Automated Non-Destructive Testing Array Evaluation System

This report describes research sponsored by EPRI and the U.S. Department of Energy (Award No. DE-FC07-03ID14536).

WARNING:
Please read the Export Control Agreement on the back cover.

Technical Report

Automated Non-Destructive Testing Array Evaluation System

1009540

Final Report, December 2004

Cosponsor
U.S. Department of Energy

Project Manager
T. Wei

EPRI Project Manager
J. Benson

EPRI • 3412 Hillview Avenue, Palo Alto, California 94304 • PO Box 10412, Palo Alto, California 94303 • USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com

DISCLAIMER OF WARRANTIES AND LIMITATION OF LIABILITIES

THIS DOCUMENT WAS PREPARED BY THE ORGANIZATION(S) NAMED BELOW AS AN ACCOUNT OF WORK SPONSORED OR COSPONSORED BY THE ELECTRIC POWER RESEARCH INSTITUTE, INC. (EPRI). NEITHER EPRI, ANY MEMBER OF EPRI, ANY COSPONSOR, THE ORGANIZATION(S) BELOW, NOR ANY PERSON ACTING ON BEHALF OF ANY OF THEM:

(A) MAKES ANY WARRANTY OR REPRESENTATION WHATSOEVER, EXPRESS OR IMPLIED, (I) WITH RESPECT TO THE USE OF ANY INFORMATION, APPARATUS, METHOD, PROCESS, OR SIMILAR ITEM DISCLOSED IN THIS DOCUMENT, INCLUDING MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, OR (II) THAT SUCH USE DOES NOT INFRINGE ON OR INTERFERE WITH PRIVATELY OWNED RIGHTS, INCLUDING ANY PARTY'S INTELLECTUAL PROPERTY, OR (III) THAT THIS DOCUMENT IS SUITABLE TO ANY PARTICULAR USER'S CIRCUMSTANCE; OR

(B) ASSUMES RESPONSIBILITY FOR ANY DAMAGES OR OTHER LIABILITY WHATSOEVER (INCLUDING ANY CONSEQUENTIAL DAMAGES, EVEN IF EPRI OR ANY EPRI REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES) RESULTING FROM YOUR SELECTION OR USE OF THIS DOCUMENT OR ANY INFORMATION, APPARATUS, METHOD, PROCESS, OR SIMILAR ITEM DISCLOSED IN THIS DOCUMENT.

ORGANIZATION(S) THAT PREPARED THIS DOCUMENT

Argonne National Laboratory

ORDERING INFORMATION

Requests for copies of this report should be directed to EPRI Orders and Conferences, 1355 Willow Way, Suite 278, Concord, CA 94520, (800) 313-3774, press 2 or internally x5379, (925) 609-9169, (925) 609-1310 (fax).

Electric Power Research Institute and EPRI are registered service marks of the Electric Power Research Institute, Inc. EPRI. ELECTRIFY THE WORLD is a service mark of the Electric Power Research Institute, Inc.

Copyright © 2004 Electric Power Research Institute, Inc. All rights reserved.

CITATIONS

This report was prepared by

Argonne National Laboratory
Nuclear Engineering Division
9700 S. Cass Avenue
Argonne, IL 60439

Principal Investigators
T. Wei
N. Zavaljevski
S. Bakhtiari
A. Miron
D. Kupperman

This report describes research sponsored by EPRI, and the U.S. Department of Energy.
(Award No. DE-FC07-03ID14536. Task FY03-1 (1.1)).

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the Department of Energy.

The report is a corporate document that should be cited in the literature in the following manner:

Automated Non-Destructive Testing Array Evaluation System, EPRI, Palo Alto, CA, and U.S. Department of Energy, Washington, D.C.: 2004. 1009540.

REPORT SUMMARY

Utilities perform eddy current tests on nuclear power plant steam generator (SG) tubes to detect degradation. This report summarizes the status of ongoing research to develop signal processing algorithms that automate analysis of eddy current test data. The research focuses on analyzing array probe data for detecting, classifying, and characterizing degradation in SG tubes.

Background

Recent advances in eddy current probe development have resulted in an array probe design capable of high-speed data acquisition over the full length of SG tubes. Probe qualification programs have demonstrated that array probes are capable of providing similar degradation detection capabilities to the much slower rotating probe technology. However, to date, utilities have not used the array probe in the field on a large-scale basis due to the large amount of data analyst resources and time required to process the vast quantity of data generated by the probe. To address this obstacle, EPRI initiated a program to develop automatic data analysis algorithms for array probes.

Objectives

- To develop algorithms that will automatically analyze array probe eddy current data and identify degradation in SG tubes.
- To achieve values for probability of detection (POD) greater than 80% at a 90% confidence level for all known degradation categories.
- To provide data analysis tools that will allow utilities to take full advantage of array probe design by efficiently and accurately processing the vast quantities of data generated by array probes.

Approach

Argonne National Laboratory (ANL) was contracted by the Department of Energy (DOE) to develop algorithms for automatically detecting, classifying, and characterizing degradation in SG tubes based on array probe inspection data. At the conclusion of the DOE project, EPRI continued funding the algorithm development efforts and began validating the algorithms on both laboratory and field array probe data. Algorithm development focused on axially and circumferentially oriented cracks. Tube regions included tube support plates (TSPs), U-bends, and free span. Algorithm results were compared to both manual data analysis results and metallographic results. Following a successful performance demonstration on EPRI's Automated Analysis Performance Demonstration Database, the analysis algorithms can be incorporated into a field data analysis system.

Results

Automated Non-Destructive Testing Array Evaluation System (ANTARES) software algorithms were developed for use on X-probe™ data. Data used for algorithm development and preliminary performance determination was obtained from the U.S. Nuclear Regulatory Commission (NRC) mock-up at Argonne and from data provided by EPRI. Filtering algorithms were introduced to reduce ovalization effects in the U-bend region. A semi-automatic segmentation algorithm was developed based on principal component analysis and filtering. Region-specific detection was implemented.

The algorithms were tested on EPRI Examination Technique Specification Sheet (ETSS) data and missed one of fifteen defects. Difficulties in discriminating between inside diameter (ID) and outside diameter (OD) defects were observed and the classification algorithm was modified to report two defect classes only: "indication" and "non-detectable degradation." The maximum depth of large defects was estimated with good accuracy; however, detailed point-by-point profiling algorithms usually implemented with rotating probe signals do not seem appropriate for X-probe signals since there is a large uncertainty in phase and amplitude variation.

Preliminary testing in the U-bend region indicated that it is feasible to find filtering and detection parameters to control detection sensitivity and false alarm rate. Detecting small defects will probably require better algorithms for TSP removal.

EPRI Perspective

Array probes have potential to provide the speed of bobbin coil probes with the accuracy of rotating coil probes. They also could eliminate the need to perform large-scale rotating coil examinations at locations where the bobbin coil probe is not qualified. For utilities to fully benefit from the potential advantages that array probes can provide, automated data analysis software must be an integral part of the inspection system. Array probes, coupled with automated data analysis, could provide efficiency improvements for both data acquisition and data analysis.

A current limitation of array probes is that acquisition of a large quantity of data requiring manual analysis can delay the inspection schedule. Utilities can only realize the benefits of increased inspection speed provided by array probes if they can rapidly analyze the large quantities of data generated by the probes. Successful development of automated data analysis software for array probe eddy current data would provide utilities with a tool to better realize benefits of the array probe design. Utilities would benefit from EPRI-developed automatic data analysis software through cost savings associated with a reduction in data analyst staffing levels, more consistent data analysis, and shorter inspection schedules.

Implementation Statement

This document does not contain any requirements that must be implemented by utilities.

Keywords

Steam generators
Automated data analysis
Eddy current
Array probe

ABSTRACT

The Automated Nondestructive Testing Array Evaluation System (ANTARES) has been developed under the auspices of the U.S. DOE NEPO program as project NEPO 3-2. The development in this project is focused on software algorithms that provide a means to quickly and consistently detect and characterize steam generator tubing degradation from data acquired from eddy current array probes. The software package ANTARES consists of four modules: data preprocessing, defect detection, defect characterization, and tube burst pressure estimation.
In addition to algorithm development and software implementation, performance testing with laboratory grown flaws has also been carried out. As a NEPO program project, the development effort has been co-managed by EPRI and DOE. The project has reached the stage where a transition to the transfer of the technology to industry is being performed. This EPRI-sponsored report is a part of that transfer and is a step to facilitate the eventual usage of the algorithms in an industrial environment. It documents the theoretical basis of the algorithms for signal and image processing and machine learning in ANTARES, software implementation, and initial testing with laboratory grown flaws. Modifications that were performed to adapt the ANTARES algorithms and software for implementation in the EPRI AutoAnalysis software are also reported.

Defect detection is based on a combination of linear and nonlinear image filtering and restoration techniques with a sensitive statistical hypothesis test, the Sequential Probability Ratio Test (SPRT). Defect characterization is based on nonlinear regression analysis and empirical rules. The detection and characterization algorithms have been modified in the version that is planned to be implemented in the AutoAnalysis software. The follow-on tube burst pressure estimation algorithm is based on an analytical model and an advanced machine learning technique, the Support Vector Machine (SVM) algorithm. This algorithm will not be implemented in AutoAnalysis.

Defect detection performance of the original version before the modifications has been evaluated with two data sets collected from the U.S. Nuclear Regulatory Commission steam generator mock-up at Argonne. Each data set includes 25 tubes. The first test set contains predominantly axial outer diameter stress corrosion cracks (SCCs) near the tube support plate. The second set contains various types of SCCs. The estimated probability of detection for cracks larger than 30%-40% through wall (TW) is larger than 60% at the 90% lower confidence limit. Probability of detection for cracks larger than 80% TW is 85% at the 90% lower confidence limit. Probability of detection for cracks less than 30% appears to be small. Reduced probability of detection is observed for defects near dents and at tube roll transitions.

Defect characterization performance of the original version has been assessed on a set of eight tubes. This is the only available set of laboratory grown flaws with defects that have been inspected using array probes and then validated by destructive analysis. Depending on difficulties in artifact and noise removal, there can be considerable variation in agreement of the estimated depths with fractography results. Defect length is usually underestimated, but an exception to this rule is possible overestimation for small defects.

The modified detection and classification/characterization ANTARES algorithms, which are to be transferred to AutoAnalysis have been tested on the Examination Technique Specification Sheets (ETSS) dataset and limited field data. Analysis speed has been significantly enhanced and performance results on the limited data are similar to the results obtained with the original version of ANTARES.

Performance testing of the burst pressure estimation algorithm is based on a sample of seven experimental burst pressure measurements, since burst pressure experimental data are very limited. Testing results are good for three cases, while in two cases ANTARES conservatively reports that the steam generator tube ligament has ruptured. The analytical model overestimates the experimental data significantly for a number of cases. Since the SVM model is trained on analytically-generated data, the SVM estimation performs similarly. However, the SVM results have the potential to draw closer to the experimental results if the training is performed on experimental data.

ACKNOWLEDGMENTS

The authors thank the U.S. Nuclear Regulatory Commission for their cooperation with this program. This project was co-managed by the U.S. Department of Energy (DOE) and EPRI and for this we are very grateful.

ACRONYMS

ANL	Argonne National Laboratory
ANTARES	Automated Non-Destructive Testing Array Evaluation System
DB	Database
DOE	Department of Energy
EC	Eddy Current
EDM	Electro-Discharge Machining
EPRI	Electric Power Research Institute
ETSS	Examination Technique Specification Sheets
GUI	Graphical User Interface
ID	Inner Diameter
IGA	Inter-granular Attack
LCL	Lower Confidence Limit
KKT	Karush-Kuhn-Tucker condition
NE	ANL Nuclear Engineering Division
NEPO	Nuclear Energy Plant Optimization
NRC	Nuclear Regulatory Commission
OD	Outer Diameter
P_b	SGT Unstable Burst Pressure
P_l	SGT Ligament Rupture Pressure

pdf	Probability Distribution Function
POD	Probability of Detection
QO	Quadratic Optimization
ROI	Region of Interest
RPC	Rotating Pancake Coil
SCC	Stress-Corrosion Cracks
SDM	Signal Disturbance Magnitude
SGT	Steam Generator Tube
SNR	Signal to Noise Ratio
SPRT	Sequential Probability Ratio Test
SRM	Structural Risk Minimization
SVM	Support Vector Machines
TSP	Tube Support Plate
TW	Through-Wall

CONTENTS

***1* INTRODUCTION 1-1**

***2* ANTARES ALGORITHM LOGIC FLOW 2-1**

2.1 Main Integrating Algorithm 2-1

2.2 Interfacing between the Algorithms 2-3

2.2.1 Signal Preprocessing Module 2-6

2.2.1.1 Image Formatting 2-6

2.2.1.2 Image Segmentation 2-6

2.2.1.3 Additional Preprocessing 2-7

2.2.2 Defect Detection Module 2-7

2.2.2.1 Initial Background Estimation 2-7

2.2.2.2 Defect Detection 2-7

2.2.2.3 Morphological Operations 2-8

2.2.2.4 Final Thresholding and Region of Interest Extraction 2-8

2.2.3 Defect Classification/Characterization Module 2-8

2.2.4 Burst Pressure Estimation Module 2-9

2.2.4.1 Overall Approach 2-9

2.2.4.2 Burst Pressure Database 2-9

***3* DETECTION ALGORITHMS 3-1**

3.1 The Sequential Probability Ratio Test (SPRT) 3-1

3.2 Image Filtering 3-2

3.3 Morphological Operations and Thresholding 3-3

***4* CLASSIFICATION/CHARACTERIZATION ALGORITHM 4-1**

4.1 Defect Characterization 4-1

4.2 Calibration Functions 4-2

4.2.1 ID Defects 4-2

4.2.2 OD Defects 4-5
4.3 Classification Rules 4-5

5 BURST PRESSURE ESTIMATION ALGORITHMS 5-1
5.1 Equivalent Rectangular Crack Profile 5-1
5.2 Analytical Model for P_l and P_b Estimation 5-3
5.3 Support Vector Machine Algorithm 5-4

6 LIGAMENT RUPTURE AND BURST PRESSURE DATABASE 6-1

7 BURST PRESSURE ESTIMATION – RANGE OF APPLICATION 7-1

8 DETECTION ALGORITHM TESTING 8-1
8.1 Original Version Test 1 8-3
8.2 Original Version Test 2 8-6
8.3 Original Version Post Test Examination 8-9
8.4 ETSS Data 8-19
8.5 U-Bend Data 8-20
8.6 Full Tube Length Data 8-25

9 CHARACTERIZATION ALGORITHM TESTING 9-1
9.1 Fractography Benchmark 9-1
9.2 Profile Comparison 9-3

10 BURST PRESSURE ESTIMATION TESTING 10-1
10.1 Module Verification 10-1
10.2 Module Validation 10-4

11 CONCLUSIONS AND FUTURE WORK 11-1

12 REFERENCES 12-1

A MODIFIED SOFTWARE DESCRIPTION A-1
A.1 Detection Interface A-1
A.1.1 Detection Interface Documentation A-3
A.1.2 Auxiliary Functions A-17
A.2 Classification Interface A-18

A.2.1 Classification Interface Documentation A-19
A.2.2 Auxiliary Functions A-24
A.2.3 MATLAB Data Files A-24

LIST OF FIGURES

Figure 2-1 ANTARES Structure Overview 2-2
Figure 2-2 Detection and Characterization Module in ANTARES 2-4
Figure 2-3 Modified Detection and Classification Algorithms 2-5
Figure 4-1 An Amplitude Calibration Curve for ID Notches 4-3
Figure 4-2 A Phase Calibration Curve for ID Notches 4-3
Figure 4-3 An Amplitude Calibration Curve for OD Notches 4-4
Figure 4-4 A Phase Calibration Curve for OD Notches 4-4
Figure 5-1 SVM P_l and P_b Estimation Module in ANTARES 5-2
Figure 8-1 Default Detection Parameters for Axial Scan 8-2
Figure 8-2 Default Detection Parameters for Circumferential Scan 8-2
Figure 8-3 Probability of Detection for Test 2 8-6
Figure 8-4 Probability of Detection for Test 2 8-9
Figure 8-5 Defect 1 Position 8-10
Figure 8-6 Defect 1 Characterization 8-11
Figure 8-7 Defect 2 Position 8-12
Figure 8-8 Defect 3 Position 8-12
Figure 8-9 Defect 3 Characterization 8-14
Figure 8-10 Defect 4 Position 8-15
Figure 8-11 Defect 4 Characterization 8-16
Figure 8-12 Defect 5 Position 8-17
Figure 8-13 Defect 6 Position 8-17
Figure 8-14 Defect 7 Position 8-18
Figure 8-15 Defect 8 Position 8-18
Figure 8-16 Ovalization Removal for an Axial Defect in Preprocessing Stage 8-21
Figure 8-17 Detection Stage Showing an Axial Defect and a False Alarm Eliminated in Characterization Stage 8-22
Figure 8-18 Ovalization Removal for a Circumferential Defect in Preprocessing Stage 8-22
Figure 8-19 Detection Stage Showing a Circumferential Defect and a False Alarm Eliminated in Characterization Stage 8-23
Figure 8-20 First Principal Component Used for Segmentation 8-26
Figure 8-21 TSP Region (Horizontal and Vertical Components Grouped by Frequencies) 8-27

Figure 8-22 U-Bend Region (Horizontal and Vertical Components Grouped by Frequencies) 8-28
Figure 8-23 Free Span Region (Horizontal and Vertical Components Grouped by Frequencies) 8-29
Figure 8-24 Detection in U-bend Region, Tube 031110 8-30
Figure 8-25 Large Volumetric Wear Characterization 8-31
Figure 8-26 Amplitude and Phase Patterns for Smaller Wear 8-31
Figure 8-27 Amplitude and Phase Patterns for Incomplete TSP Removal, Tube 031110 8-32
Figure 8-28 Detection in U-Bend Region, Tube 086053 8-32
Figure 8-29 Amplitude and Phase Pattern for Incomplete TSP Removal, Tube 086053 8-33
Figure 9-1 Depth Estimation Evaluation 9-2
Figure 9-2 Length Estimation Evaluation 9-2
Figure 9-3 Profile Estimation for Sample 1 9-3
Figure 9-4 Profile Estimation for Sample 2 9-4
Figure 9-5 Profile Estimation for Sample 3 9-4
Figure 9-6 Profile Estimation for Sample 4 9-5
Figure 9-7 Profile Estimation for Sample 5 9-5
Figure 9-8 Profile Estimation for Sample 6 9-6
Figure 9-9 Profile Estimation for Sample 7 9-6
Figure 9-10 Profile Estimation for Sample 8 9-7
Figure 10-1 SVM vs. Analytical Estimation for SVM Training Data in the P_l Database 10-3
Figure 10-2 SVM vs. Analytical Estimation for SVM Training Data in the P_b Database 10-4
Figure A-1 Detection Interface Flow Diagram A-2
Figure A-2 Classification Interface Flow Diagram A-18

LIST OF TABLES

Table 4-1 General Classification Rules 4-6
Table 6-1 Experimental Points in P_i Database 6-1
Table 7-1 Rules for Reporting the Burst Pressure Estimation Module Results in ANTARES 7-2
Table 8-1 Table of Test Outcomes 8-3
Table 8-2 Initial Detection Results for Test 1 8-4
Table 8-3 Final Detection Results for Test 1 8-5
Table 8-4 Initial Detection Results for Test 2 8-7
Table 8-5 Final Detection Results for Test 2 8-8
Table 8-6 ETSS Testing Results for Conservative Depth Estimation 8-19
Table 8-7 Modifications in the Default Detection Settings 8-20
Table 8-8 Preliminary Results for U-Bend Region 8-24
Table 8-9 The Number of Indications for X-Probe Testing Data 8-25
Table 9-1 Fractography Benchmark Results 9-1
Table 10-1 Test Results 10-2
Table 10-2 Comparison between Burst Pressure Estimation Module and Experimental Results 10-5

1
INTRODUCTION

The Automated Nondestructive Testing Array Evaluation System (ANTARES) has been developed under the auspices of the U.S. DOE NEPO program as project NEPO 3-2. The general objective is to develop and test an advanced eddy current inspection system to improve the detection and characterization of defects in steam generator tubes. The development in this NEPO project is focused on software algorithms that provide a means to quickly and consistently detect and characterize steam generator tubing degradation from data acquired from eddy current array probes. A newly designed array probe has recently been introduced into the nuclear industry for steam generator tube inspections [1-3]. The use of this probe has the potential of significantly shortening inspection schedules and reducing inspection costs. The current limitation of the probe is that the quantity of the data acquired cannot be efficiently analyzed in a manual mode. With the large amount of data produced by high-resolution probes, automated techniques are expected to play an important role toward minimizing the difficulties that are currently encountered with prototype arrays that are under development by the industry. Automating the data analysis stage of in-service inspection (ISI) rests on the development of modern computer-based algorithms that incorporate various signal processing and data analysis schemes involving frequency and spatial domain filters in combination with linear and nonlinear multivariate data analysis models [4-8]. Most of the automatic algorithms have been developed for single coils [4, 5]. Recently, ongoing development of automatic algorithms for array probes has been reported [6-8]. The software package ANTARES has been developed to address the need for automated analysis of array probes. The package consists of four modules: data preprocessing, defect detection, defect characterization, and tube burst pressure estimation. In addition to algorithm development and software implementation, performance testing with laboratory grown flaws has also been carried out. As a NEPO program project, the development effort has been co-managed by EPRI and DOE. The project has reached the stage where the transfer of the technology to industry is being performed. This EPRI-sponsored report is a part of that transfer and is a step to facilitate the eventual usage of the algorithms in an industrial environment. The report documents the basis of the automated eddy current array probe data analysis algorithms for tube flaw detection in ANTARES and the modifications selected to adapt the algorithms and software for implementation in the EPRI AutoAnalysis software. Both ANTARES detection and classification/characterization algorithms have been modified for implementation in EPRI AutoAnalysis software. While the follow-on ANTARES tube burst pressure estimation machine learning algorithms are not being implemented in AutoAnalysis, they are also included in this report for completeness. The performance testing of the detection algorithms with laboratory grown flaws is presented here for both the original and modified algorithms. Limited field data testing results for the modified algorithm are also described.

