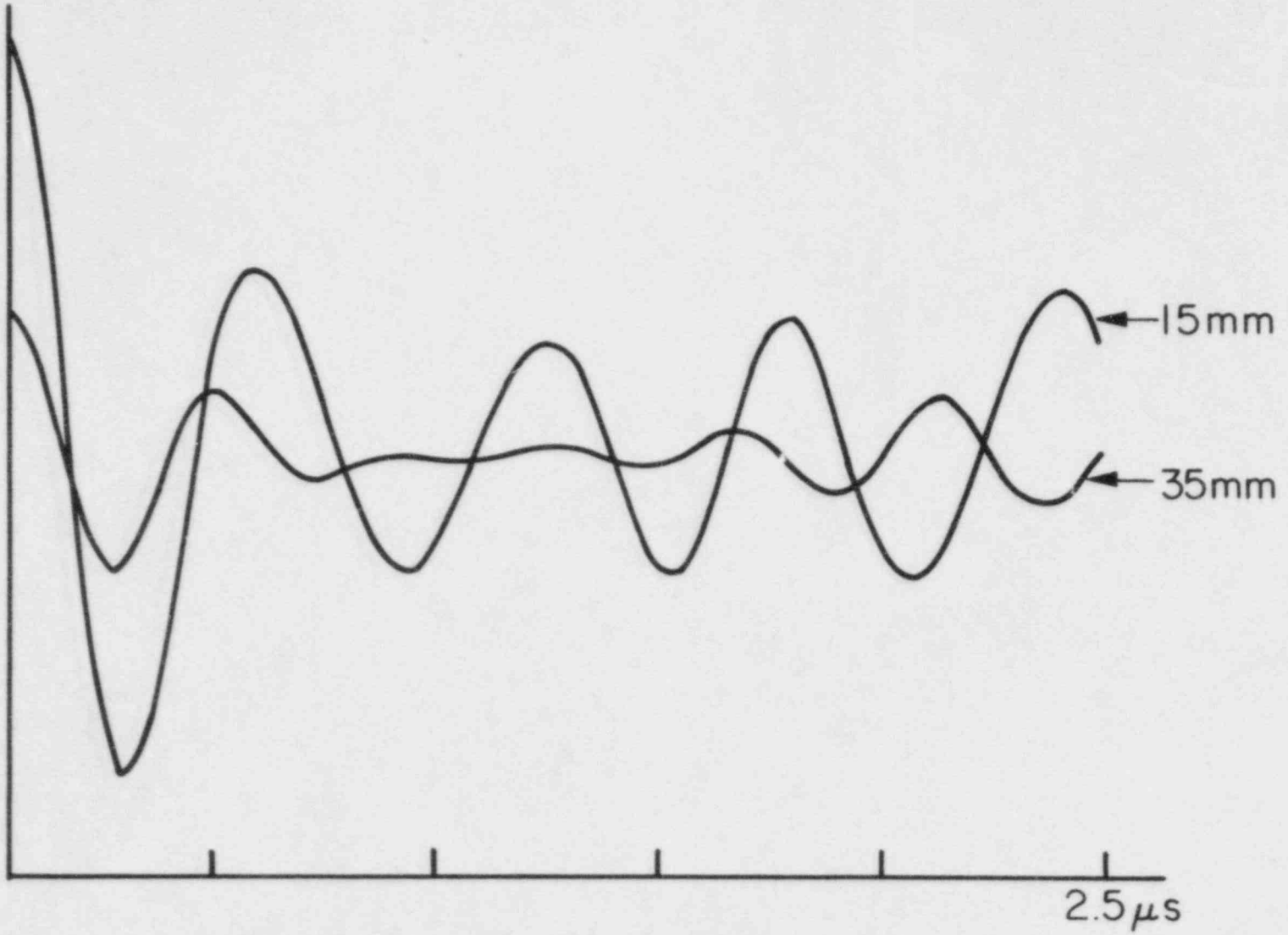


FIGURE C-4 Noise Autocorrelation Estimates

NRC FORM 336 (2-84) NRCM 1102, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