Defect detection performance with the original ANTARES algorithms is evaluated with two data sets collected during X-probe™ inspection of the U.S. Nuclear Regulatory Commission steam generator mock-up at Argonne. Each data set includes 25 tubes. The first test set contains predominantly axial OD stress corrosion cracks (SCCs) near the tube support plate (TSP). The second set of 25 tubes contains miscellaneous types of SSCs.

The number of laboratory grown flaws with defects that have been inspected using X-probe™ array and then validated by destructive analysis is limited. Defect characterization performance assessment is therefore based on a small benchmark test of eight tubes.

Detection and classification/characterization algorithm modifications for AutoAnalysis were tested using the Examination Technique Specification Sheets (ETSS) and limited field data.

Burst pressure experimental data are very limited. Performance testing of the estimation algorithms implemented in ANTARES is based on a sample of seven experimental burst pressure measurements.

2
ANTARES ALGORITHM LOGIC FLOW

Eddy current testing (ECT) techniques are routinely used for inspection of steam generator (SG) tubes in nuclear power plants. Early detection and characterization of defects contributes to improvement in safety and reliability. The objective of the automated data analysis system ANTARES is to aide the NDE field inspector by providing real-time assistance with defect detection and characterization using array probe data. However, in between the acquisition of the eddy current voltage signals by the array probe and the automatic algorithms for defect detection and characterization in ANTARES, are a number of data processing steps and hardware components. The array probe traversing the steam generator tube provides the modulated analog voltage signals. This signal information is amplified and digitized by EC instruments which produce what will be referred to as raw EC data. This raw EC data is then transmitted to UNIX workstations or PCs where software packages such as EddyView™ [9] or its equivalent, convert the raw EC data into calibrated data which can be analyzed. Options are also available in such packages for mixing or filtering the data. Calibration follows the industry standards for each of the frequencies. Scaling, rotation, etc., are all part of the standard procedure. Once the EC data has been so processed then it is ready to be transmitted to an interface which produces ANTARES-format compatible data. This data is then fed into ANTARES. The various signal and image processing and data analysis schemes in ANTARES, involving linear and nonlinear filtering in combination with statistical detection methods and nonlinear multivariate data analysis models produce flaw detection and characterization information that can be used by the field inspector to enhance the confidence of the data interpretation. Machine learning algorithms are also provided for estimating tube burst pressure as a follow-on to the flaw detection and characterization.

2.1 Main Integrating Algorithm

The basic structure of the data analysis scheme implemented in ANTARES is presented in Figure 2-1. ANTARES is implemented in MATLAB [10]. It consists of a collection of MATLAB procedures that are embedded as source code into user-accessible applications.

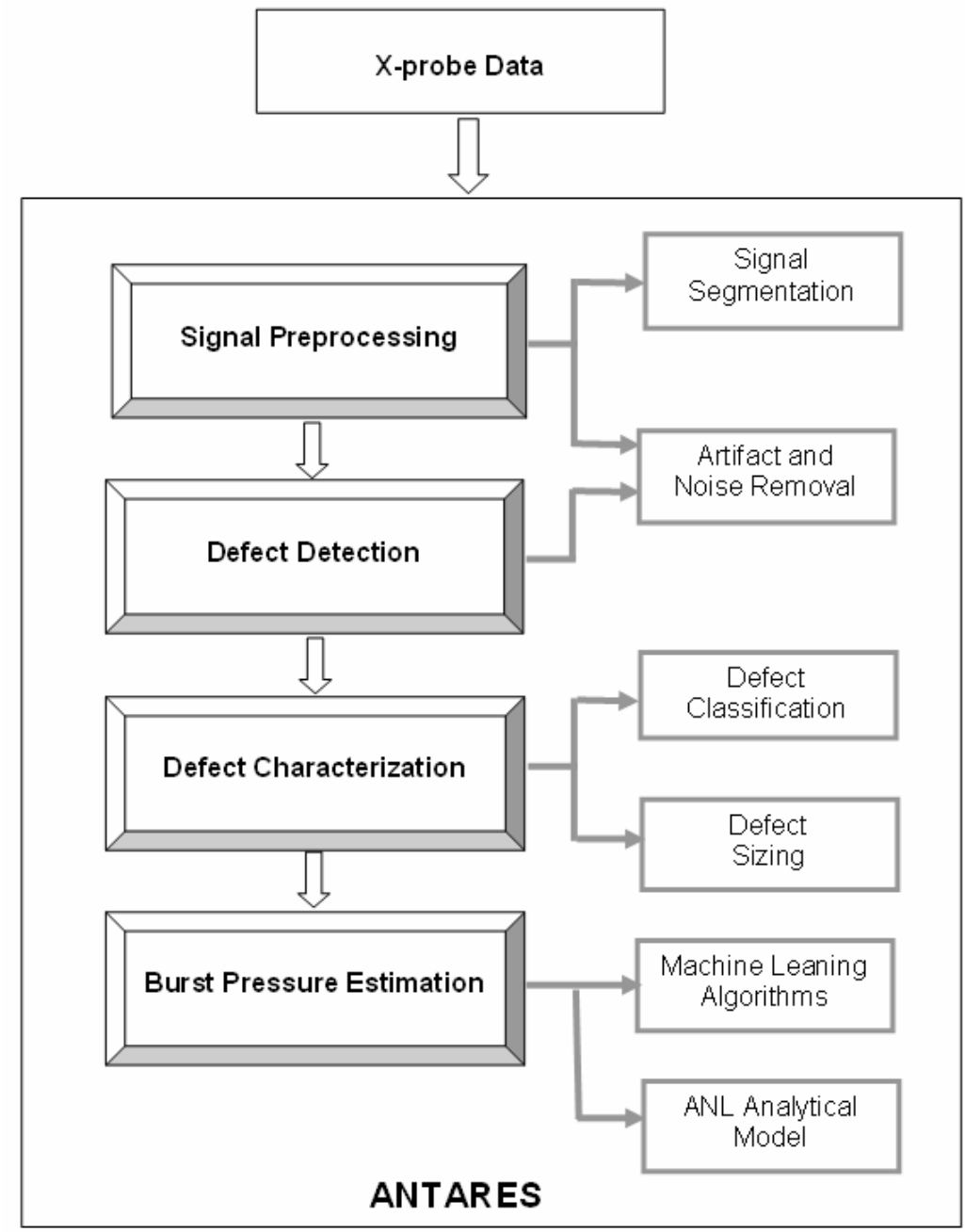

Figure 2-1
ANTARES Structure Overview

To facilitate discussion of the algorithms specifically and the function and interfacing between the algorithms, the discussion will be couched in terms of the various ANTARES modules and the logic flow between them. These are signal preprocessing algorithms, defect detection algorithms, defect characterization algorithms and burst pressure estimation algorithms. The main ANTARES algorithms have been implemented in four modules:

1. Signal Preprocessing
2. Defect Detection
3. Defect Characterization
4. Burst Pressure Estimation

This report describes these modules in the current version of ANTARES. These algorithms have been developed during the period of and by the NEPO project [11-14]. The implementation is specific for X-probe™ applications. Modifications of the software in preparation for the transfer to AutoAnalysis are also described.

The Signal Preprocessing module includes algorithms for data segmentation and noise and artifact removal based on filtering techniques. The segmentation procedure in the pre-processing stage is specific for the NRC steam generator mock-up data and alterations will have to be made for other data sources.

The Defect Detection module is based on the Sequential Probability Ratio Test (SPRT) and image processing algorithms. Chapter 3 describes these algorithms.

The Defect Characterization module is based on nonlinear regression. The calibration curves and the rules for the defect sizing algorithm are specific for the X-probe™ data collected at the frequencies of 70 kHz, 200kHz, 300kHz, and 600 kHz and are modified in the version for AutoAnalysis, since a different set of frequencies is used for the analysis. Chapter 4 describes the algorithm for defect characterization.

The Burst Pressure Estimation module is based on a specific machine learning algorithm, the Support Vector Machine (SVM), and on an alternative mechanistic model based on an analytical approach. The SVM algorithm requires an existing burst pressure database for training (learning). Chapter 5 describes the estimation algorithms while Chapter 6 outlines the burst pressure database.

The details of the logic flow interfacing between the algorithms are described next.

2.2 Interfacing between the Algorithms

Figure 2-2 shows in more detail the information flow for the signal preprocessing, defect detection, and characterization modules and the external interfaces in the original version of ANTARES. The internal interfacing between the modules is also shown. Figure 2-3 shows the modifications in the version for transfer to AutoAnalysis.

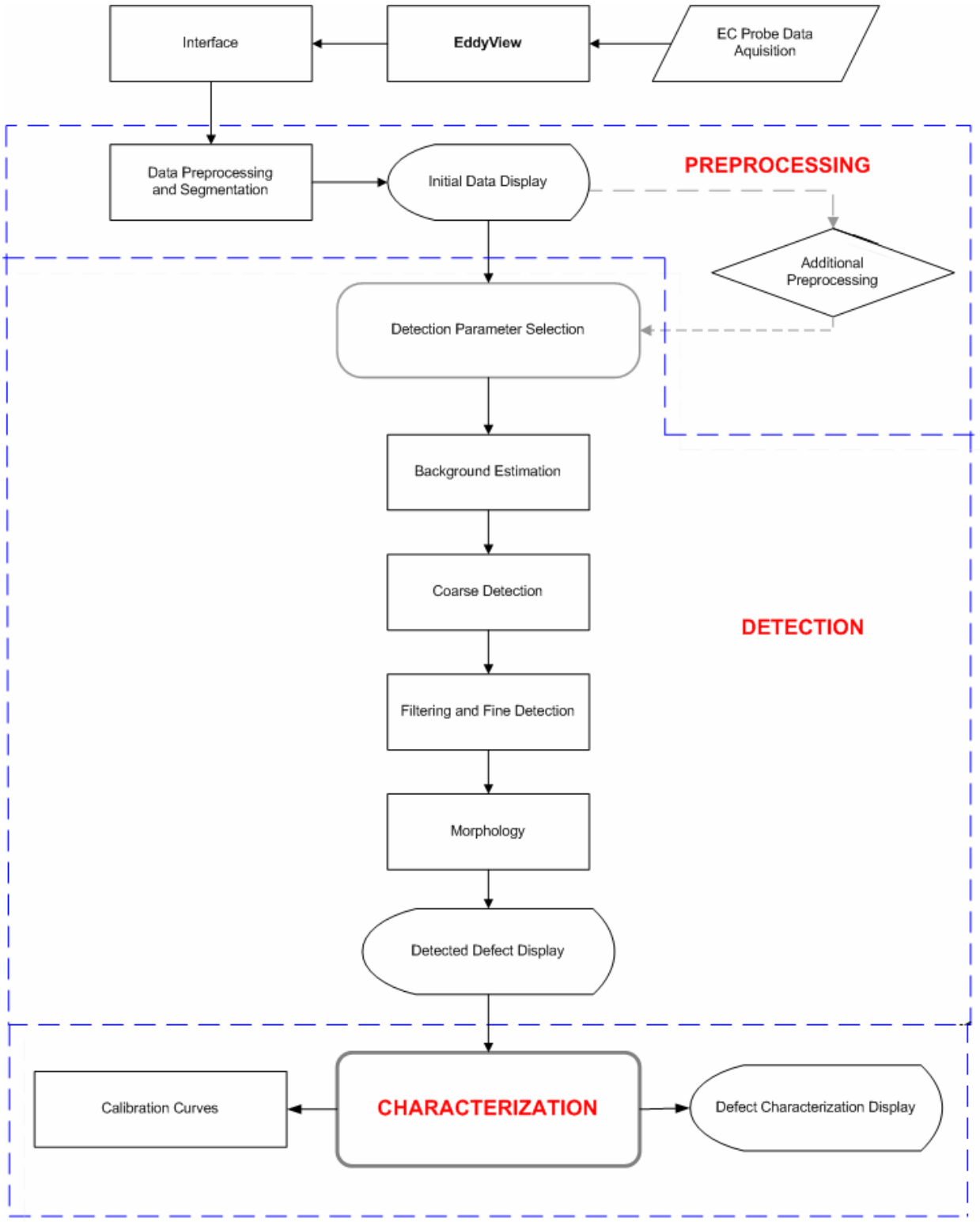

Figure 2-2
Detection and Characterization Module in ANTARES

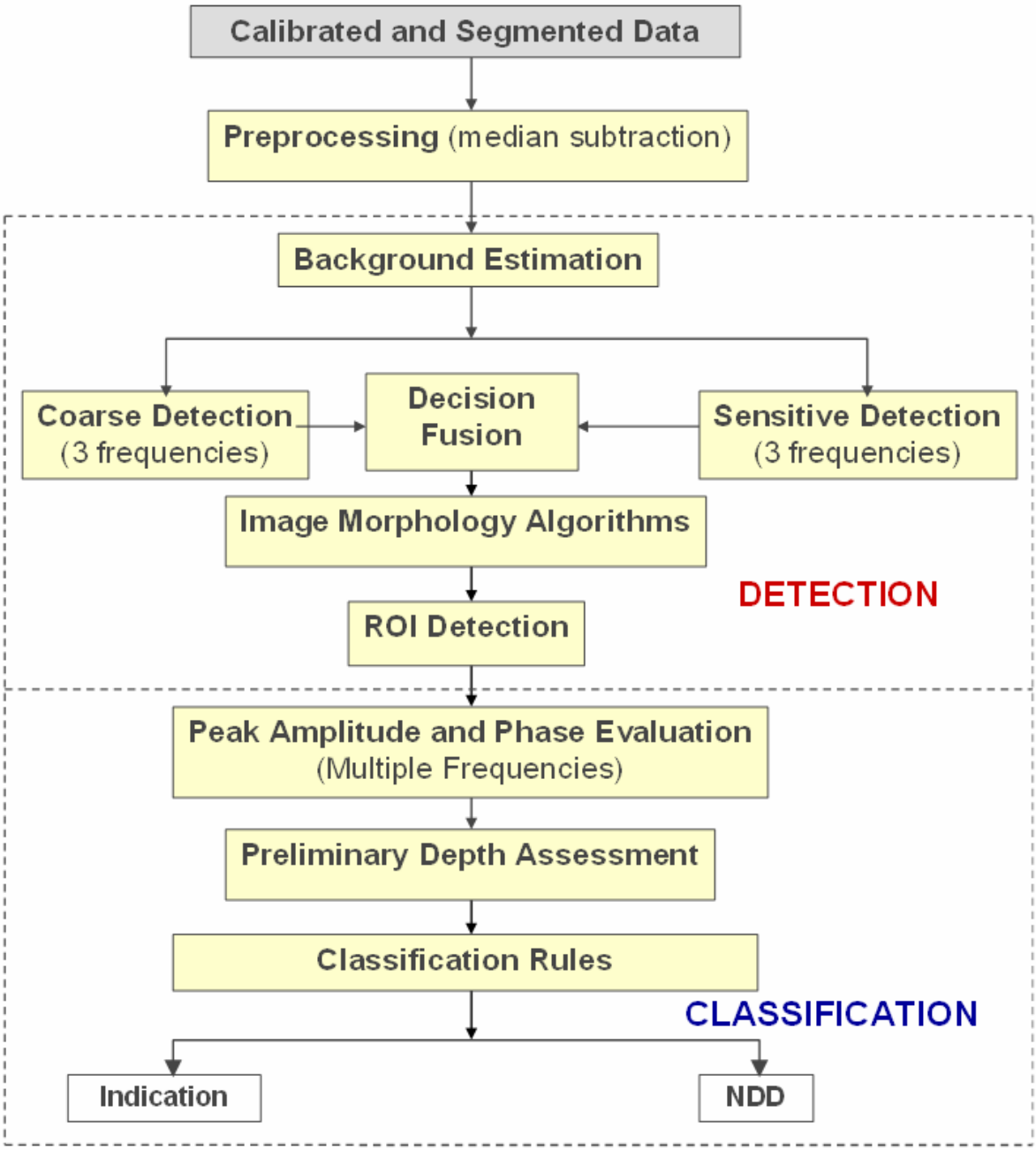

Figure 2-3
Modified Detection and Classification Algorithms

2.2.1 Signal Preprocessing Module

2.2.1.1 Image Formatting

The original assumption for ANTARES input data is that the calibration has already been completed in EddyView™ (or its equivalent). The modified version of ANTARES now uses calibration procedures developed by Michigan State University.

The first preprocessing step is the creation of multi-component images in the MATLAB format. Eight images are used in the present version: four horizontal and four vertical components. All images are stored with rows corresponding to the circumferential direction.

2.2.1.2 Image Segmentation

The second preprocessing step is data segmentation. In the original version, segmentation is based on the specific structure of the U.S. NRC mock-up at Argonne. The landmark signal between segments has a large negative component in the vertical direction at 200 kHz. Information about the magnitude of the landmark signal and the size of the transition region around landmarks is provided by the user. Default values are also provided. After that, automatic data segmentation is performed. The user is prompted to select a background line for each segment, which is then subtracted from all image components. The data preprocessed in this way are used as input to the detection algorithm.

In the modified version, it is assumed that segmentation is performed using an externally provided segmentation algorithm, possibly based on landmarking using bobbin coils, but an automated segmentation algorithm for full length tube data based on the principal component analysis technique has been developed and tested for implementation. Segmentation is needed for full length tubes to improve algorithm sensitivity, since artifacts and noise vary significantly along the tube and specific processing algorithms for some parts of the tube are needed. Therefore, segmentation can be based on specific tube elements. We tested segmentation based on tube support plates. It was found that the circumferential scan at 300 kHz provides the best TSP discrimination, due to low signal-to-noise ratio (SNR), but lower frequencies might be useful as well. Several segmentation methods have been investigated, including edge detection in images [21], wavelet analysis [22], and principal component analysis (PCA) [23]. We have found that the standard edge detection methods used in image processing are not very effective for localized tube support plates. The reason is that most edge detection methods require image gradient evaluation, but transitions in local TSPs are often gradual and noisy, which makes gradient evaluation inaccurate. Wavelet decomposition provides the way to find decomposition levels where edges can be separated from high frequency noise. However, application of sophisticated wavelet decomposition methods for edge detection only is not justified, since simpler methods could be as efficient, but much faster. However, if wavelet decomposition proves valuable for local TSP removal, application of wavelet analysis for segmentation would become more attractive. At present, segmentation is performed using PCA. It has been observed that the axial position of local TSPs can be determined with good precision when the PCA of the circumferential scan data at 300 kHz is applied along the circumferential direction. The first principal component indicates the axial positions of the largest variability. TSP regions are characterized by large negative values and can be found by thresholding. Since preprocessing and detection parameters could vary among the segments, it is important to classify segments properly into free span, TSP, and U-bend regions. A completely automated procedure might be

possible, but for initial implementation we used *a priori* information about the number of TSPs for a given steam generator to determine the number of segments. Due to very limited testing, this algorithm is not included in the code for implementation in AutoAnalysis.

2.2.1.3 Additional Preprocessing

In the original version, line subtraction is performed for each segment based on user interactive input. This is not appropriate for automated procedures and has been substituted by a two dimensional median filter subtraction. Additional preprocessing algorithms may be added in future ANTARES versions. Wavelet de-noising [22] has proved effective for the RPC probes analyzed in our preliminary studies [11]. These preprocessing algorithms might be added as an option for improving the signal-to-noise ratio in characterization algorithms.

Two preprocessing algorithms, automatic initial background estimation and image filtering, are integral parts of the detection algorithm, as described below.

2.2.2 Defect Detection Module

The X-probe™ is designed and calibrated in such a way that the signal components from artifacts are primarily in the horizontal direction. Therefore, the best discrimination between defects and background is achieved at vertical components. Vertical components are selected for detection algorithm, while both components are used for characterization. Due to a good signal-to-noise ratio, the frequency of 200 kHz is used for background estimation. All frequencies available from X-probe™ arrays are used in ANTARES, to enable detection and characterization of both OD and ID defects. The lowest frequency is not used in detection due to very low SNR.

2.2.2.1 Initial Background Estimation

The parameters for automatic initial background estimation are selected through the GUI. A rectangular background neighborhood is defined. This neighborhood is a sliding window for a non-linear filtering operation. The minimum value found in the neighborhood is assigned to the central pixel in the current window. In the first iteration, the mean and the variance of the filtered image are computed row by row and these values are used as input to the SPRT to make initial decision which pixels belong to the background. In the second iteration, only regions determined as a part of the background are used to provide mean and variance for defect detection.

2.2.2.2 Defect Detection

Defects are detected using the SPRT in two steps. Details of the SPRT algorithm are given in Chapter 3. First, a coarse SPRT test is performed to find main peaks, using the mean and variance values from estimated background, and a large detection threshold, obtained by multiplying the small SDM factor used for sensitive detection by a magnification factor. The definition of the SDM factor is given in Chapter 3.

A sensitive detection procedure is used in the second step to enhance the signals from small peaks. The original image is smoothed using a spatial filter. The Wiener filter and a two-dimensional median filter, described in Chapter 3, are implemented. The difference between the original image and the filtered image is used as input to SPRT with a small SDM factor.

The defect map for each frequency is formed as a union of defects found in two steps. The procedure is repeated for all frequencies and the final defect map is determined as a union of all maps. This inevitably leads to false alarms, which are reduced by morphological imaging operations erosion and dilation and additional thresholding as follows.

2.2.2.3 Morphological Operations

Morphological operations are defined in Chapter 3. Erosion is used immediately after SPRT detection to remove SPRT alarms confined to very small regions. A small rectangular structural element was found the most appropriate structural element for erosion in ANTARES. Erosion operation is very useful for removal of small false defects, but produces too sharp peaks for larger defects. This is important for final background estimation and artifact removal. If the region of the defect is too narrow, some parts of the defect signal will be falsely assigned to background, increasing the background signal. Since this signal is eventually subtracted from the original image, the filtered image has reduced peaks. Dilation helps in restoring large peaks. It has been observed empirically that slightly larger neighborhoods should be used for dilation than for erosion. Since another purpose of the dilation operation is to restore peak connectivity in both directions, a diamond shaped structural element seems more appropriate than a rectangular region.

The dilated alarm map is used to mask the regions with defects. The background is extended over the boundaries using interpolation methods. This final background image is subtracted from the original image.

2.2.2.4 Final Thresholding and Region of Interest Extraction

Since morphological operations do not take into account the values of the remaining pixel, some small peaks remain after subtraction and lead to unnecessary false alarms. This usually happens in tubes without defects, where the automatic background estimation algorithms could trigger alarms during sensitive SPRT application. Since most tubes are usually defect-free, this could lead to impractical false alarm rates. To avoid this, a final thresholding is performed. All tubes with number of defect pixels less than a number prescribed by the user are declared to be without defects. In addition, all pixels with magnitude smaller than the given threshold value are removed from the alarm map.

The final threshold is dependent on the magnitude of the signals from X-probe™ and expected minimum detectable values. For the laboratory grown defects, this value was found to be of the order of magnitude of the response of a 20% OD notch. However, to detect very shallow defects, this threshold must be reduced. For a new application, an initial training phase is needed to adjust this value.

The coordinates of the regions of interest (ROI) with potential defects are determined using an automated procedure for object detection in images, described in Chapter 3.

2.2.3 Defect Classification/Characterization Module

The main function of this module is to determine the profile and length of the defects found in the detection algorithm. However, this module also serves for further false alarm elimination.

Although most of the false alarms are eliminated in the detection module using the algorithms described in Chapter 3, some false alarms remain due to the trade off between the false and the missed alarm rate in any detection algorithm. Heuristic rules have been implemented and the final discrimination between defects and non-defects is made at the end of the characterization module. In the original version, final decision about the defect type was left to the user. In the modified version, a set of additional rules has been implemented to enable automated defect classification.

2.2.4 Burst Pressure Estimation Module

The NEPO project methodology implemented in this module uses the ANTARES code estimation of SGT crack characterization as input to estimate the Steam Generator Tube (SGT) ligament rupture and burst pressures (P_l and P_b, respectively). It is based upon a machine learning algorithm. Only axial crack profiles were considered in this research.

2.2.4.1 Overall Approach

Currently, the ANTARES characterization module final output is an estimated crack depth as a function of crack length, based on the information collected by the X-probe™ array. The axial crack profile is then converted into an equivalent rectangular crack profile [15]. This information, along with SGT dimensions, temperature and material properties, is then used to predict P_l and P_b. As shown in Figure 2-1, two methods are used for P_l and P_b estimations. One method, hereafter referred in this report as the analytical model, is based on a flow-stress model developed by ANL under the auspices of a Nuclear Regulatory Commission (NRC) program [15]. The other machine learning methodology offers a data-driven alternative to the analytical model estimations for P_l and P_b, and uses an advanced pattern-recognition method, based on the Support Vector Machine (SVM) algorithm [16]. Here, the SVM was used to solve a regression problem, through a general-purpose code developed by ANL [17].

2.2.4.2 Burst Pressure Database

Similar to most of the data-driven numeric algorithms, the use of the SVM algorithm is divided into two phases: training and estimation. Up to this point, the analytical model was used to generate the training data for the SVM algorithm. Two databases were developed: one for the P_l and the other one for the P_b SVM-based estimations. The analytical model that provides for the most conservative results for P_l and P_b estimations was used in the data generation. Two SVM training cases, which correspond to the most conservative P_l and P_b results respectively, were defined, and the optimum SVM training was performed beforehand for each case. The training results, equivalent rectangular crack dimensions as well as SGT-specific information introduced by the user through a GUI are inputs for the SVM algorithm to estimate P_l and P_b.

Currently, the SVM training was performed on data generated by using the analytical model. Therefore, as shown in this report, the SVM and analytical model estimation are, and should be close. However, this may not be the case when sufficient experimental data are used to train the SVM algorithm. The results presented in this report show that a machine learning based methodology utilizing experimental or hybrid data, as an alternative to a mechanistic-based method would be feasible for P_l and P_b estimations. Whenever new P_l and P_b experimental data points become available, they will be included in the corresponding database. Eventually, the database would be large enough that the SVM training can be performed using only experimental data.

3
DETECTION ALGORITHMS

3.1 The Sequential Probability Ratio Test (SPRT)

The Sequential Probability Ratio Test (SPRT) [18] is a statistical hypothesis test that differs from the standard fixed sample test in the way in which statistical observation are employed. In the fixed sample test, a given number of observations are used to select one hypothesis from two or more alternatives. The SPRT, however, examines one observation at a time, and at some point makes a decision and selects a hypothesis.

In the tests implemented here, normal signal behavior is defined to be that the signal adheres to a Gaussian probability density (pdf) with mean 0 and variance σ^2. Normal signal behavior is referred to as the null hypothesis, H_0. The alternative hypothesis, H_1, is that the signal adheres to a Gaussian pdf with mean M and variance σ^2. In the specific implementation of the SPRT algorithm in this code, the mean of the alternative hypothesis is expressed as a multiple of the standard deviation. The multiplication factor is denoted as the signal disturbance magnitude (SDM) and is defined as

$$M = SDM \times \sigma \qquad \textbf{Equation 3-1}$$

The SPRT algorithm operates as follows. At each time step in a calculation, a test index is calculated and compared to two threshold limits A and B (defined below). The test index is equal to the logarithm of the likelihood ratio (L_n), which for a given SPRT is the ratio of the probability of the observed data sequence $\{Y_2\}= \{y_1, y_2, \ldots y_n\}$ if the alternative hypothesis is true to the probability of the observed data sequence if the null hypothesis is true:

$$L_n = \frac{\text{Prob. of observed sequence}\{\mathrm{Y_n}\}\ \textit{given}\ \mathrm{H_1}}{\text{Prob. of observed sequence}\{\mathrm{Y_n}\}\ \textit{given}\ \mathrm{H_0}} \qquad \textbf{Equation 3-2}$$

Here H_1 corresponds to a defect signal, while H_0 corresponds to a non-defect signal.

The logarithm of the likelihood ratio for the Gaussian distribution defined above can be written in the following form

$$\ln(L_n) = \frac{M}{\sigma^2}\sum_{k=1}^{n}(y_k - \frac{M}{2}) \qquad \textbf{Equation 3-3}$$

If the logarithm of the likelihood ratio satisfies $\ln(L_n) \geq \ln(B)$, then H_1 is accepted. If the logarithm of the likelihood ratio satisfies $\ln(L_n) \leq \ln(A)$, then H_0 is accepted. If the logarithm of the likelihood ratio falls between these values, then new observation is added and test repeated. The threshold limits are

$$A = \frac{\beta}{1-\alpha}$$ **Equation 3-4**

$$B = \frac{1-\beta}{\alpha}$$ **Equation 3-5**

where α is the false alarm probability and β is the missed alarm probability.

The implementation of the basic test in various steps of the defect detection algorithm and the selection of the alternative hypothesis mean for specific steps is described in Chapter 2.

3.2 Image Filtering

Linear and nonlinear spatial filtering techniques [19, 20] have been implemented in ANTARES.

Wiener filter is a linear filter, which is able to adapt to local image variance. Classical pixel-wise Wiener filter output *b(i,j) i=1,2, … N, j=1,2, …M* for a given input noisy image *a(i,j), i=1,2,… N, j=1,2, …M* is given by the following expression

$$b(i,j) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(a(i,j) - \mu)$$ **Equation 3-6**

where the local image mean μ and variance σ are computed in a N-by-M neighborhood

$$\mu = \frac{1}{NM}\sum_{i,j\in\eta} a(i,j)$$ **Equation 3-7**

$$\sigma^2 = \frac{1}{NM}\sum_{i,j\in\eta} a^2(i,j) - \mu^2$$ **Equation 3-8**

and ν is the noise variance. This filter is adaptive. It performs more smoothing when the variance of the image is small. The filter is optimal when the noise is additive, has constant power ("white noise") and the Gaussian distribution.

In addition to the Wiener filter, two other filtering operations have been implemented through sliding window neighborhood. Both filters are nonlinear in nature.

The first is the median filter, which substitutes the central neighborhood pixel by the median value in the neighborhood.

The median of n observations, x_i, *i=1,2,…n* is denoted by $med(x_i)$ and defined by

$$med(x_i) = \begin{cases} x_{(\nu+1)} & \text{n} = 2\nu + 1 \\ \frac{1}{2}(x_{(\nu)} + x_{(\nu+1)}) & \text{n} = 2\nu \end{cases}$$ **Equation 3-9**

where $x_{(i)}$ denotes the i-the order statistic.

A two dimensional median filter applied to image *a(i,j), i=1,2, … N, j=1,2, …M* is defined as

$$y_{ij} = med(x_{i+r,\,j+s};\ (r,s) \in A)$$ **Equation 3-10**

where *A* is a neighborhood of the central pixel *(i,j).* This approach is useful if the image noise is not Gaussian. The other nonlinear filter implemented in ANTARES determines the minimum value in the sliding window and is used for the initial background estimation.

3.3 Morphological Operations and Thresholding

Morphological imaging operations are based on shapes in images.

Erosion and dilation are two fundamental morphological operations. Erosion removes pixels from object boundaries, while dilation adds pixels. The number of pixels removed or added to the objects depends on the size and shape of the structuring element used to process the image. Both erosion and dilation are defined by a structural element.

A structural element consists of a matrix of 0's and 1's. The center pixel identifies the pixel being processed. The pixels in the structuring element containing 1's define the neighborhood of the structuring element. Common structural elements are rectangles, rings and diamond-shaped objects. The state of any pixel in the output image is determined by applying a rule to the processed pixel and its neighbors.

In erosion of binary images, the output pixel is set to 0 if any of the pixels in the input pixel's neighborhood is 0. In dilation, the output pixel is set to 1 if any of the neighborhood pixels is 1.

Morphological image processing does not take into account the magnitude of the detected potential defects. Final thresholding is implemented to remove small defects. In the modified code, different final thresholds could be provided for each of the following segment types: free span, roll transition, tube support region, and U-bend.

Morphological operations are also used for automated ROI extraction. The vertical tube image taken at 300 kHz, after detection and morphology operations, is converted to a binary image and an automated boundary detection function from the MATLAB Image Toolbox [21] is used. Direct implementation of the boundary detection function introduces too large a number of ROIs. Additional rule-based ROI elimination is implemented.

4
CLASSIFICATION/CHARACTERIZATION ALGORITHM

4.1 Defect Characterization

Defect characterization is performed in two steps:

1. Defect extraction and initial sizing;
2. Defect classification and final sizing.

Defect characterization in the original version of ANTARES is based on the available training data collected from the U.S. NRC mock-up at Argonne with the X-probe™ analysis frequencies of 70 kHz, 200 kHz, 300 kHz, and 600 kHz. The characterization algorithm does not use the 70 kHz frequency data. Changes in higher frequencies introduced in some X-probe™ designs required new calibration curves and modifications in heuristic rules for the modified version of the classification/characterization algorithm.

In the original version of ANTARES, at the beginning of the characterization algorithm, the filtered image from the detection algorithm is displayed and the user is prompted to interactively select a defect region for evaluation. In the modified version, all ROI found automatically are further analyzed without user intervention.

When a defect region is selected, the program then computes the amplitude and the phase for every pixel in the selected region. The pixel that corresponds to the maximum amplitude at the reference frequency is determined. The reference frequency depends on the assumption of the defect type. In the original version, the reference frequency is 200 kHz for OD defects and 300 kHz for ID defects. A single reference frequency of 300 kHz is used in the modified version. The pixel with the maximum amplitude is used to evaluate amplitudes and phases at other frequencies. This information is used for defect characterization. Heuristic rules are implemented to eliminate some false indications.

In the original implementation, the defect type is determined by inspection of the amplitude and phase behavior displayed on the screen. For example, increase of the amplitude and counterclockwise rotation of the phase with increased frequency, as well as small phase angles indicate ID defects. However, this ideal discrimination is rarely accomplished in practice. The range of phase angles for ID defects is very small. Due to the presence of noise and artifacts and imperfect filtering algorithms, the errors in the phase estimation are large. For large defects, the phases of ID and OD defects could be in the same interval, resulting in substantial uncertainty in the classification of defects to OD or ID type. It is also very difficult to discriminate between small OD defects and artifacts due to imperfect background removal, because the phase angles could be in the same range. Therefore, in the original version the final decision is made by a trained analyst. The program displays the following information for both types of defects: the phase and amplitude at all four frequencies, as well as an estimated profile for both types. The

surface plots of original and final filtered images are also displayed. Based on these displays, one of four possible options can be selected: ID defect, OD defect, non-quantifiable indication, or non-defect. In many cases, the algorithm automatically offers zero-depth estimations for either ID or OD defects. If defects are selected, the estimated type, position, length, and profile are saved in a file. In the modified version, heuristic rules are implemented in the program for automated defect classification, as described in section 4.3.

4.2 Calibration Functions

A simple calibration procedure based on nonlinear regression is implemented. A sophisticated multivariate calibration procedure does not seem appropriate for the following reasons. The number of defect pixels obtained at the original X-probe™ resolution is very small and extraction of a larger number of features is inaccurate. Moreover, the training data are very sparse and complex multivariate regression functions would lead to data overfitting.

Calibration functions are fitted using the standard consisting of 18 EDM notches. The standard data were exported from EddyView™. Both axial and circumferential C scans are analyzed using the detection algorithm with the default detection setup. The final filtered images are used to calculate phase and amplitudes of each notch at all four frequencies used for X probes. The most consistent phase and amplitude behavior is obtained at 200 kHz and 300 kHz for OD notches and 300 kHz and 600 kHz for ID notches. These frequencies are therefore used for calibration curve fitting.

The functional form and the parameters depend on the defect type and the scan type. In addition, the parameters of the fitting functions are frequency dependent. The shape of the depth dependence on the defect signal amplitude suggested an exponential fitting function, while the dependence on the phase difference is best fitted by polynomials. Typical calibration curves for ID defects are given in Figures 4-1 and 4-2, and for OD defects in Figures 4-3 and 4-4. In the figures, the middle line represents the best fit, while the outer lines give the 95% confidence intervals. The curves are the outputs from nonlinear fitting algorithms from the MATLAB Statistical Toolbox [23].

The functional forms used for the depth estimation are given below. For each type of defects there are four depth estimations based on the amplitude and phase at two frequencies.

4.2.1 ID Defects

amplitude (AMP) calibration

axial: $depth\,(\%TW) = 100 \times (1 - \exp(-a_a \times AMP))$ **Equation 4-1**

circumferential: $depth\,(\%TW) = 100 \times (1 - \exp(-a_c \times \sqrt{AMP}))$

phase (ϕ) calibration

axial: $depth\,(\%TW) = p_{a0} + p_{a1} \times \phi + p_{a2} \times \phi^2$ **Equation 4-2**

circumferential: $depth\,(\%TW) = p_{c0} + p_{c1} \times \phi$

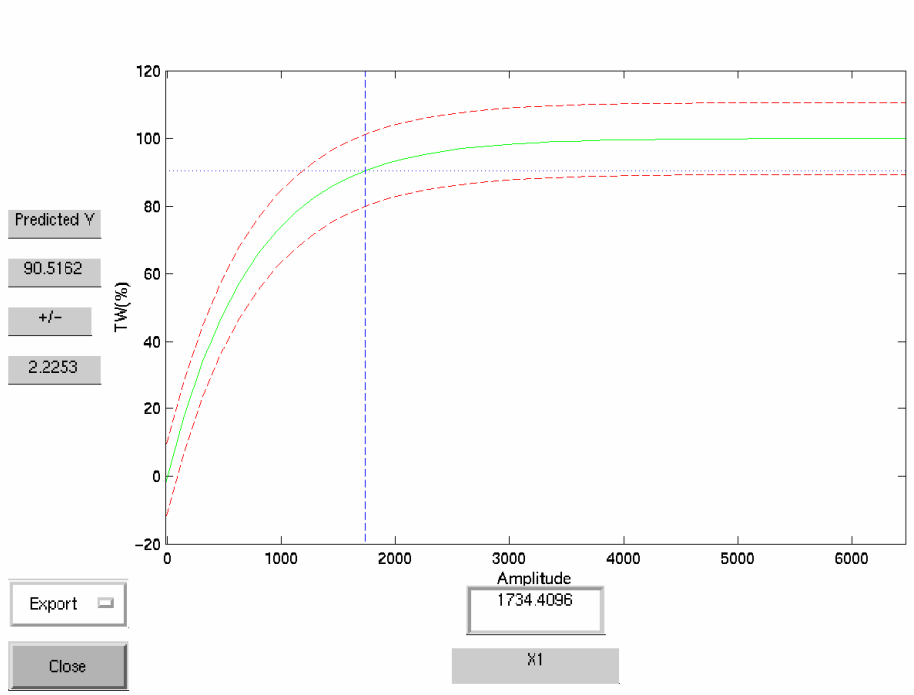

Figure 4-1
An Amplitude Calibration Curve for ID Notches

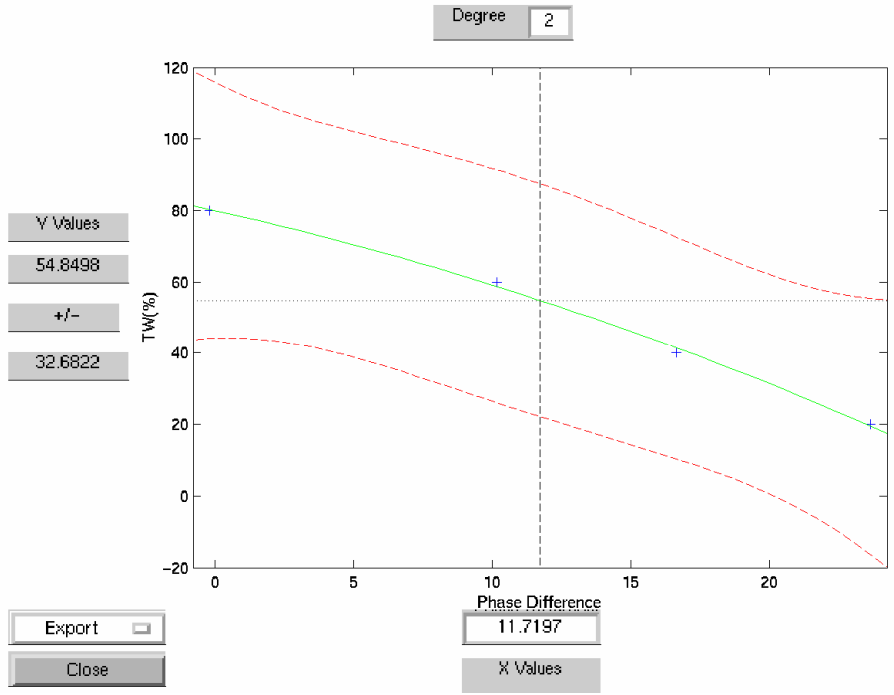

Figure 4-2
A Phase Calibration Curve for ID Notches

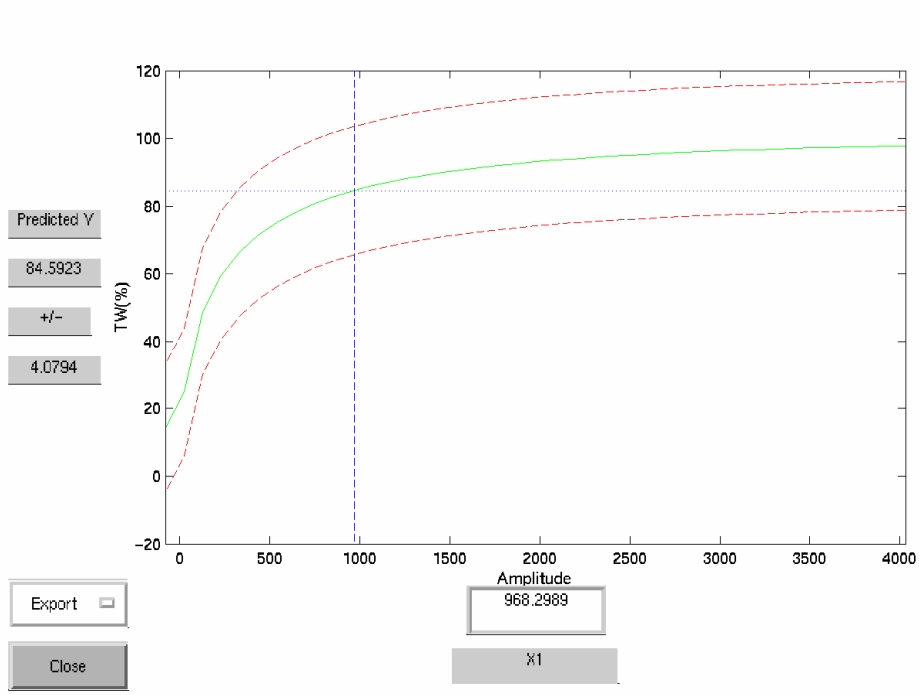

Figure 4-3
An Amplitude Calibration Curve for OD Notches

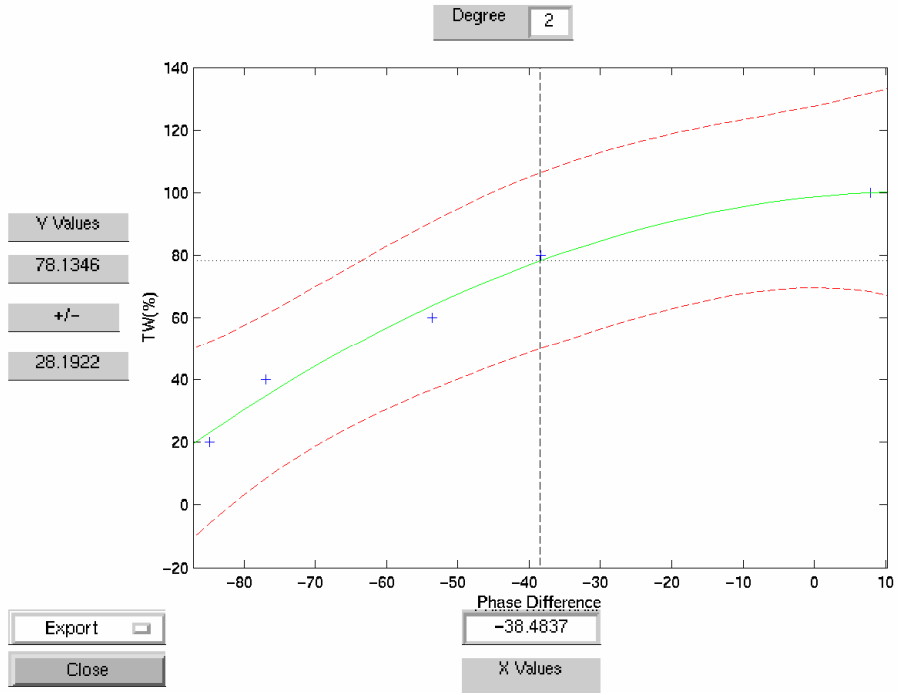

Figure 4-4
A Phase Calibration Curve for OD Notches

4.2.2 OD Defects

amplitude (AMP) calibration

axial: $depth\ (\%TW) = 100 \times (1 - \exp(-b_a \times \sqrt{AMP}))$ **Equation 4-3**

circumferential: $depth\ (\%TW) = 100 \times (1 - \exp(-b_c \times \sqrt{AMP}))$

phase (ϕ) calibration

axial: $depth\ (\%TW) = c_{a0} + c_{a1} \times \phi + c_{a2} \times \phi^2$ **Equation 4-4**

circumferential: $depth\ (\%TW) = c_{c0} + c_{c1} \times \phi + c_{c2} \times \phi^2$

The amplitude functional dependence is best fitted by an exponential function of the square root of the amplitude. The different function for axial ID amplitude functional dependence is the result of smaller fitting error and variance and is not based on prior physical insight. The best fit for the phase calibration is obtained by a second order polynomial, except circumferential ID defect, which are best fitted by a linear function.