SEE INSTRUCTIONS ON THE REVERSE

1. REPORT NUMBER (Assigned by TIDC, add Vol. No., if any)
NUREG/CR-4365

2. TITLE AND SUBTITLE
Design and Development of a Special Purpose SAFT System for Nondestructive Evaluation of Nuclear Reactor Vessels and Piping Components

3. LEAVE BLANK

4. DATE REPORT COMPLETED
MONTH: July | YEAR: 1985

5. AUTHOR(S)
S. Ganapathy, B. Schmult, W. S. Wu,
T. G. Dennehy, N. Moayeri, and P. Kelly

6. DATE REPORT ISSUED
MONTH: August | YEAR: 1985

7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Department of Electrical and Computer Engineering
The University of Michigan
Ann Arbor, MI 48109-1092

8. PROJECT/TASK/WORK UNIT NUMBER

9. FIN OR GRANT NUMBER
A4047

10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)
Division of Engineering Technology
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

11a. TYPE OF REPORT
Final Report

b. PERIOD COVERED (Inclusive dates)
10/1/79 - 9/30/83

12. SUPPLEMENTARY NOTES

13. ABSTRACT (200 words or less)

This report describes the design details of a special purpose system for real-time nondestructive evaluation of reactor vessels and piping components. The system consists of several components and the report presents the results of the research aimed at the design of each component and recommendations based on the results. One major component of the NDE system, namely the real-time SAFT processor, was designed with sufficient details to enable the fabrications of a protype by GARD Inc. under a subcontract from The University of Michigan and the report includes their results and conclusions.

14. DOCUMENT ANALYSIS - a. KEYWORDS/DESCRIPTORS
Synthetic Aperture Focusing Techniques (SAFT)

b. IDENTIFIERS/OPEN-ENDED TERMS

15. AVAILABILITY STATEMENT
Unlimited

16. SECURITY CLASSIFICATION
(This page)
Unclassified
(This report)
Unclassified

17. NUMBER OF PAGES

18. PRICE

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

FOURTH CLASS MAIL
POSTAGE & FEES PAID
USNRC
WASH D C
PERMIT No G 67

NRC FORM 335 (2 84) NRCM 1102, 3201, 3202

SEE INSTRUCTIONS ON THE REVERSE

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

1. REPORT NUMBER (Assigned by TIDC, add Vol. No., if any)

NUREG/CR-4365

2. TITLE AND SUBTITLE

Design and Development of a Special Purpose SAFT System for Nondestructive Evaluation of Nuclear Reactor Vessels and Piping Components

3. LEAVE BLANK

4. DATE REPORT COMPLETED

MONTH	YEAR
July	1985

5. AUTHOR(S)

S. Ganapathy, B. Schmult, W. S. Wu, T. G. Dennehy, N. Moayeri, and P. Kelly

6. DATE REPORT ISSUED

MONTH	YEAR
August	1985

7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Department of Electrical and Computer Engineering
The University of Michigan
Ann Arbor, MI 48109-1092

8. PROJECT/TASK/WORK UNIT NUMBER

9. FIN OR GRANT NUMBER

A4047

10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Division of Engineering Technology
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555

11a. TYPE OF REPORT

Final Report

b. PERIOD COVERED (Inclusive dates)

10/1/79 - 9/30/83

12. SUPPLEMENTARY NOTES

13. ABSTRACT (200 words or less)

This report describes the design details of a special purpose system for real-time nondestructive evaluation of reactor vessels and piping components. The system consists of several components and the report presents the results of the research aimed at the design of each component and recommendations based on the results. One major component of the NDE system, namely the real-time SAFT processor, was designed with sufficient details to enable the fabrications of a protype by GARD Inc. under a subcontract from The University of Michigan and the report includes their results and conclusions.

14. DOCUMENT ANALYSIS - a. KEYWORDS/DESCRIPTORS

Synthetic Aperture Focusing Techniques (SAFT)

b. IDENTIFIERS/OPEN ENDED TERMS

15. AVAILABILITY STATEMENT

Unlimited

16. SECURITY CLASSIFICATION

(This page)

Unclassified

(This report)

Unclassified

17. NUMBER OF PAGES

18. PRICE