Due to changes in the probe design and frequencies used in the modified algorithm, new calibration curves are computed using the same procedure. The X-probe™ data for the 18 notch EDM standard taken at frequencies of 100 kHz, 200 kHz, 300 kHz, and 400 kHz were obtained from X-probe™ developers and used to compute new calibration curves. The analytical forms of the calibration functions are the same, but the coefficients are different. The fitting coefficients for both versions are stored in the MATLAB data files. The modified version uses the fitting coefficients obtained at frequencies of 100 kHz, 200 kHz, 300 kHz, and 400 kHz, since those frequencies are used in recent X-probe™ field examinations. Future modifications in inspection frequencies would require new coefficient evaluation.

4.3 Classification Rules

Heuristic rules have been implemented in the classification/characterization module to discriminate between defects and non-defects. Predicted zero depth is equivalent to non-defect. However, predicted non-zero depth does not necessarily indicate a defect. The decision about the type of indication is made automatically in the modified version by applying a set of rules. General classification rules are given in Table 4-1. Specific implementation contains a combination of these rules and empirical constraints.

Since the details of the classification are not very comprehensive, primarily due to signal distortions during preprocessing, both ID and OD defects and unknown indications are lumped in a single class “Indication” and all indications positively declared as non-defects are classified as “non-detectable degradations” in the modified classification algorithm. Each indication classified as OD or ID is assigned a maximum depth estimate internally in the classification function, based on the calibration curves and sizing algorithm rules. Limited testing is inconclusive about the performance of the sizing algorithm in the modified code and the depth estimate is not reported in the final classification algorithm output.

Table 4-1
General Classification Rules

	Rules Based on Phase	Comments	Decision
1	Clockwise phase rotation with frequency increase Large phase	*Theoretically* should be valid for all frequencies *In practice*, rotation direction is determined by the least square trend fit	OD
2	Counterclockwise phase rotation with frequency increase Small phase	*Theoretically* should be valid for all frequencies *In practice,* rotation direction is determined by the least square trend fit	ID
3	Insignificant phase rotation with frequency increase Phase at each frequency is larger than 100°	Any frequency	No defect
4	Irregular phase rotation Small phase Signal present at high frequencies	Phase errors for ID defect could cause irregularities	ID
	Rules based on amplitude	**Comments**	**Decision**
5	Amplitude less than the specified threshold	For all frequencies	No defect
	Rules based on phase and amplitude	**Comments**	**Decision**
6	After filtering, signal present at only one frequency	Any frequency	No defect
7	Large amplitude Large phase	Lower frequencies	No defect
	Rules based on sizing		
8	Irregular rotations Signal not present at all frequencies Sizing indicates zero depth or depth less than 5% TW	Based on some rules already included in the sizing algorithm	No defect
9	Irregular rotations Signal not present at all frequencies Sizing indicates non-zero depth None of the previous rules apply	Sizing could be very inaccurate when applied to non-defect signals	Unknown
10	Estimated depth is less than a prescribed threshold	Could cause some missed defects	No defect

5
BURST PRESSURE ESTIMATION ALGORITHMS

This section presents theoretical considerations of the two models included in ANTARES that estimate P_l and P_b. The section contains three main subsections. The first subsection discusses the ANL equivalent rectangular crack model. The second subsection considers the ANL analytical model for P_l and P_b estimation. The last subsection outlines the theoretical aspects for the SVM algorithm.

5.1 Equivalent Rectangular Crack Profile

The analytical model for P_l and P_b estimations implemented in ANTARES is based on throughwall (TW) and part-throughwall rectangular axial cracks (see Section 5.2). In contrast to rectangular electro-discharge machining (EDM) notches, the stress-corrosion cracks (SCC) that occur in operating SGT are irregularly shaped, have variable depths along their lengths and may have one or more TW penetrations in localized regions. Similarly, the ANTARES crack characterization module would have as output an estimated irregular axial crack profile [13].

Currently, no widely accepted models are available for predicting the ligament rupture pressure of cracks with such complex geometries. From a limit analysis viewpoint, it can be argued that the collapse behavior of a crack tip ligament with an irregular point-by-point variation of crack depth should be similar to that of a crack with a smoothed-out, “average” crack depth profile [15]. Therefore, in this research, the axial crack profile estimated by the ANTARES crack characterization module was assumed to be equivalent to a rectangular crack profile, as if the SGT behaves similarly in the two cases during the plastic collapse of the ligament.

In this research, the axial equivalent rectangular crack profile was conservatively selected to be the rectangular crack with the lowest estimated P_l from a set of candidates with various lengths and depths [15]. The candidate crack dimensions were chosen by equating the equivalent rectangular area to the original crack area predicted by ANTARES.

The above procedure was coded into a MATLAB function, which is used at the front end of the interface between the characterization and burst pressure estimation modules in ANTARES. Reference should be made to Figures 2-1 and 5-1. The equivalent rectangular crack dimensions are used in both SVM and analytical model estimations for P_l and P_b.

In the analytical model for P_b, the estimation does not consider the equivalent crack depth (see Section 5.2), as if the whole ligament disappears and the TW crack has an equivalent crack length. This assumption is very approximate, but may be very conservative. Also, the P_l analytical model is based on the limit analysis viewpoint discussed above and may also be conservative. A more advanced analytical model for P_l and P_b estimation is currently under

consideration at ANL [24]. The model will be an extension of the equivalent rectangular crack model but will also account for the crack growth. Thus, the future model will have the potential to predict incremental ligament rupture after the initial ligament rupture event, until the full ligament ruptures or the SGT fails unstably (bursts). For now, and to be on the conservative side, a set of rules were implemented in ANTARES to define the limits on the range of applications of the current methodologies.

However, the assumptions discussed above may have a reduced impact on the estimation results, if a fully data-driven approach to estimate P_l and P_b is taken. Assuming that enough experimental data exists to train an SVM model for a specific case, the equivalent rectangular crack dimensions may simply be regarded as SGT features. Thus, such a numeric method would be based only on experimental data.

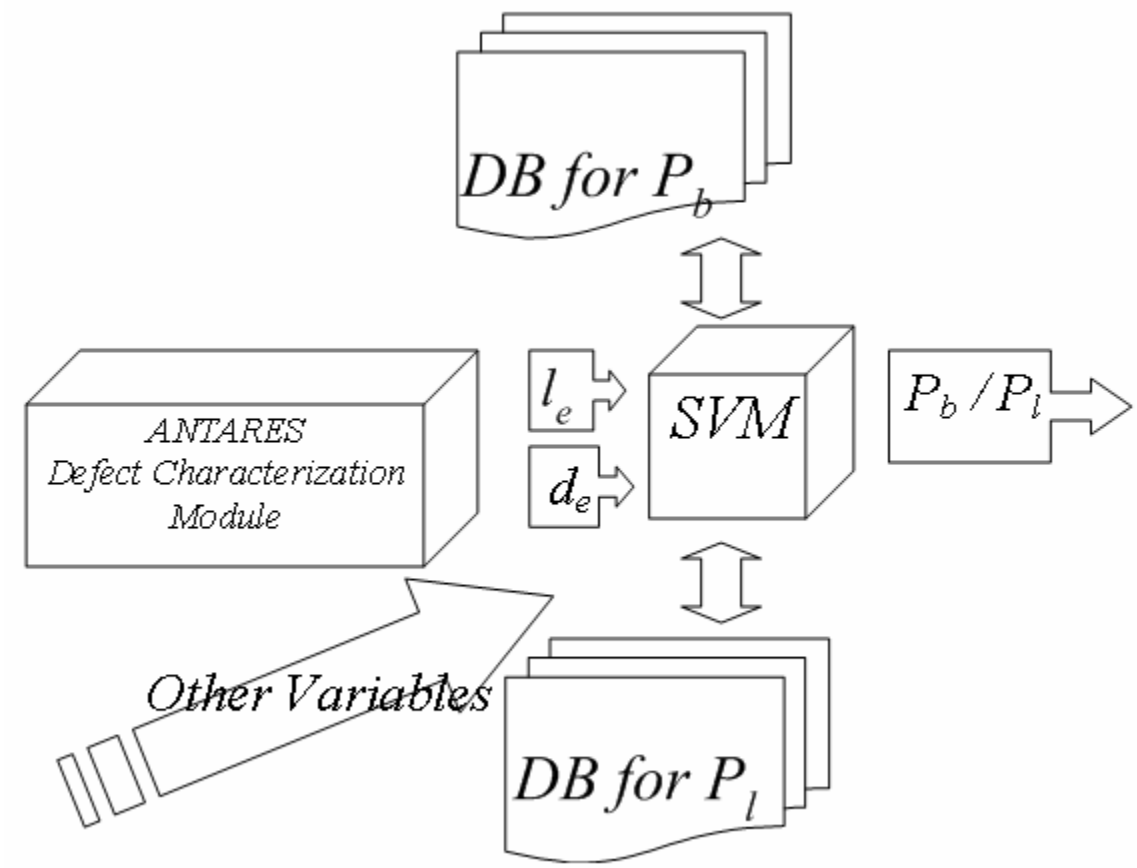

Figure 5-1
SVM P_l and P_b Estimation Module in ANTARES

5.2 Analytical Model for P_l and P_b Estimation

This section describes the analytical model for P_l and P_b estimation as a function of equivalent rectangular crack dimensions, SGT dimensions and material properties. This model was developed by ANL under the auspices of an NRC program [15, 25]. Only P_l and P_b estimations for axial cracks are currently treated in ANTARES and are discussed in this report.

The unstable burst pressure of a thin wall, internally pressurized cylindrical shell with a single rectangular TW axial crack is estimated by an equation originally proposed by Hahn et all [25] and later modified by Erdogan [27]:

$$P_b = \frac{\bar{\sigma} h}{mR} = \frac{P_u}{m}$$ **Equation 5-1**

where:

$\bar{\sigma}$ is the flow stress and equals $k\,(S_y + S_u)$;

S_y and S_u are the yield and ultimate tensile strengths, respectively;

m = 0.614 + 0.481 λ + 0.386 exp(-1.5 λ), the bulging factor;

$$\lambda = \left[12\left(1-\nu^2\right)\right]^{\frac{1}{4}} \frac{c}{\sqrt{Rh}} = \frac{1.82c}{\sqrt{Rh}}$$

$P_u = \dfrac{\bar{\sigma} h}{R}$ is the burst pressure of the unflawed virgin tube;

R and h are the SGT mean radius and wall thickness, respectively;

υ is the Poisson's Ratio, and

$2c$ is the axial crack length.

A general failure criterion for predicting the pressure for the rupture of the crack tip ligament in a tube with a partial TW rectangular axial crack can be expressed as [15]:

$$P_l = m_p P_u$$ **Equation 5-2**

where

m_p, the ligament stress magnification factor, is taken to be [27]:

$$m_p = \frac{1-\alpha\dfrac{1}{mh}}{1-\dfrac{a}{h}};$$

$$\alpha = 1+\beta\left(\frac{a}{h}\right)^2\left(1-\frac{1}{m}\right); \text{ and}$$

a is the rectangular crack depth.

Equation 5-1 considered an average ligament stress for the axial cracks.

The flaw stress factor, *k*, and parameter β are critical factors in predicting P_b and P_l. In this report, a value of 1.0 was chosen for β; *k* was considered to have a value of 0.55 for Alloy 600 SGT tubes that operate at temperatures lower than 100 ºC and 0.506 for higher temperatures. These values were obtained from correlation and best-fit experiments performed at ANL [15, 25].

Another important parameter in P_b and P_l estimation is the SGT material type. The most conservative results are obtained when the Alloy 600 material is annealed at high temperature since the heat treatment diminishes the flow stress. To cover this most conservative case, the training data points for the SVM algorithm were generated using annealed Alloy 600 for which the ultimate and yield tensile strengths in Equation 5-1 were considered to be 607 and 179 MPa respectively [25].

The analytical model described in this section was coded into MATLAB functions. As discussed in Section 5.1, the equivalent rectangular dimensions obtained from the ANTARES characterization module, as well as the above discussed SGT temperature, dimensions and material properties were inputs in estimating the analytical model P_l and P_b.

5.3 Support Vector Machine Algorithm

The SVM algorithm is a universal constructive learning procedure based on the statistical learning theory [16]. It was initially developed for classification problems with separable data and later adapted to solve regression problems. The latter application was considered here to estimate P_l and P_b as a function of the equivalent rectangular crack dimensions profile discussed in Section 5.1.

The SVM combines several distinct concepts and features [33]:

- **New implementation of the Structural Risk Minimization (SRM) principle.** A more robust cost function is considered in the SRM implementation and a regularized risk functional that allows for errors is implemented.
- **Input samples are mapped onto a high-dimensional (feature) space by using a priori determined, non-linear function based on kernel approach.** Thus, the complexity of the problem in the highly-dimensional space is control independently of the feature space dimensionality.

- **Linear functions with constraints on complexity are used to approximate or discriminate the input samples in high-dimensional spaces.** As opposed to many other learning approaches, such as neural networks, in which non-linear approximations are used directly in the sample space, the SVM uses linear estimation techniques in the feature space. The linear estimators provide for accurate estimates for the model complexity and better optimization approaches for finding the global minimum solution for the empirical risk than the non-linear estimators. Thus, smaller data set for training may be required for the SVM than the data set used for training the artificial neural networks.
- **Duality theory of optimization is used to make estimation of model parameters in a high dimensional feature space computationally tractable.** As shown in the optimization theory, solving a dual problem is equivalent of solving the original (primal) problem, if the cost and constraint functions are strictly convex [30]. For the SVM, a quadratic optimization (QO) problem must be solved to determine the parameters of a linear basis function expansion, and for high-dimensional feature spaces, the large number of parameters makes this problem difficult, if not impossible to solve. However, in its dual form, this problem is practical to solve, since it scales in size with the number of training samples, and the linear function corresponding to the solution of the dual problem is given in the kernel representation rather than in the typical basis function representation. The solution in the kernel representation is written as a weighted sum of *support vectors*, which are a subset of the training data that correspond to the solution of the learning problem.

Mathematically, the SVM linear regression problem statement is:

given a set of training data $X = \{(x_1, y_1),...,(x_m, y_m)\} \subset \Im x \Re, \Im \in \Re^d$, find the optimum (smooth) $f(x)=<w, x> + b$, $w \in \Im$, having at most ε deviation for all training data points in X. **Equation 5-3**

In Equation 5-3, w is a set of weights to be determined (optimized), b is a free parameter and $<\,,\,>$ is the inner product in the input space $\Im$ of dimensions d.

The SRM principle provides a mechanism for choosing an optimal model that fits a finite data sample. The SRM principle penalization approach offers a formalism for adjusting (controlling) the complexity of approximating functions that fit available data. This approach is typically employed with adaptive methods, such SVM, that use a wide, flexible set of approximating functions when the true parametric form is unknown. The penalization approach may be also used when the parametric model is known but the number of samples is relatively small. In the SVM formulation, a penalization (regularization) approach to the SRM principle translates to:

find f in Equation 5-3 that minimizes the risk functional

$$R[f] = \int L(x, y, f(x)dp(x, y)) = R_{reg} = R_{emp} + \frac{\lambda}{2}\|w\|^2 \qquad \textbf{Equation 5-4}$$

where

$L(x, y, f(x))$ is the loss (cost) function which measures the discrepancy between the real output and the one produced by the learning machine for a given x;

$p(x, y) = p(x)\, p(y \mid x)$ is the joint probability distribution function (pdf);

R_{reg} is the regularized risk functional;

$R_{emp} = \frac{1}{m}\sum_{i=1}^{m} L(x_i, y_i, f(x_i))$ is the empirical risk functional obtained from averaging the risk over the training data; and

λ is the regularization parameter that controls the penalty relative to R_{emp} and should be independent of the training data.

The regression formulation for the SVM algorithm uses a more robust Vapnik ε-insensitive loss function instead of a standard quadratic loss function, which is appropriate only for the Gaussian noise:

$$|\xi|_{\varepsilon} = \begin{cases} 0, \text{if } |\xi| \le \varepsilon \\ |\xi| - \varepsilon, \text{otherwise.} \end{cases}$$ **Equation 5-5**

By combining Equation 5-3 through 5-5 and by additional reshaping of the SRM principle, we arrive at the following objective (primal) formulation for the SVM regression problem:

$$\textbf{minimize } \frac{C}{m}\sum_{i=1}^{m}\left(\xi_i + \xi_i^*\right) + \frac{1}{2}\|w\|^2$$

$$\textbf{subject to } \begin{cases} y_i - \langle w, x_i \rangle - b \le (\varepsilon + \xi_i) \\ \langle w, x_i \rangle + b - y_i \le (\varepsilon + \xi_i^*) \\ \xi_i + \xi_i^* \ge 0 \end{cases}$$ **Equation 5-6**

where:

$C = \frac{1}{\lambda}$ is a constant that determines the trade-off between the smoothness of f and the amount up to which deviations larger than ε are tolerated; and

ξ_i, ξ_i^* are "slack" variables introduced to cope with the otherwise infeasible constraints of the optimization problem.

It turns out that the optimization problem in Equation 5-6 can be solved more easily in its dual formulation [30]. Moreover, the dual formulation provides the key for extending the SVM to nonlinear, kernel-based functions. The key idea in the dual formulation is to construct a Lagrange function from both the primal function shown in Equation 5-6 as well as the corresponding constraints, by introducing a dual set of variables.

The Lagrange function is given by the primal objective function that needs to be minimized minus the sum of all products between constraints and the corresponding Lagrange multipliers

[30]. Optimization is then seen as a minimization of the Lagrange function with respect to the primal variables or maximization with respect to the Lagrange multipliers, i.e. dual variables. Thus, it has a saddle point at the optimal solution in terms of the primal and dual variables. Here the Lagrange function is used only as a theoretical device to derive the dual objective function.

Following the above procedure, we construct the Lagrange function for the primal objective function and set the partial derivative of the Lagrange function with respect to the primal variables to 0. Then, the following equivalent dual optimization problem results:

$$\textbf{maximize } -\varepsilon\sum_{i=1}^{m}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{m} y_i(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\langle x_i, x_j\rangle$$

$$\textbf{subject to } \begin{cases} \sum_{i=1}^{m}(\alpha_i^* + \alpha_i) = 0 \\ \alpha_i^*, \alpha_i \in \left[0, \frac{C}{m}\right] \end{cases}$$

Equation 5-7

where α_i^*, α_i are the dual Lagrange multipliers.

From setting to 0 the partial derivative of the Lagrange equation with respect to the primal variable *w*, the following results:

$w = \sum_{i=1}^{m}(\alpha_i^* - \alpha_i)x_i$ and therefore the solution of the initial SVM regression problem (Equation 5-3) is:

$$f(x) = \sum_{i=1}^{m}(\alpha_i^* - \alpha_i)\langle x_i, x\rangle + b$$

Equation 5-8

Equation 5-8 represents the support vector expansion, i.e. the weight *w* can be completely described as a linear combination of the selected training data, for which the Lagrange coefficients are not 0. Moreover, the complete algorithm can be described in terms of inner products. These observations are very important for the formulation of a nonlinear SVM.

Before discussing the nonlinear SVM extension, a few remarks are in order for the computation of the free parameter *b* in Equation 5-8. This parameter can be computed by exploiting the so called Karush-Kuhn-Tucker (KKT) conditions which state that the product between the dual variables in Equation 5-7 and the constraints in Equation 5-6 has to vanish at the optimal solution [31, 32]. Several useful conclusions can be drawn from this calculation. One conclusion is that $\alpha_i^*\alpha_i = 0$, i.e. when one dual variable is nonzero, the other one must vanish. Another conclusion is that only for $|f(x_i) - y_i| \geq \varepsilon$ the Lagrange multipliers may be nonzero, i.e. for all samples

inside the ε-sensitive region, α_i^* and α_i vanish. Thus, not all the training data set is needed to describe *w*, and therefore, we have a sparse expansion of *w* in terms of x_i. The training data points for which the Lagrange multipliers are nonzero are called support vectors. Therefore, as discussed at the beginning of this section, the function complexity is independent of the input space dimensionality and depends only on the number of support vectors.

With these observations in mind, the next step is to generalize the procedure and make the SVM algorithm nonlinear. This can be achieved by preprocessing the training data x_i by a map $\Phi: \Im \rightarrow F$ into some feature space *F* and then applying the standard linear SVM regression algorithm described above into the feature space. As noted already, the SVM algorithm only depends on the inner products between various patterns and therefore, it is sufficient to define and use the kernel K(x, x') = <Φ(x), Φ(x')> instead of Φ explicitly. This allows the rewriting of the SVM algorithm as follows:

$$\textbf{maximize } -\varepsilon\sum_{i=1}^{m}(\alpha_i^*+\alpha_i)+\sum_{i=1}^{m}y_i(\alpha_i^*-\alpha_i)-\frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K\left(x_i,x_j\right)$$

$$\textbf{subject to } \begin{cases}\sum_{i=1}^{m}(\alpha_i^*+\alpha_i)=0\\ \alpha_i^*,\alpha_i\in\left[0,\frac{C}{m}\right]\end{cases} \qquad \textbf{Equation 5-9}$$

The solution of the linear SVM regression problem in the feature space is then:

$$w=\sum_{i=1}^{m}(\alpha_i^*-\alpha_i)\Phi(x_i) \textbf{ and } f\left(x\right)=\sum_{i=1}^{m}(\alpha_i^*-\alpha_i)K\left(x_i,x\right)+b \qquad \textbf{Equation 5-10}$$

Note that the difference to the linear case is that *w* is no longer explicitly given and the quadratic optimization (QO) problem in Equation 5-9 corresponds to finding the smooth function in the feature space, not in the input space. However, due to the fact that kernels can be associated with the smoothness properties via regularization operators, the corresponding functions are also smooth in the input space [33]. Moreover, the equivalent kernel representation of the SVM algorithm provides for a generalization of the procedure since the solution depends only on the a priori determined inner product (kernel) in the feature space.

The kernel functions must satisfy several mathematical conditions for the nonlinear algorithm to hold and their selection is an ongoing research area in the current SVM development. Several kernels were considered into the SVM code used in this research:

Gaussian Kernel, $K(x,x')=\exp(-\frac{\|x-x'\|^2}{\sigma^2})$;

Linear Kernel, K(x, x') = <x, x'>;
and Polynomial Kernel, K(x, x') = $(<x, x'>+c)^p$ **Equation 5-11**

In coding the SVM procedure, the most complicated and time consuming part in the learning process is to solve the QO problem in Equation 5-10. The ANL code uses an advanced QO method called the Active Set Algorithm which iteratively searches for the set of variables that satisfy the equality constraints in the QO problem. In addition, the code implements decomposition methods to make large scale problem feasible. Reference [17] contains more information on the general-purpose ANL code that implements the SVM algorithm.

In running the SVM code, at least three important parameters must be selected by the user: ε, C and the kernel K (see Equation 5-9). For the code used in this research a selection of Gaussian kernel introduces another parameter, whereas a selection of the polynomial kernel introduces two other parameters (see Equation 5-11). All these parameters potentially affect the final model complexity.

Rigorous model selection for the SVM regression is still an open issue. As discussed above, the regularization parameter C affects the trade-off between the model complexity and the training error. The optimum value can be adjusted using resampling methods [28].

In the case of regression problems with noise, parameter ε reflects the noise variance. For SVM regression problems without noise (interpolation), the parameter should reflect the pre-specified accuracy of interpolation (larger ε's correspond to smaller number of support vectors) [28].

Based on a small number of empirical results for SVM classification problems it appears that the type of inner product kernel usually does not have much influence on the classification error [16]. For SVM regression problems, preliminary results seem to indicate that the choice of kernel is important [28]. The development and selection on good kernels are still open issues.

In this research, the necessary SVM training data set in Equation 5-3 was generated by using the most conservative analytical model described in Section 5.2. For the SVM model used to estimate the ligament rupture pressure, the output *y* in Equation 5-1 corresponds to P_l and the input space corresponds to the rectangular equivalent crack dimensions discussed in Section 5.1. For the SVM model used to estimate the unstable burst pressure, the output, P_b is formulated as a function of the rectangular equivalent crack length, since in this case the ligament is assumed to have ruptured (100 % TW). As discussed in Section 5.1, this assumption may be too conservative, and may influence the SVM results, since in the current application, the SVM estimations are based on data generated by the analytical model shown in Section 5.2. All the other parameters, such as SGT temperature, dimensions and material properties, were maintained constant during data generation for both SVM models.

SVM training was performed for both most conservative P_l and P_b SVM cases discussed above. During training, optimum SVM parameters that produce the smallest errors in the training data sets were found. For these parameters, the corresponding case-dependent support vectors were also found. This information is used in the burst Pressure Estimation module of ANTARES when SVM estimations for P_l and P_b are computed.

Note that the current SVM implementation is based on training data generated by using an analytical model. However, when sufficient experimental data points for P_l and P_b become available for SVM training, corresponding SVM cases can be defined or updated. Then, the

SVM analysis will be based on experimental data or on hybrid models (experiment + analytical data points) as an alternative to the analytical model estimations.

The procedure described in this section is summarized in Figure 5-1. Note that the SVM block in Figure 5-1 may be replaced by an Analytical Model block. In this case, the connection to the P_l and P_b databases are not required. However, other variables such as SGT properties and temperature need to be specified by the user.

6
LIGAMENT RUPTURE AND BURST PRESSURE DATABASE

This section presents the P_l and P_b databases used in this project for the SVM training. The databases were developed to provide for an effective and reliable information management related to the SGT P_l and P_b. The databases will be maintained and updated and the corresponding optimum SVM models derived whenever new experimental information becomes available.

As discussed in the previous sections, the analytical model was used to generate training data for the SVM algorithm. Also, several SGT ligament rupture and unstable burst pressure experiments supported by NRC programs were performed in the High-Temperature Pressure and Leak-Rate Test Facility and Room-Temperature High-Pressure Test Facility at ANL [15, 34]. Representative and pertinent results were included in two databases, one for P_l and the other one for P_b. Table 6-1 shows the experimental data points in the P_l database.

Table 6-1
Experimental Points in P_l Database

P_l, MPa	L_e, in	D_e, in	Temp, °C	OD, in	Thk, in	Material	Type	Source
12.4	0.25*	0.045*	20	0.875	0.05	Alloy 600, annealed	axial	Exp., SGL-413, lab-grown SCC, [14]
6.2	0.30*	0.049*	20	0.875	0.05	Alloy 600, annealed	axial	Exp., SGL-480, lab-grown SCC, [14]
32	0.55*	0.024*	20	0.875	0.05	Alloy 600, annealed	axial	Exp., SGL-494, lab-grown SCC, [14]
11.5	0.34*	0.046*	20	0.875	0.05	Alloy 600, annealed	axial	Exp., SGL-493, lab-grown SCC, [14]
13.8	1.5	0.04	20	0.875	0.05	Alloy 600, annealed	axial	Exp., T3EA80X1.5, EDM, [33]
10.3	0.33*	0.039*	20	0.875	0.05	Alloy 600, annealed, assumed	axial	Exp., 412 or 535, lab-grown SCC,
13.8	0.58*	0.043*	20	0.875	0.05	Alloy 600, annealed, assumed	axial	Exp., 732 or 855, lab-grown SCC

* The equivalent rectangular crack dimension was computed from posttest fractography profile

The databases were developed as Microsoft Access applications. This database system was preferred since it has powerful import and export capabilities and user-friendly facilities to extract the desired data sets for selective SVM training cases. In addition, the creation or modification of fields, data entries, error checks, tables or queries in the database is handled by an extensive collection of easy to use features.

The SGT database for P_l includes the following parameters (fields):

1. Data point identification;
2. Ligament burst pressure P_l generated using the analytical model or obtained from an experiment, in MPa (column 1 in Table 6-1);
3. Crack length or equivalent rectangular crack length, in inch (column 2 in Table 6-1);
4. Crack depth or equivalent rectangular crack depth, in inch (column 3 in Table 6-1);
5. SGT temperature at which the experiment was conducted or at which the data point was generated, in ºC (column 4 in Table 6-1);
6. SGT outer diameter (OD), in inch (column 5 in Table 6-1);
7. SGT thickness, in inch (column 6 in Table 6-1);
8. SGT material or material identification (column 7 in Table 6-1);
9. Crack type (column 8 in Table 6-1);
10. Descriptive data point information (column 9 in Table 6-1).

The first 7 fields in the P_l database contain numeric data only; the last 3 fields may contain text. Data can be entered in the numerical fields only after passing appropriate error checks introduced during the database design.

The P_b database contains the same fields discussed above, with the exception of crack depth dimension which is assumed to be 100 % TW in this research. Also, the second field in the P_b database corresponds to the unstable burst pressure generated using the analytical model or obtained from an experiment.

The analytical model that provides for the most conservative results for P_l and P_b estimations was used in the data generation. As discussed previously, this corresponds to Alloy 600 SGTs that were annealed at high temperatures. The annealing treatment resulted in a loss of flow stress, which weakened the SGT. Thus, the analytical model for this material consistently provides for the lower P_l and P_b estimates when all the other analytical model parameters are maintained constant. Therefore, efforts were focused to characterize the case that provides for the most conservative analytical results for P_l and P_b estimates.

As discussed in Section 5.2, functions that code the analytical model for P_l and P_b estimations were developed in MATLAB. These functions use the equivalent rectangular crack dimensions in their computations. Also, for the SVM analysis, it was assumed here that P_l and P_b depend only on the SGT equivalent rectangular crack dimensions (for P_b the crack is assumed through wall (TW)). Other variables, such as: temperature and SGT material properties, OD and thickness could be added later in the model, if necessary and/or data is available. Using the MATLAB functions discussed above, 336 analytical data points were generated and considered representative of the most conservative case for the P_l database. Similarly, 40 analytical data points were generated for the P_b database. The following parameters were considered in data generation:

- The equivalent rectangular crack length was uniformly varied from 0.1 in to 2 in, in steps of 0.05 in, during the analytical model calculation of P_b;
- The equivalent rectangular crack length was uniformly varied from 0.1 in to 1.5 in, in steps of 0.05 in, during the analytical model calculation of P_l;
- The equivalent rectangular crack depth was varied from 10 to 90 % through wall (TW) in variable steps, during P_l and P_b analytical model calculation. The data sample density for depths in between 40 and 90 % TW was greater than the corresponding quantity for depths in between 10 and 40 % TW, since the former region is considered to be the region of interest in ANTARES.
- Only axial cracks were considered during both P_l and P_b analytical model data generation;
- SGT at 20°C during both P_l and P_b analytical model data generation;
- Constant SGT dimensions (7/8 in OD and 0.05 in thickness) for both P_l and P_b data generation;
- Alloy 600, annealed at 1100°C for 10 minutes, as SGT material for both P_l and P_b analytical model computations. This material is assumed have ultimate and yield tensile strengths equal to 607 and 179 MPa respectively (see Equation 5-1).

The 336 data points corresponding to the conservative case described above were added to the P_l database. Similarly, the 40 data points mentioned above were used to populate an Access database for P_b computations.

Seven experimental points were also added to the P_l database. These points correspond to the most conservative case above and were taken from references [15, 34]. The experimental data points are presented in Table 6-1.

Whenever new experimental data points become available, the equivalent database will be updated. Eventually, the P_l and P_b databases would include enough experimental data points for each case of interest. Then, the SVM training could be based only on experimental data. In this research, the experimental data points were used for testing the burst pressure module in ANTARES, as described in Section 5. The optimum SVM training was performed by using the generated data points that correspond to the most conservative analytical model case discussed above.

The process of extracting data of interest from the database was automated by using programmable Microsoft Access queries. Queries are an essential tool in any database management system and are used to select records, update tables and add new records to tables. In this research several queries were designed to select group of records that meet specified criteria. For example, the numeric data file used in the SVM training for the SVM P_l model was extracted from the P_l database using a specialized query. Similarly, training data for SVM P_b model was extracted from the P_b database.

Note that the P_l database contains 100 additional data points that were also generated using the analytical model. A different material (Alloy 600 M90 SGT) was used in this data generation; 50 data points are characteristic of 3/4 inch OD tubes and the rest are characteristics of 7/8 inch OD tubes. These records were used in the initial stages of this project in testing the applicability of the SVM algorithm for this problem and database extraction capabilities.

7
BURST PRESSURE ESTIMATION – RANGE OF APPLICATION

This chapter summarizes the range of applications for P_l and P_b estimates that are currently implemented in ANTARES and for which the verification and validation described in Chapter 10 was performed. The ANTARES user should be aware of the following code limitations:

- Both analytical and SVM estimations for P_l and P_b are based on approximating the real crack profile through ANTARES defect characterization module calculations. This estimated profile has associated uncertainties due to several approximations made during the estimation process. In addition, the defect profile estimated by ANTARES is converted into an equivalent rectangular crack profile which represents one of the most important parameters for both SVM and analytical P_l and P_b estimations. However, the methodology used to compute the equivalent rectangular crack dimensions introduces supplementary uncertainties in P_l and P_b estimations since, currently, it does not account for the crack growth (see Section 5.1).
- The code handles only axial cracks since the data used for SVM training considered only this case for now. If the selected crack is circumferential, an error message box appears and the SGT Pressure Estimations are overridden.
- The results are representative for Alloy 600 SGT, with 7/8 in OD and 0.05 in thickness at 20°C. For these cases, the results should be conservative since the SGT material is assumed to be annealed at high temperatures. This material was used in the data generation for the SVM training and is also used in the analytical model computations for P_l and P_b estimates.

In addition, the set of rules summarized in Table 7-1 are implemented in the Burst Pressure Estimation module. The second and third columns in Table 7-1 represent the equivalent rectangular crack depth (ERD in % TW) and length (ERL and in), respectively.

In addition, whenever both P_l and P_b are reported for a corresponding case (analytical or SVM) the following supplementary rule applies:

- If $P_l > P_b$, then P_b is not displayed and the module reports that it concludes that the SGT burst when the ligament ruptures.

The rules described in Table 7-1 account for the fact that limited ranges for equivalent rectangular crack dimensions were used during data generation for SVM training. Therefore, the SVM results are representative only for these ranges. In addition, physical considerations were considered in the supplementary rule above. The implementation of these rules in ANTARES was also tested during the verification and validation of this module.

Table 7-1
Rules for Reporting the Burst Pressure Estimation Module Results in ANTARES

Case ID	ERD (% TW)	ERL (in)	Action
1	ERD < 10	Any	Do not report SVM P_l or P_b. Report analytical P_l and P_b. Inform the user that the crack dimensions are outside the training range for SVM computations.
2	10 < ERD < 90	ERL < 1.5	Report analytical and SVM P_l and P_b.
3	10 < ERD < 90	1.5 < ERL < 2	Do not report SVM P_l. Report both analytical P_l and P_b and SVM P_b. Inform the user that crack length is outside the SVM training range for P_l computation.
4	10 < ERD < 90	ERL > 2	Do not report SVM P_l and P_b. Report both analytical P_l and P_b. Inform the user that the crack length is outside the SVM training range.
5	ERD > 90	Any	Do not compute the SVM P_l and P_b and analytical P_l. Report analytical P_b only. Inform the user that the crack dimensions are outside the training range for SVM computation and the tube ligament is conservatively declared ruptured due to model uncertainties.

8
DETECTION ALGORITHM TESTING

The performance level of the ANTARES detection algorithm is determined by a set of parameters, which can be adjusted for specific applications. A set of default values is provided, but the user can change any of these parameters through the GUI. The default values in the original version were determined during the initial project training phase based on the X-probe™ array data collected at the U.S. Nuclear Regulatory Commission steam generator mock-up at Argonne. In the modified version, the default values are plant specific. A set of default values was obtained during limited testing for U-bend data.

The detailed explanation of the parameter setup is given in the ANTARES user manual [14]. For completeness, a short description is given here. Four groups of detection parameters are defined: *probabilities*, *background*, *detection*, and *morphology*.

Probabilities include the false alarm probability *alpha* (α) and the missed alarm probability *beta* (β) in SPRT. ***Background*** parameters define the initial background estimation algorithm. ***Detection*** parameters define the defect detection algorithm. The *SDM factor* defines the sensitivity of the SPRT test used in each of these algorithms. The parameter *range* defines an appropriate sliding window for spatial filtering. The *SDM factor* is used in both steps of the defect detection algorithm. For the coarse detection step implemented for large peaks, this threshold is multiplied by the parameter *magnification.* The parameter *range* defines the sliding window for filtering. The parameter *threshold* is an empirical factor and is used for final removal of small peaks. *Morphology* parameters define the structuring elements for erosion and dilation. The structuring element for erosion is rectangular, while the dilation structural element is an equilateral diamond shape. *Detection limit* is the maximum number of nonzero pixels in the alarm map of a normal segment.

Default detection parameters are used for performance testing. Parameter settings for axial and circumferential scans are presented to the user through a GUI in the original version. Figures 8-1 and 8-2 depict the default setup for axial and circumferential scans, respectively.

Indications of defects found in the detection module are further examined in the characterization module. A number of heuristic rules have been implemented in the characterization algorithm for false call elimination. At the end of the defect characterization, information about the phase and the amplitude of a selected defect is displayed in a separate window. Based on the information presented on the characterization display, the user makes the final decision about the type of indication, whether or not an actual defect is present.

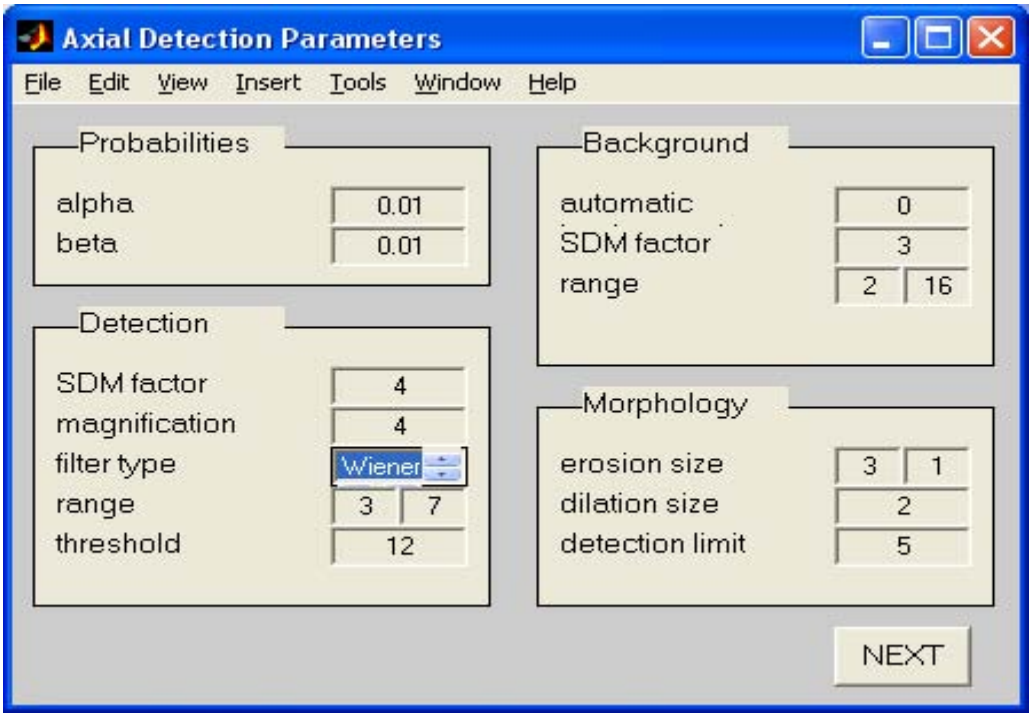

Figure 8-1
Default Detection Parameters for Axial Scan

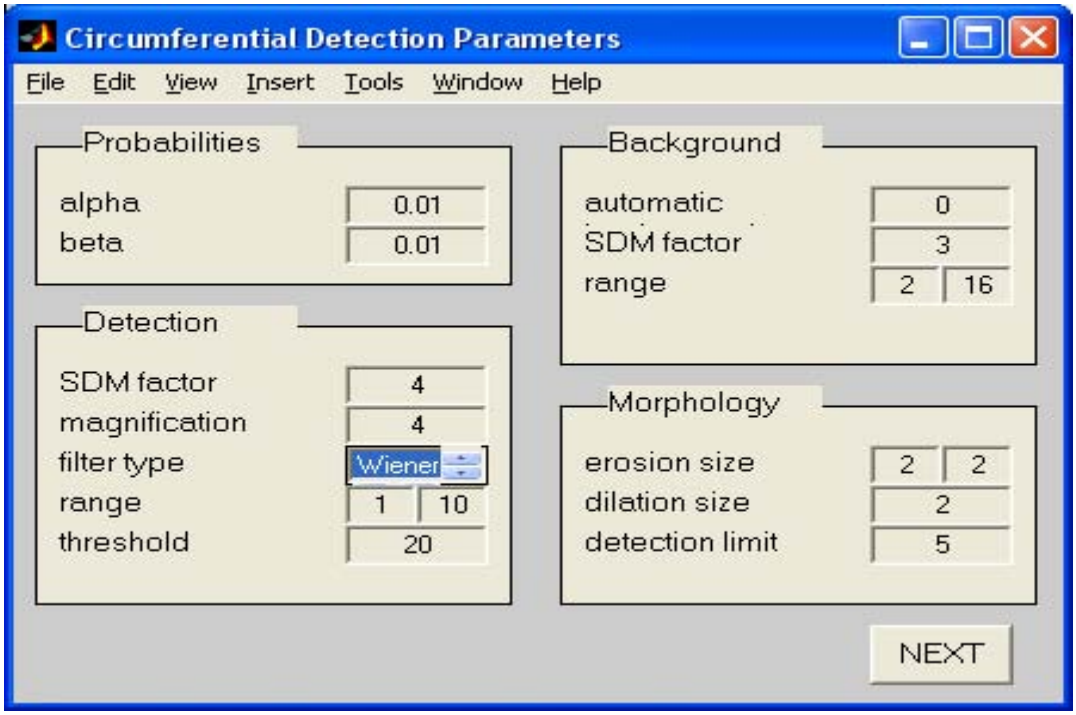

Figure 8-2
Default Detection Parameters for Circumferential Scan

Initially, two blind tests were performed for the original version of ANATRES with the USNRC mock-up data.. Test 1 includes 25 mock-up tubes with axial OD SCCs in the TSP region. Test 2 includes 25 miscellaneous SCCs. Each mock-up tube consists of nine segments, which means that the total number of evaluated segments is 225 for each test.

After the initial tests were performed, the modified version was then tested using the Examination Technique Specification Sheets (ETSS) dataset and limited field data.

The notation used for performance evaluation is given in Table 8-1. Positive decision is *defect*, negative decision is *non-defect*. Decisions are made for each segment.

Table 8-1
Table of Test Outcomes

Notation	Explanation
TP	true positive (correct *defect* decision)
TN	true negative (correct *non-defect* decision)
FP	false positive (incorrect *defect* decision)
FN	false negative (incorrect *non-defect* decision)
$FNR = \frac{FN}{TP + FN}$	false negative rate (missed alarm rate)
$FPR = \frac{FP}{TN + FP}$	false positive rate (false alarm rate)

8.1 Original Version Test 1

Initial detection results output by the detection module are presented in Table 8-2. For this data set, all defects were found with the default parameters, but the number of false alarms was 21, which means that the false alarm rate was 11.4%. Further analysis of initial indications using the characterization algorithm eliminated 16 false and 2 true calls, leading to the final false alarm rate of 2.7% and the final missed alarm rate of 5%. Detailed results are given in Table 8-3.

Table 8-2
Initial Detection Results for Test 1

Tube Number	TP	TN	FP	FN
1	2	6	1	0
2	2	7	0	0
3	3	6	0	0
4	2	6	1	0
5	1	8	0	0
6	2	6	1	0
7	1	6	2	0
8	1	7	1	0
9	2	6	1	0
10	1	8	0	0
11	2	6	1	0
12	1	7	1	0
13	1	7	1	0
14	2	6	1	0
15	2	5	2	0
16	1	6	2	0
17	1	6	2	0
18	1	7	1	0
19	2	7	0	0
20	2	7	0	0
21	2	6	1	0
22	4	5	0	0
23	1	8	0	0
24	1	7	1	0
25	1	7	1	0
Total	**41**	**163**	**21**	**0**

Table 8-3
Final Detection Results for Test 1

Tube Number	TP	TN	FP	FN
1	2	7	0	0
2	2	7	0	0
3	3	6	0	0
4	2	7	0	0
5	1	8	0	0
6	2	7	0	0
7	1	6	2	0
8	1	8	0	0
9	2	7	0	0
10	1	8	0	0
11	2	7	0	0
12	1	8	0	0
13	1	8	0	0
14	2	7	0	0
15	2	7	0	0
16	1	7	1	0
17	0	7	1	1
18	1	8	0	0
19	2	7	0	0
20	2	7	0	0
21	2	7	0	0
22	4	5	0	0
23	1	8	0	0
24	1	7	1	0
25	1	8	0	0
Total	**39**	**179**	**5**	**2**

The probability of detection (POD) curve for this test is given in Figure 8-3. Since the number of samples is small, the estimated probability values have large uncertainties. This is visible from wide confidence limits, which are especially large when the number of samples per bin is small.

The estimated probabilities presented in Figure 8-3 imply that the axial OD SCCs near TSP larger than 40% TW can be detected with the probability near 60% or larger in 90% of the detection tests, but a similar defect larger than 90% TW can be detected with probability larger than 85% in 90% of the detection tests. An anomaly in the region from 60%-80% has occurred due to a missed crack in a dent. Probability of detection for cracks less than 40% TW is estimated to be larger than 50% in 90% of the tests.

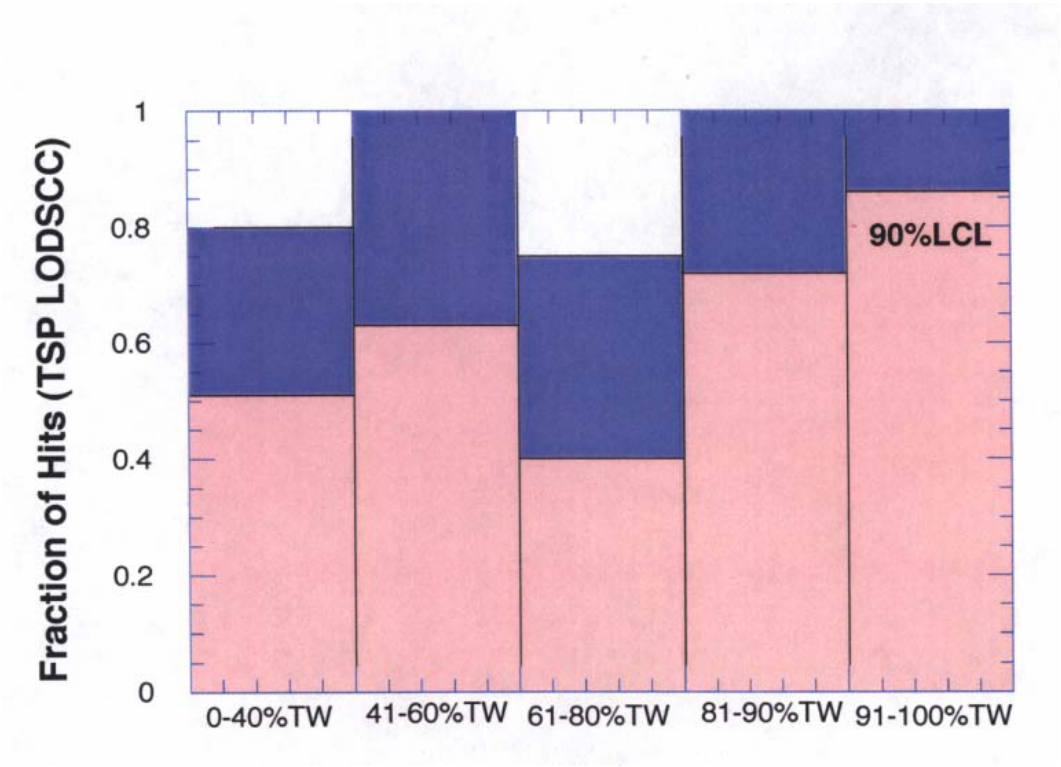

Figure 8-3
Probability of Detection for Test 2

8.2 Original Version Test 2

Initial detection results with default parameters are presented in Table 8-4. For this data set, 4 defects were missed and the number of false alarms was 27. Therefore, the missed alarm rate was 10% and the false alarm rate was 14.5%. Further analysis of initial indications using the characterization algorithm eliminated all but 4 false indications, but it also removed 3 indications which were true defects. Detailed results are given in Table 8-5. The final false alarm rate is only 2%, but the final missed alarm rate is 17.5%. This result indicates that the characterization rules are not properly optimized to balance false and missed alarm rates. Although the missed alarm rate is large, most missed alarms are shallow, which is evident from the shape of the estimated POD curve, given in Figure 8-4.

Table 8-4
Initial Detection Results for Test 2

Tube Number	TP	TN	FP	FN
1	1	7	1	0
2	1	8	0	0
3	2	4	3	0
4	2	6	1	0
5	1	7	1	0
6	3	6	0	0
7	1	6	2	0
8	2	6	1	0
9	2	6	1	0
10	0	8	0	1
11	2	5	1	1
12	1	6	2	0
13	2	6	1	0
14	1	7	0	1
15	1	7	1	0
16	1	5	3	0
17	1	5	3	0
18	2	6	1	0
19	1	6	2	0
20	1	6	1	1
21	1	7	1	0
22	1	8	0	0
23	1	8	0	0
24	3	5	1	0
25	2	7	0	0
Total	**36**	**158**	**27**	**4**

Table 8-5
Final Detection Results for Test 2

Tube Number	TP	TN	FP	FN
1	1	8	0	0
2	1	8	0	0
3	2	7	0	0
4	2	7	0	0
5	1	8	0	0
6	3	6	0	0
7	1	7	1	0
8	1	7	0	1
9	2	6	1	1
10	0	8	0	1
11	2	6	0	1
12	1	7	1	0
13	1	7	0	1
14	1	7	0	1
15	1	8	0	0
16	1	7	1	0
17	1	8	0	0
18	1	7	0	1
19	1	8	0	0
20	1	7	0	1
21	1	8	0	0
22	1	8	0	0
23	1	8	0	0
24	3	6	0	0
25	2	7	0	0
Total	**33**	**181**	**4**	**7**

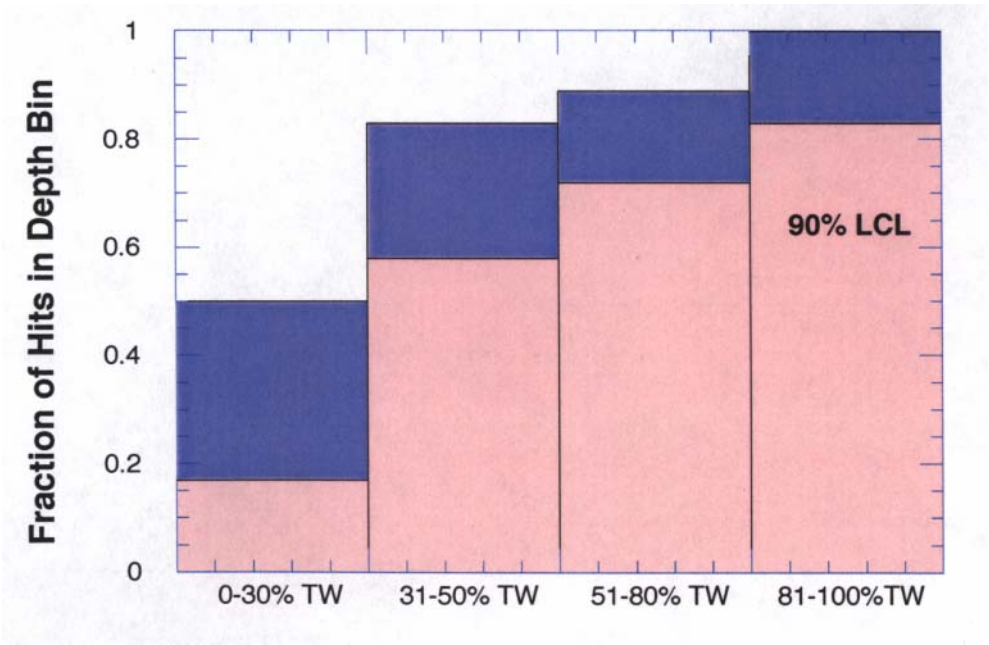

Figure 8-4
Probability of Detection for Test 2

The POD curve indicates that the cracks larger than 30% TW can be detected with the probability near 60% or larger in 90% of the tests. Probability of detection for cracks less than 30% TW appears to be very small.

8.3 Original Version Post Test Examination

The conclusions of the final defect analysis for the original version are that two defects were missed in Test 1 and seven defects in Test 2. One of these defects was, mistakenly, included in both data sets. A detailed description of the eight missed defects is presented here in order to provide for possible algorithm modifications to enhance the performance.

Figure 8-5 shows the surface plot of the X probe array signal around the region of a missed axial ID 25% TW defect. The filtered defect signal obtained after detection with the default threshold and final background subtraction is also presented. The presence of some defect signals can be observed. Smaller indications are discarded as non-defect. Analysis of the largest indication is presented in Figure 8-6. Phase angles are measured in the counter-clockwise direction and computed relative to a reference angle of 150 degrees.

Since the flaw type (OD vs. ID) is not known *a priori*, characterization is performed for both cases and inspection of the plot gives indication about the possible type. The phase rotation with frequency increase is generally a good indicator of the flaw type. The difference in the estimation under the two assumptions is due to the difference in the reference frequency at which the peak is estimated for ID and OD flaws. Calibration curves also depend on the defect type, as described in Chapter 4.

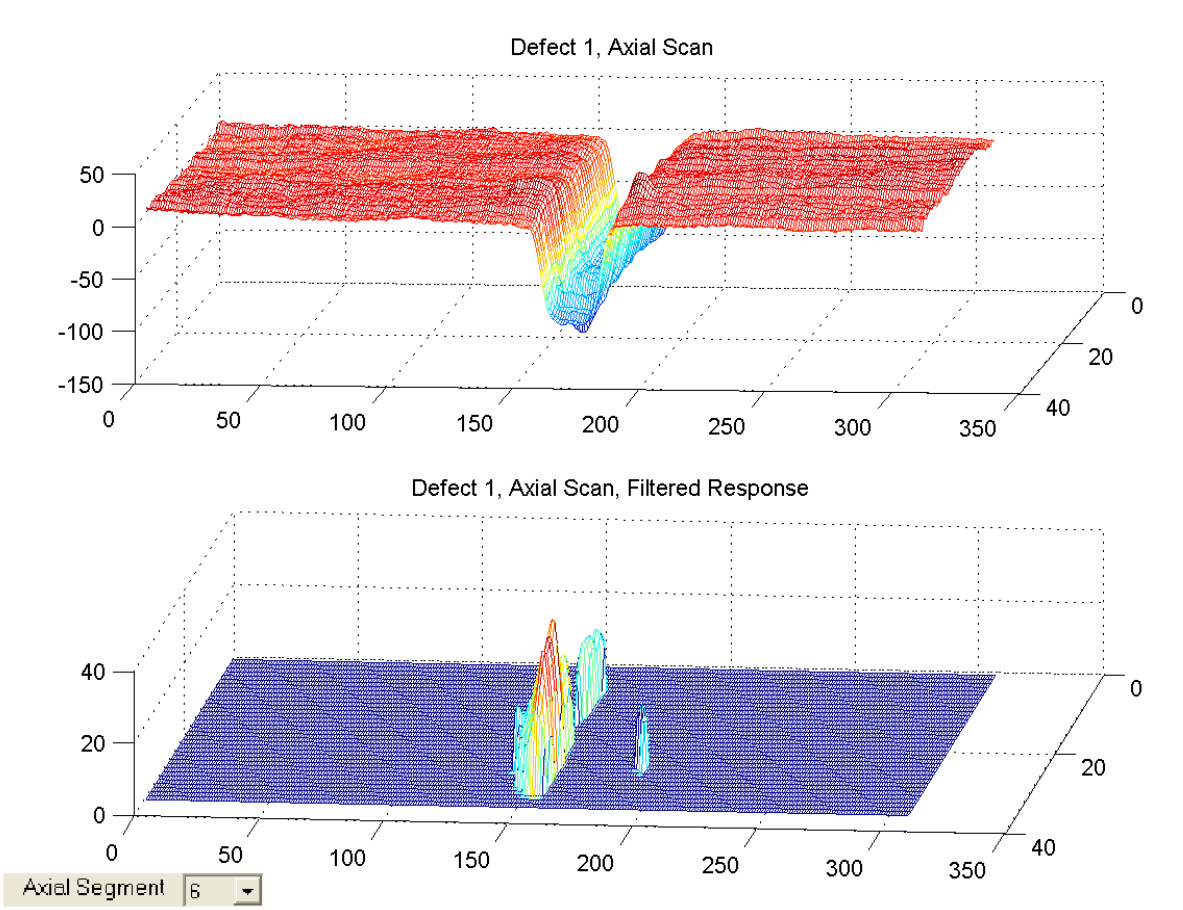

Figure 8-5
Defect 1 Position

Small positive phase angle and phase increase with frequency visible in Figure 8-6 indicate a small ID flaw. However, the estimated defect depth is smaller than 5%, which is considered as the absolute limit for defect detection, and the signal is classified as a false call.

Defect 2 is an axial ID 25% TW defect near a dent. The expected position of this defect is in the TSP region in Figure 8-7. No defects were found in this region with the default threshold. The threshold has been reduced to a level that triggers multiple false alarms, but the sharp peak in the TSP region is the result of incorrect TSP removal, not a flaw signal.

Defect 3 is an IGA 55% TW and is correctly located, as presented in Figure 8-8.

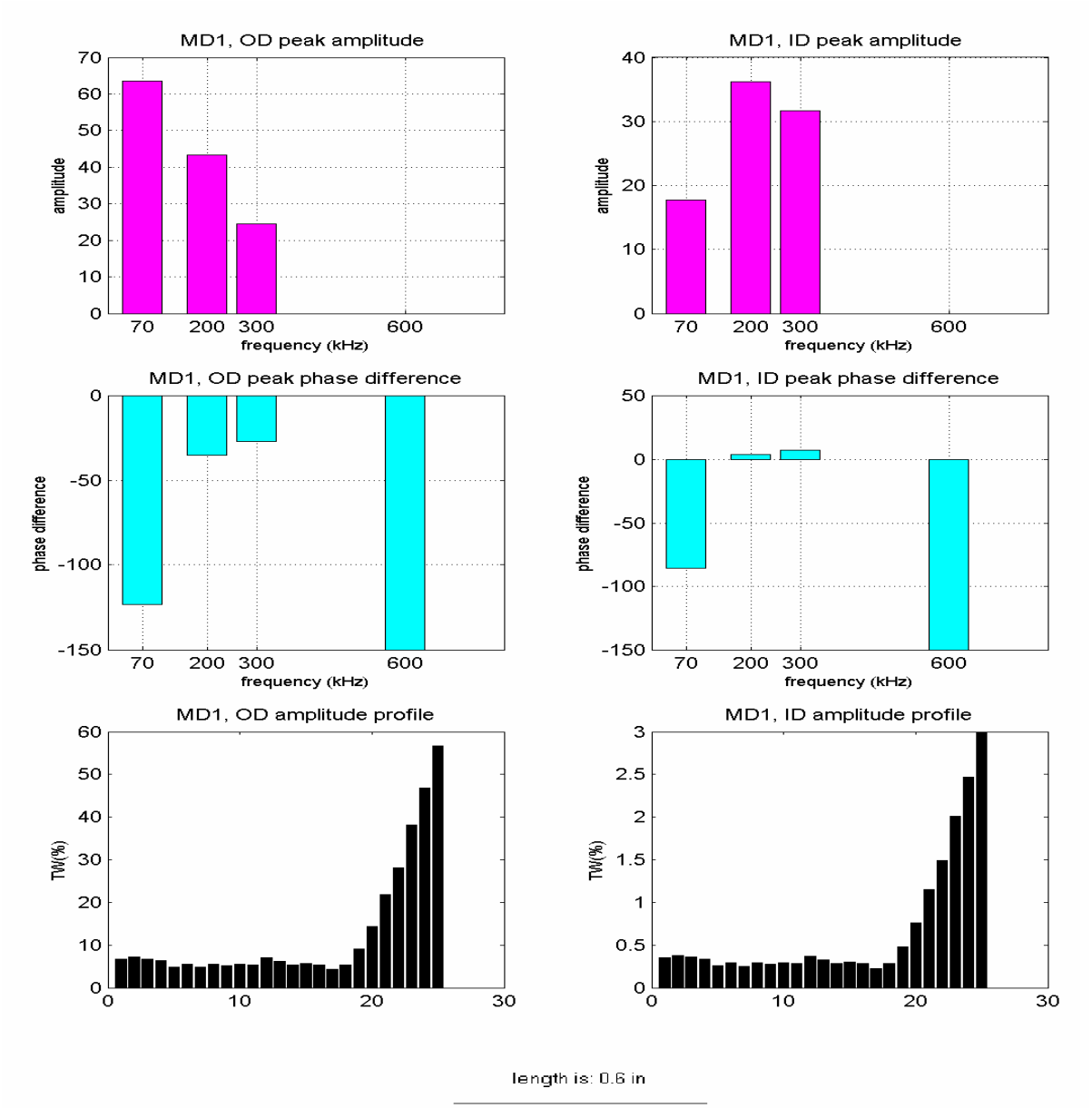

Figure 8-6
Defect 1 Characterization

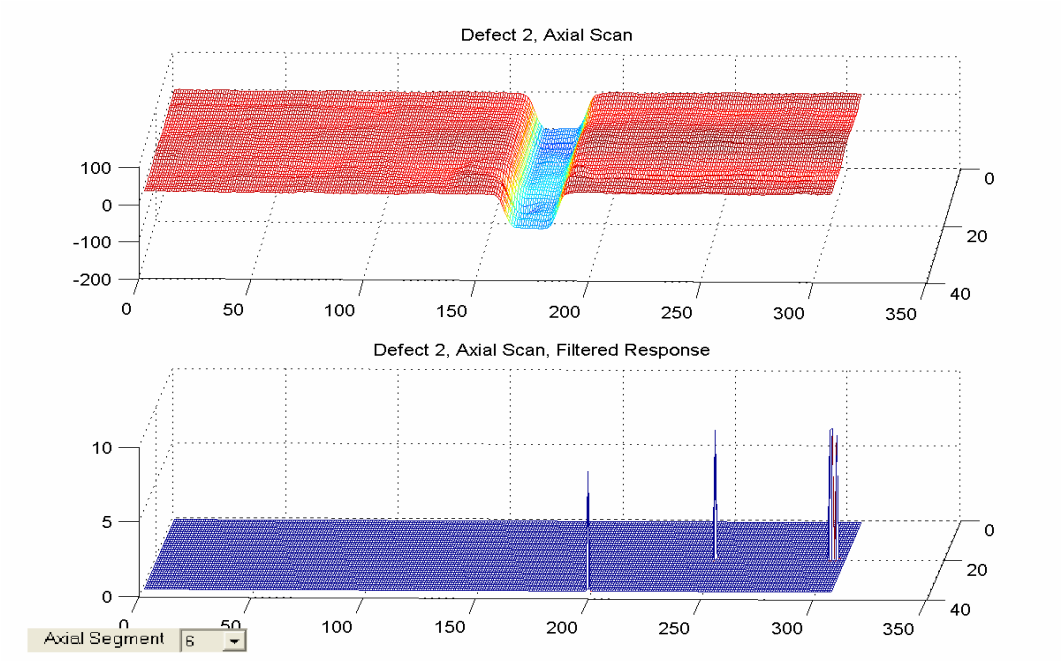

Figure 8-7
Defect 2 Position

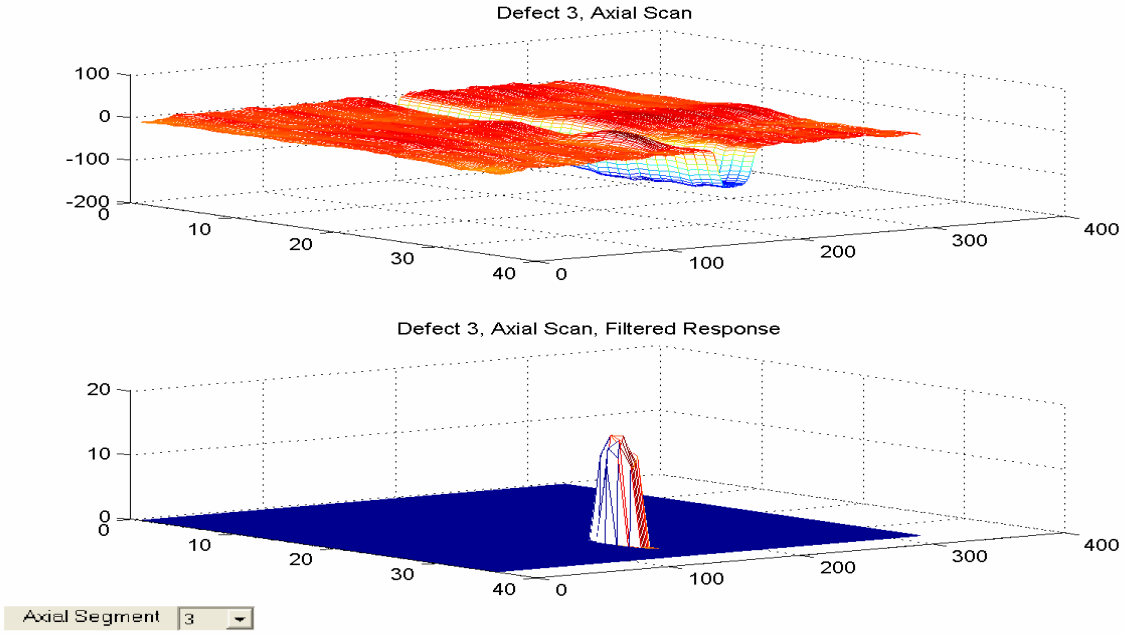

Figure 8-8
Defect 3 Position

Although the defect signal is evident in the filtered image, characterization does not indicate defect, as presented in Figure 8-9. The problem here is probably due to incorrect final filtering for small defects. Peak analysis at 200 kHz indicates only one nonzero component. While peak analysis at 300 kHz indicates two components, it uses only one of these components for size estimation. The direction of phase change is characteristic for an OD defect. The implemented heuristic rules incorrectly eliminated this defect. The rule can be modified to include a defect that is visible at only one frequency and gives an estimate larger than 10%, but that could increase the false alarm rate. Nevertheless, this modification will be considered in the future testing.

Defect 4 is an axial ID 20% TW defect near a dent. It has not been detected with the standard threshold, but a slight threshold reduction to 12 shows a defect signal, as presented in Figure 8-10. Characterization, however, indicates an OD flaw, as presented in Figure 8-11. The problem here is probably due to an incorrect background estimation and subsequent subtraction.

Defect 5 is an ID 50% TW defect at roll transition. Several indications in this region are visible in Figure 8-12. However, all of them are classified as non-defects by the characterization algorithm. Since a periodic artifact component is visible in this case, pre-filtering could improve characterization.

Defect 6 is a shallow axial OD defect. Figure 8-13 shows the tube segment with the defect. The vertical component of the signal in this region is below the default threshold value. When the threshold value is reduced to 5, some indications could be found. However, such a low threshold is not acceptable, since the false alarm rate would be too large.

Defect 7 is a circumferential ID 35% TW defect in the transition region presented in Figure 8-14. The signal-to-noise ratio in this region is low and it is difficult to discriminate small defects from artifacts. The indication presented in the filtered image is found with the threshold equal to half of the default value, but is classified as non-defect.

Defect 8 is an axial 70% ID defect in dent. However, the vertical signal is small, as shown in Figure 8-15. Overly conservative artifact (dent) removal leads to a single non-zero component and the defect is eliminated.

Overall, the main difficulty in detection and classification are for shallow defects, or the defects in artifacts, where the signal to noise ratio is small. Notable are poor detection for defects in dents and at roll transitions. Some improvement could be accomplished using better characterization rules or filtering. However, in some cases the probe signal is too small to be detected by an automatic algorithm with a reasonable false alarm rate.

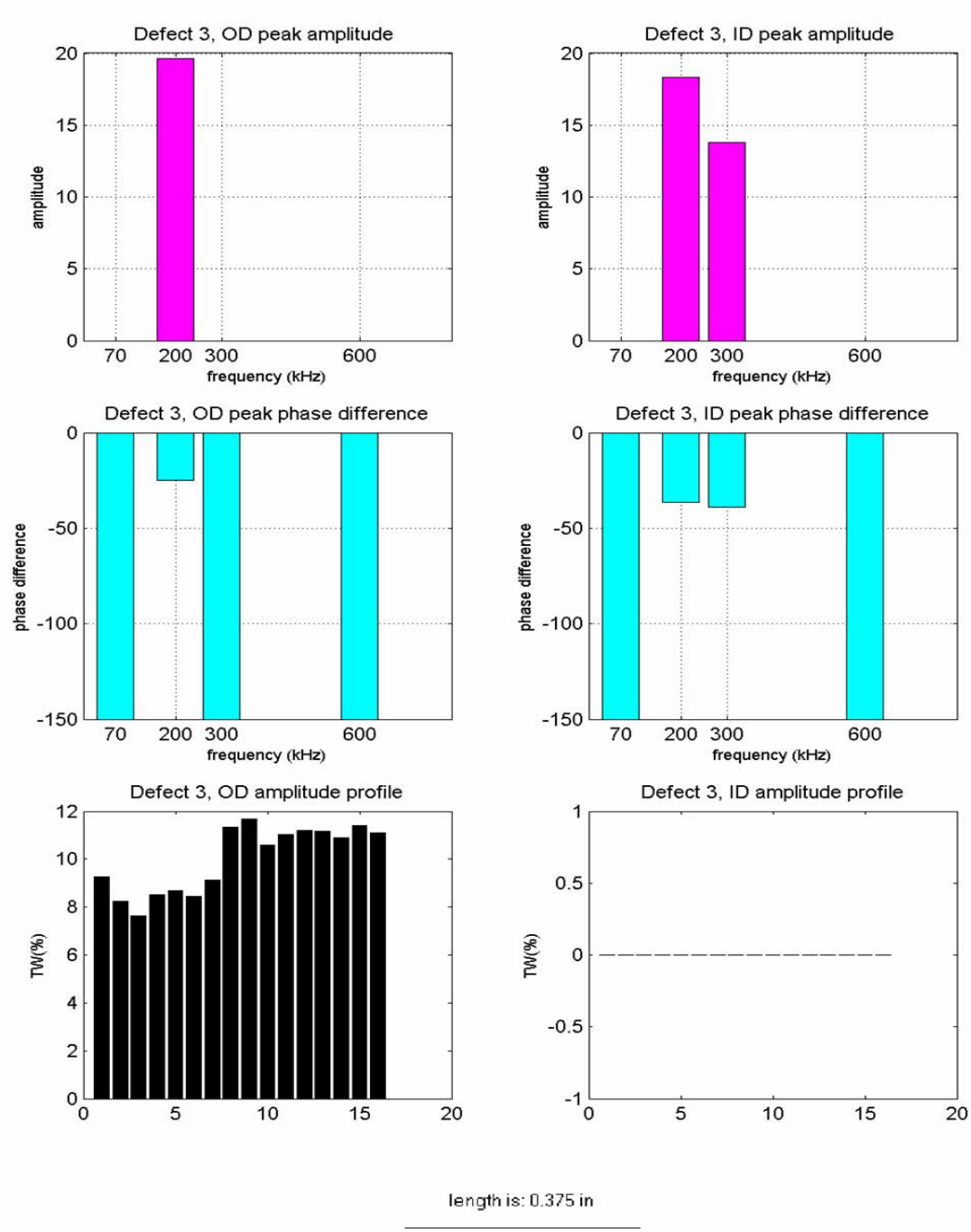

Figure 8-9
Defect 3 Characterization

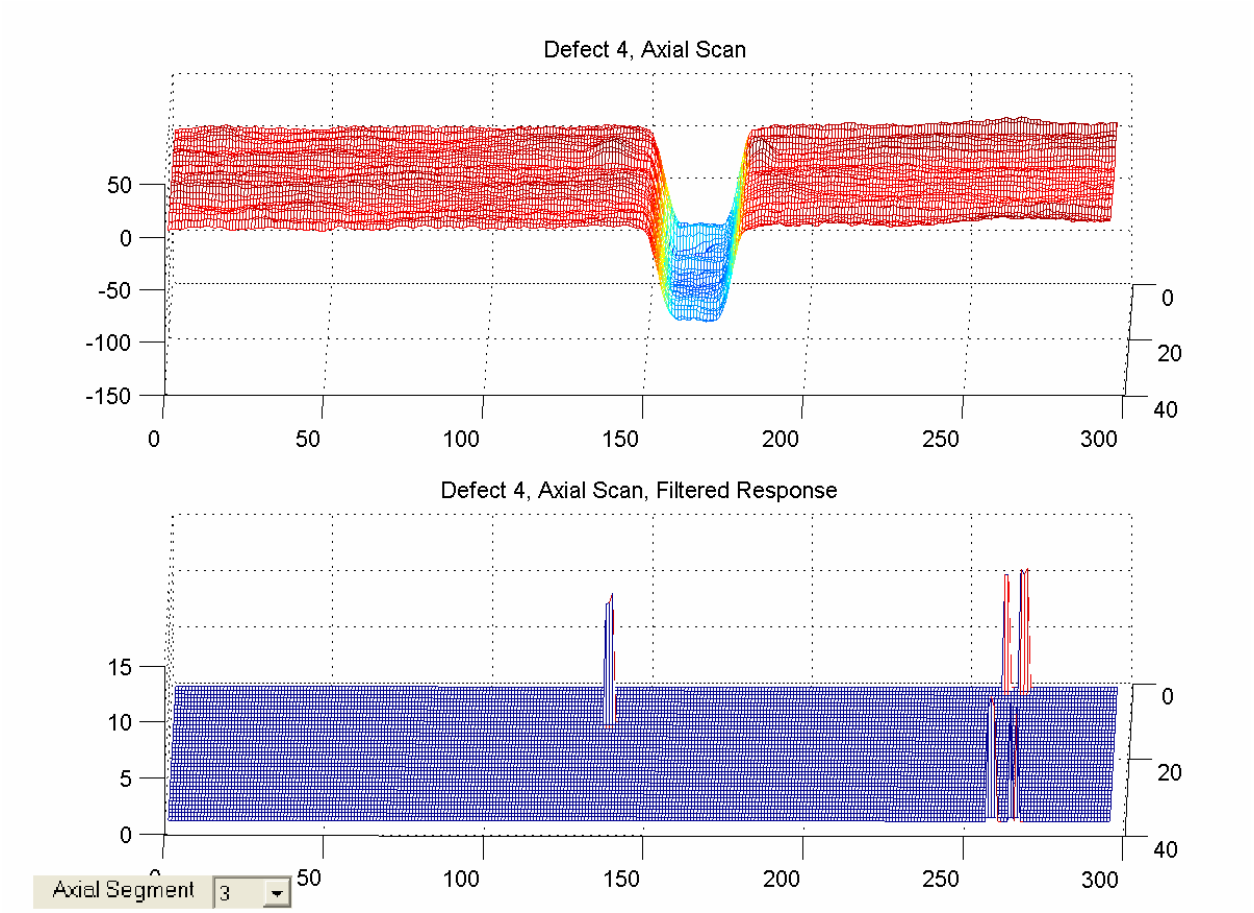

Figure 8-10
Defect 4 Position

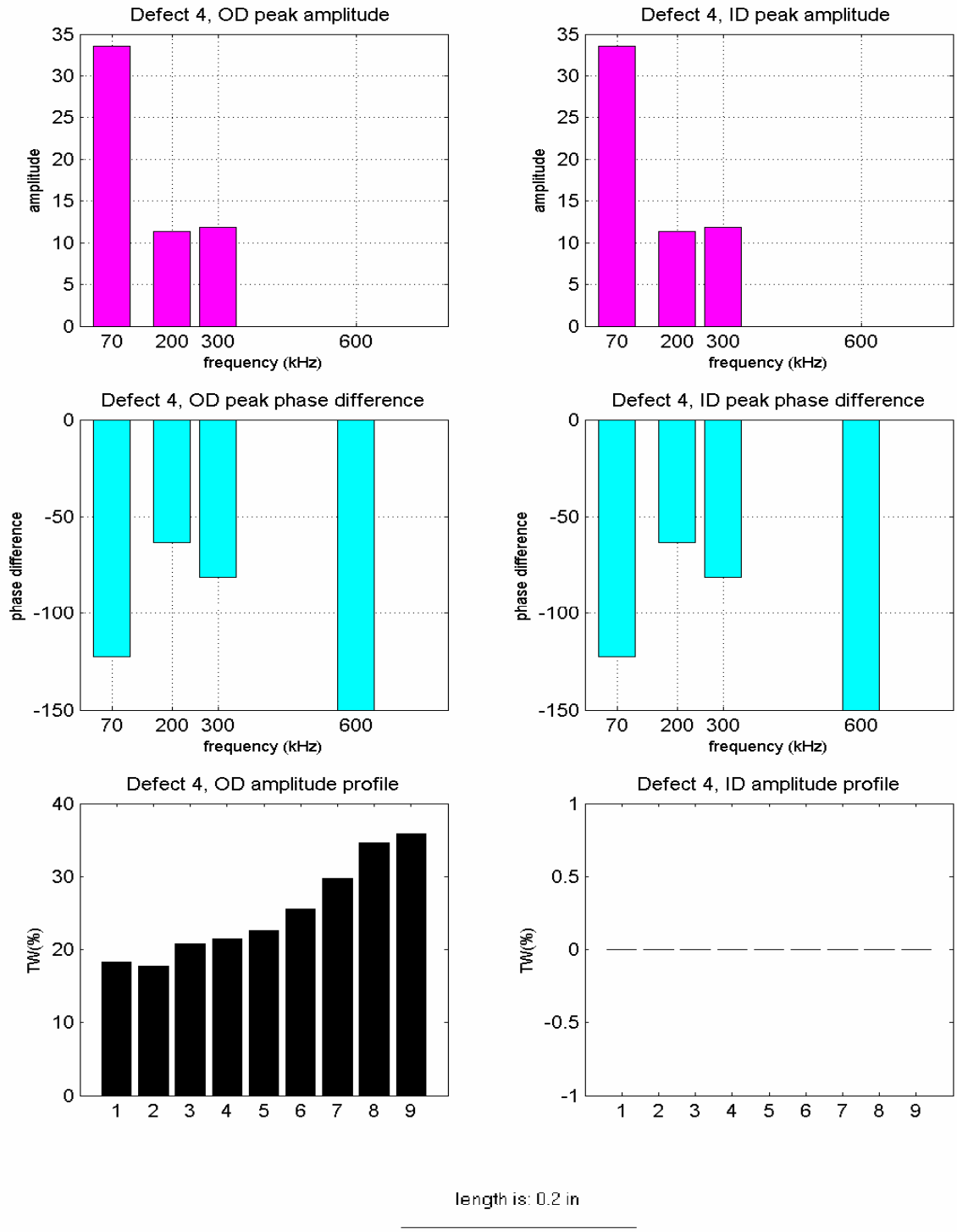

Figure 8-11
Defect 4 Characterization

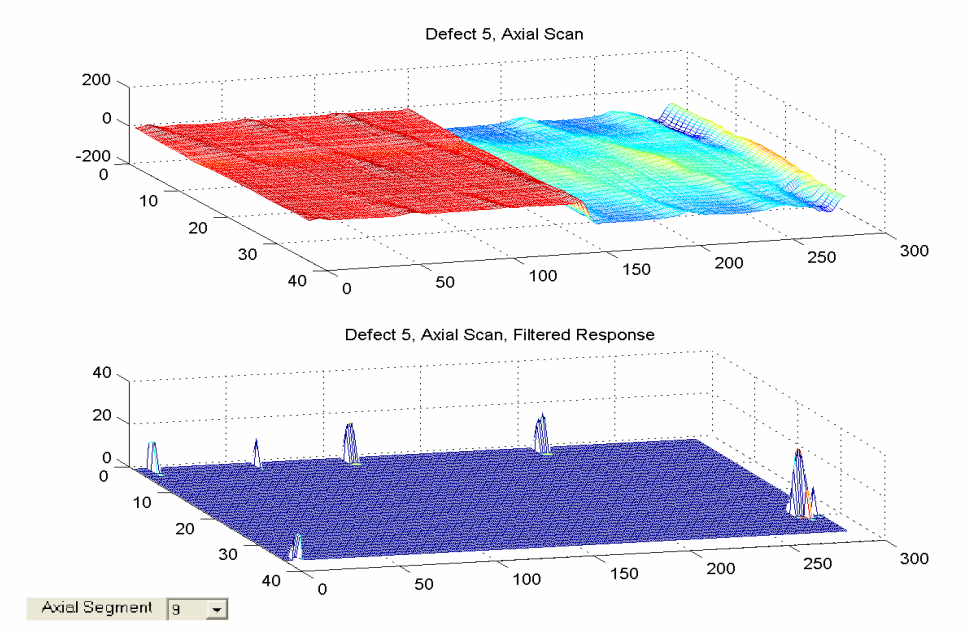

Figure 8-12
Defect 5 Position

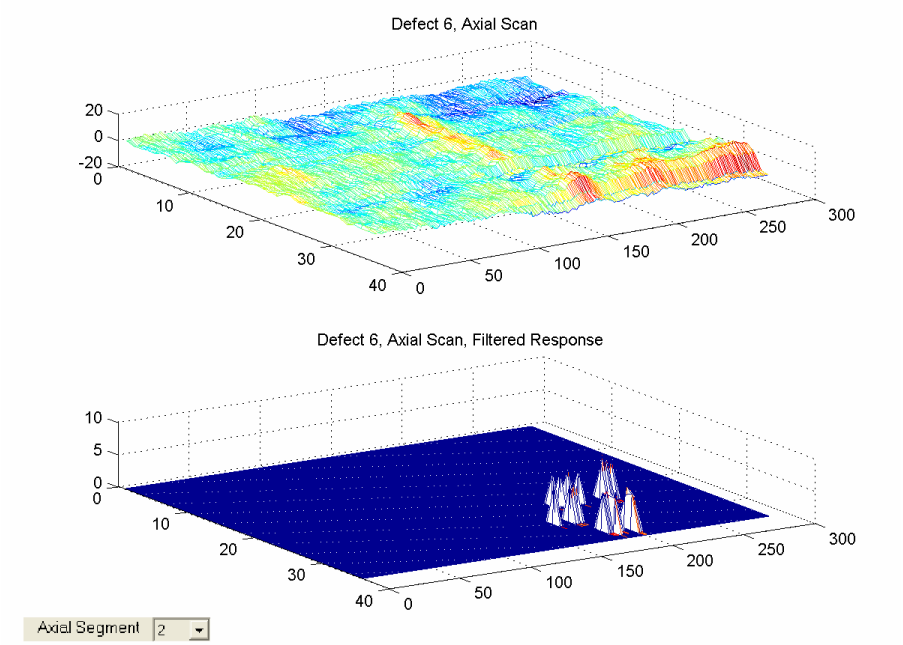

Figure 8-13
Defect 6 Position

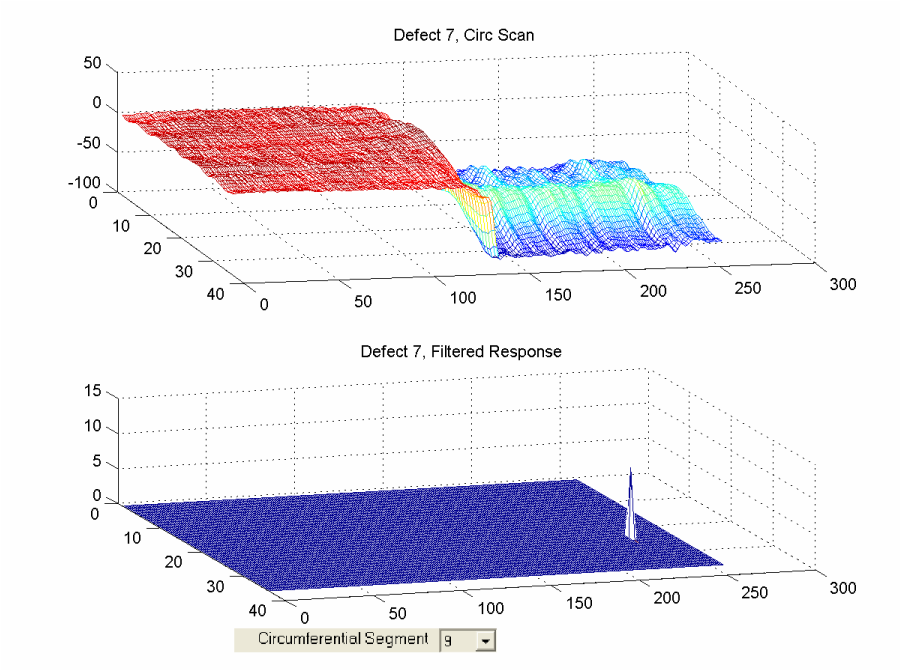

Figure 8-14
Defect 7 Position

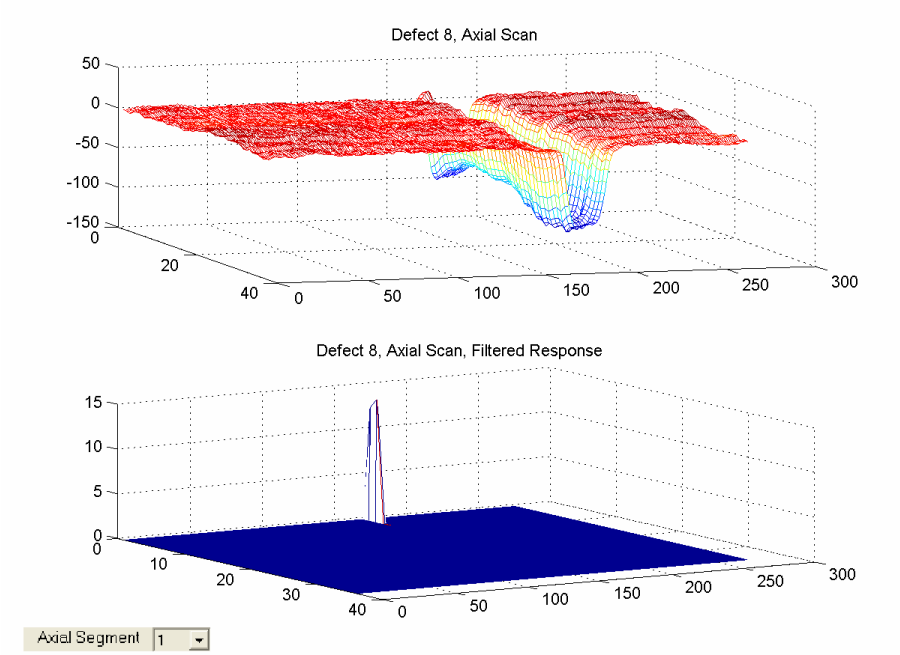

Figure 8-15
Defect 8 Position

8.4 ETSS Data

A database with benchmark defects designated as ETSS (Examination Technique Specification Sheets) was received from EPRI to test the performance of the modified calibration curves. The performance evaluation is presented in Table 8-6. Only one defect was missed with the default ANTARES settings. Standard depth estimation rules implemented in the original ANATRES characterization algorithm resulted in relatively large errors. In the standard implementation, defect type (ID or OD) is determined based on the amplitude and phase trends and the estimated maximum depth corresponds to that type. However, some ambiguity in the discrimination between large OD and ID flaws has been observed, which degraded the depth estimation. For the ETSS data set it was found that better estimation accuracy is obtained when the maximum depth is selected regardless of the declared type of the defect. The results presented in Table 8-6 are encouraging, but might be caused by generally large defects and high SNR, which is rarely the case in field inspection. The problems encountered in OD and ID discrimination for the modified calibration curves led to the decision to report the final classification results in the automated classification algorithm in two classes only: “ indication” and “ non-detectable degradation”. Uncertainty in calibration and sizing accuracy was also the reason to abandon maximum depth reporting in the modified automated version of the software.

Table 8-6
ETSS Testing Results for Conservative Depth Estimation

ETSS #	File	Defect Pos.	Type	~Max Depth (%)		~Length (A(mm),C(◦))	
				ANT	Met	ANT	Met
20400.1	dhr024c96b	tube sheet	CODSCC	52	58	80	120
	dhr049c96b	tube sheet		52	47	80	87
20402.1	dhr135c99b	TSP-def A	LODSCC	missed	70	-	12
		TSP-def B		70	73	10	12
		TSP-def C		70	86	10	15
		TSP-def D		100	100	13	15
20500.1	dhr032c96a	expansion	CPWSCC	100	83	57	45
	dhr033c96a	expansion		100	100	68	100
20501.1	dhr036c98a	exp-def2	LPWSCC	57	67	9	5
		exp-def5		100	94	9	20
		exp-def8		75	72	7	5
20502.1	dhr033c97	TSP,dent,d1A	LPWSCC	70	77	8	10
		TSP,dent,d1B		55	63	8	5
		TSP,dent,d2		100	100	17	12
		TSP,dent,d3		80	87	17	10

8.5 U-Bend Data

Modifications introduced in ANTARES are tested on the X-probe™ data from U-bend regions of 10 Diablo Canyon steam generator tubes received from EPRI. Since ANTARES requires initial parameter adjustment, this data set served both for training and for testing. Reported +point indications have been used as a guidance about possible defects. However, the sensitivity and resolution of the +point probe and the X-probe™ are different and it is hard to delineate algorithmic performance and probe response. At this point, the default detection parameters have been selected to enable detection of the largest indications visible in the filtered X-probe™ data in the tube designated in the data set as sg24r5c62. At the same time, the attempt was made to control the false calls by selecting the threshold which will produce similar number of defects in ANTARES for tubes with a small number of +point indications. With additional information, it is possible to further optimize detection parameters and filters, as well as characterization rules.

Default detection parameters for the U-bend region that have been modified with respect to the default parameters for mock-up data are listed in Table 8-7.

Table 8-7
Modifications in the Default Detection Settings

	Axial Scan		**Circumferential Scan**	
	Mock-up	U-bend	Mock-up	U-bend
Background range	(2,16)*	(2,12)	(2,16)	(2,10)
Detection threshold	12	50	20	20

* the first number represents the number of pixels in the axial direction, the second number in the circumferential direction

Modifications in the background range are the result of the decrease of the number of points along the circumferential direction from 36 to 32. Threshold modification is due to modifications in the signal-to-noise ratio introduced during filtering. The axial threshold is unreliable, since only one axial indication is reported by +point probe in the training data set. ANTARES has found a number of axial defects, but some of them have a corresponding circumferential component.

Figure 8-16 presents the raw data for an axial flaw and the result of application of the axial median filter with a sliding window of 31 pixels.

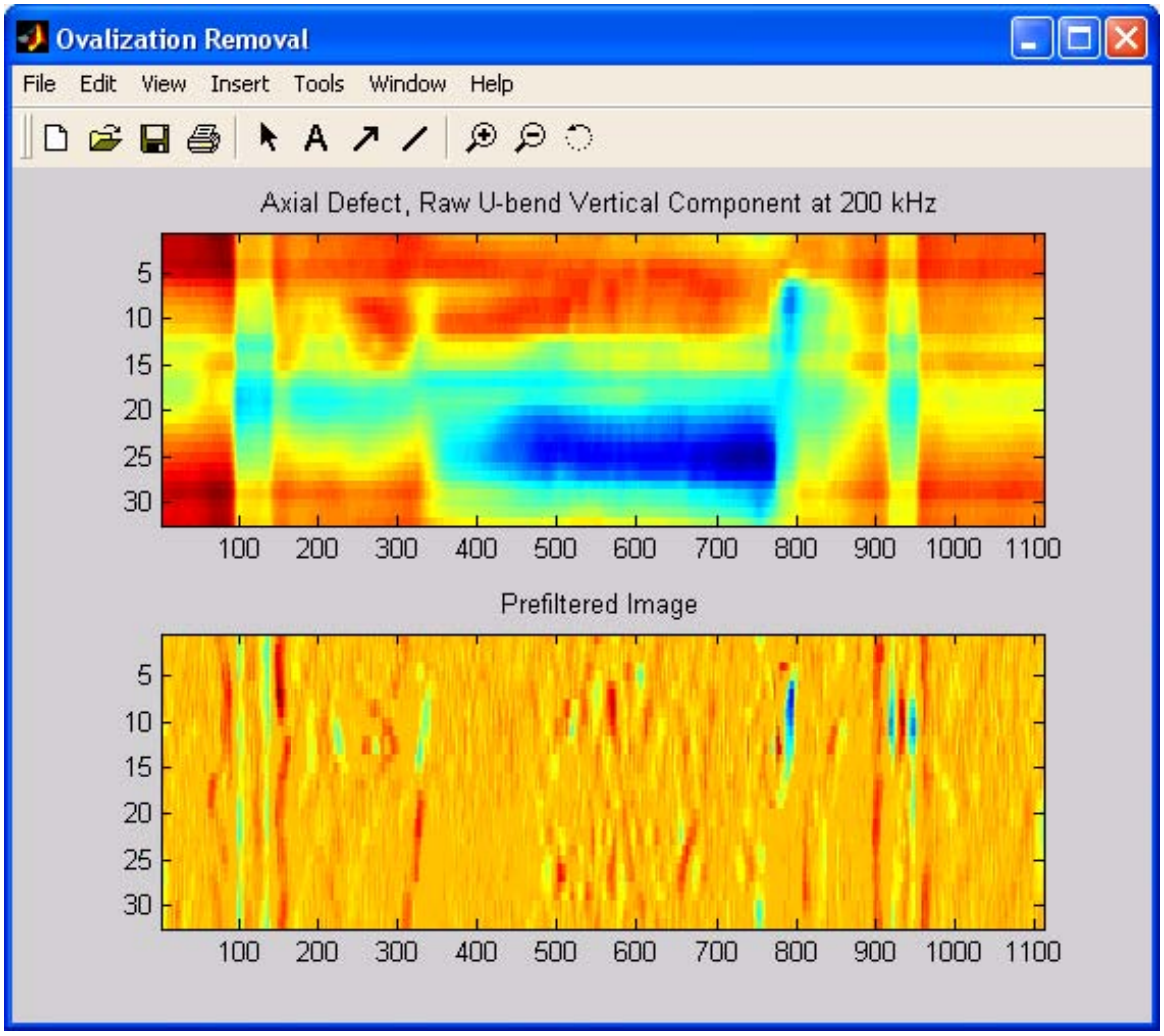

Figure 8-16
Ovalization Removal for an Axial Defect in Preprocessing Stage

Although the filtered image still has high noise, two indications only are produced by the ANTARES detection algorithm, as presented in Figure 8-17. One of these defects is further eliminated using characterization rules.

Additional circumferential filtering is useful for circumferential scans. Figure 8-18 presents the raw and filtered data for a circumferential defect. The circumferential median filter has a window of 11 pixels. Detection results are presented in Figure 8-19. One indication is eliminated during characterization.

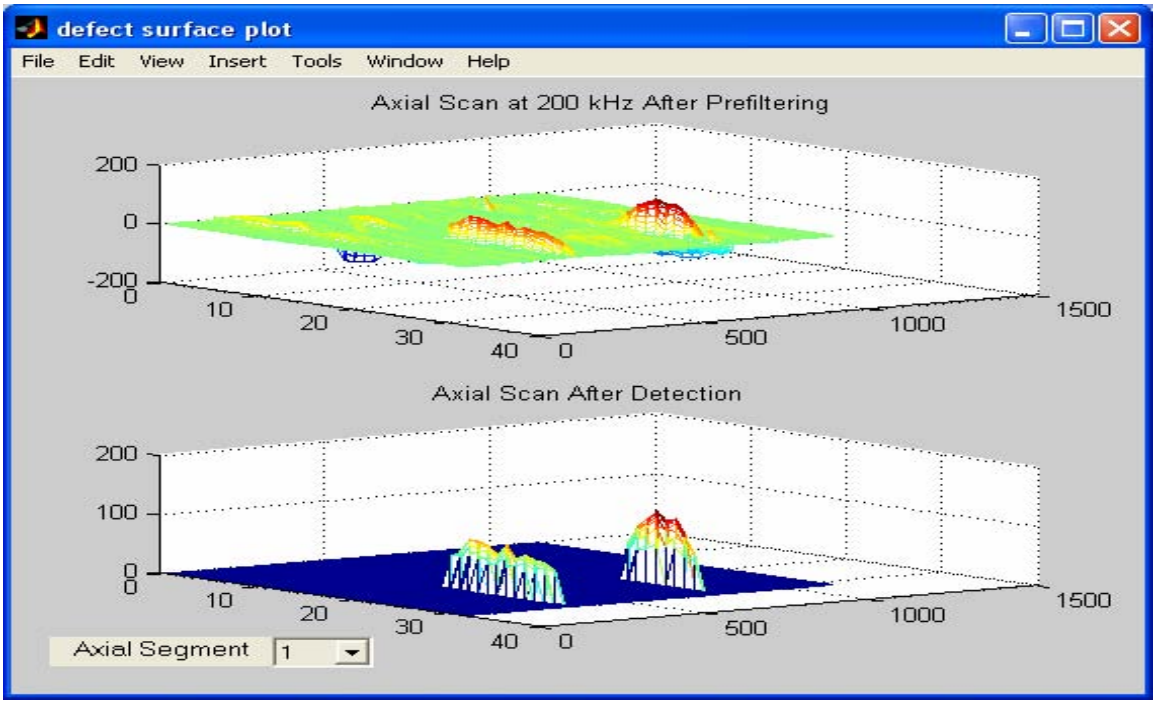

Figure 8-17
Detection Stage Showing an Axial Defect and a False Alarm Eliminated in Characterization Stage

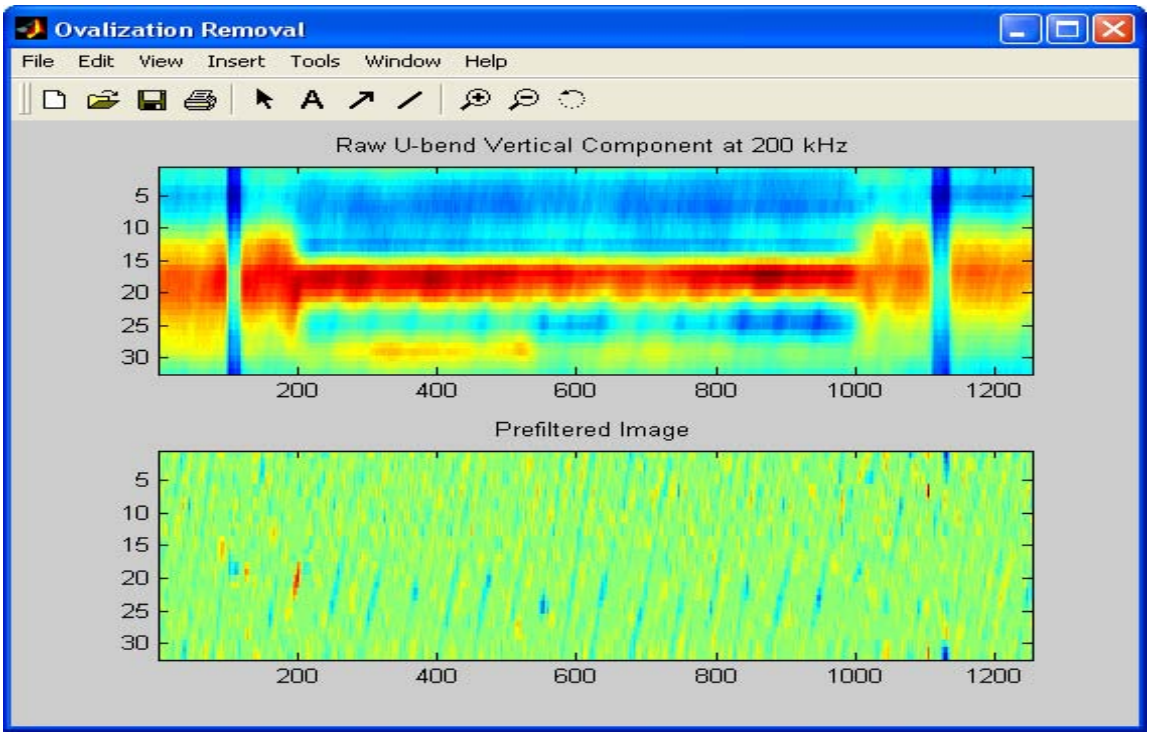

Figure 8-18
Ovalization Removal for a Circumferential Defect in Preprocessing Stage

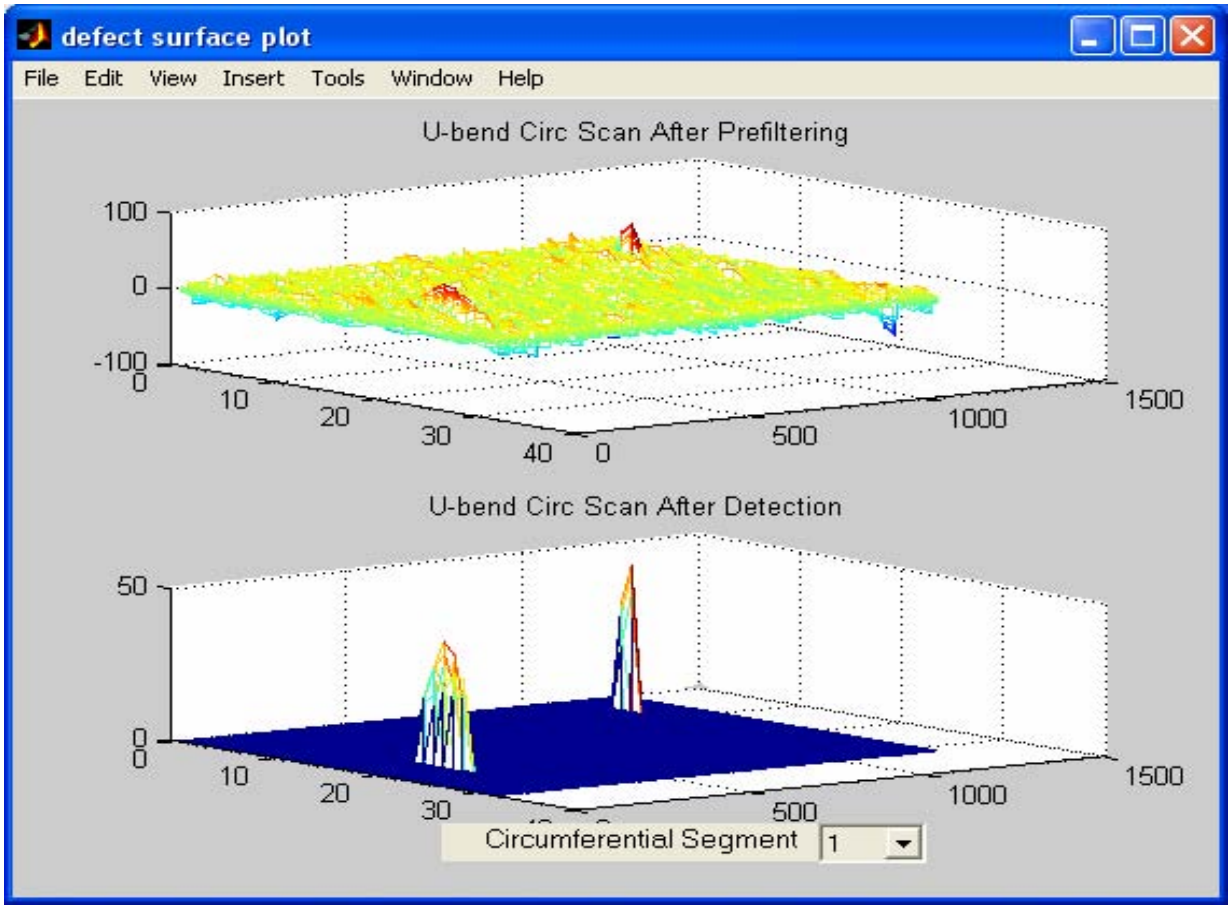

Figure 8-19
Detection Stage Showing a Circumferential Defect and a False Alarm Eliminated in Characterization Stage

Preliminary results for 10 tubes from the Diablo Canyon U-bend are summarized in Table 8-8. Number of indications is the result obtained by the detection algorithm. Empirical rules in the characterization algorithm eliminate some of these indications as false defects.

According to the +point inspection, most defects are ID. However, a number of defects have been characterized as OD using modified ANTARES calibration. Just like the results for ETSS data, discrimination between OD and ID defects was difficult in numerous cases and somewhat arbitrary. For that reason, we presented only the total number of defects of both types. In addition, the fact that the same defect could be found both in axial and in circumferential scans is not taken into account in Table 8-8.

Table 8-8
Preliminary Results for U-Bend Region

SG	Row	Col	After Detection		After Characterization		Number of +Point Indications
			Number of Indications		Number of Axial Defects	Number of Circ Defects	
			Axial	Circ			
21	1	24	2	3	1	0	1
21	1	43	5	5	1	1	1
21	5	54	0	6	0	3	7
22	4	51	3	11	0	4	21
23	3	86	6	12	3	5	2
23	3	93	3	5	1	2	1
24	5	62	6	40	5	23	35
24	5	68	1	4	0	2	5
24	6	23	0	7	0	3	5
24	6	53	0	2	0	1	1

Comparison with the +point probe results indicated some significant differences. For example, the number of defects found in ANTARES for tubes sg21r5c54 and sg22r4c51 is much smaller than the number of +point probe indications. This is due either to filtering or to the X-probe™ response, since a smaller number of peaks is visible in the filtered data than in the corresponding +point plots. On the contrary, significantly larger number of indications and defects is found by ANTARES for tube sg23r3c86. This is due to the very noisy data after preliminary filtering and indicates that additional filtering is needed. Finally, the results for tube sg24r5c62 show a large number of indications, but many of them are eliminated during characterization. In general, the initial number of indications after the detection phase is much larger for the U-bend data than for the mock-up data. This is an issue for the original version of ANTARES, where each indication is characterized individually. This was additional incentive to implement automated ROI extraction.

Differences between ANTARES results and +point probe indications could be attributed to some extent to differences in spatial resolution, probe speed, and signal-to-noise ratio of X-probe™ and +point probe, as well as general technique limitations. There is also a possibility of +point probe overcalls.

The results presented in Table 8-8 were obtained using manual cropping of defects and visual inspection of phase and amplitude trends. Testing with a completely automated procedure developed for implementation in the EPRI AutoAnalysis software was repeated for a subset of the U-bend data, downloaded from the EPRI AutoAnalysis web site and recommended as an array probe testing data set. The raw data used in the testing presented in Table 8-8 was calibrated in EddyView™, while the downloaded raw data was calibrated using a MATLAB calibration software provided by Michigan State University [35]. The detection threshold has been changed to accommodate differences in calibration. The results are presented in Table 8-9.

Table 8-9
The Number of Indications for X-Probe Testing Data

Data	ANL Algorithms				+Point Indications
	Axial		**Circumferential**		
	After Detection	**After Classification**	**After Detection**	**After Classification**	
DHR005C062I008	2	2	42	21	35
DHR005C068I016	1	1	2	0	5
DHR006C023I017	0	0	5	1	5
DHR006C053I015	0*	0	0*	0	1

* To obtain indications in this tube, the threshold has to be very small, which significantly increases false alarms in other tubes.

The difference in the detection and classification results could be attributed to differences in calibration, but is more likely due to the somewhat decreased performance of the automated ROI extraction and automated classification rules in comparison with visual inspection of detection results by a trained analyst, which was used in the original version of ANTARES.

8.6 Full Tube Length Data

Data for approximately 30 full length tubes from the Palo Verde steam generator was received. However, difficulties were encountered in the initial data analysis, due to the lack of information about the standards used with these data and defect types and positions. To start off the analysis, a tube with large indications, numbered 031110, was selected. Preliminary analysis in EddyView™ has shown three indications of various sizes. These indications were also found in previous inspections with X-probe™ and +point probe. The segmentation threshold was adjusted to obtain 14 TSP segments, 14 free span segments, and U-bend, using the segmentation method explained in Chapter 2. The plot of the first principal component for this tube evaluated for the vertical component of the circumferential scan at 300 kHz is given in Figure 8-20. Slow trend observed in the first principal component is removed using the median filter with a large window. Negative peaks correspond to TSPs, while a large positive peak corresponds to a large indication in this tube. Two smaller negative peaks in the U-bend regions are also visible.

Positions of the seven TSPs at both sides were correctly determined and the remaining region is assigned as the U-bend region. Figure 8-21 presents a typical TSP region. Figure 8-22 presents a U-bend region, while Figure 8-23 presents a free span region. Images are displayed in the standard ANTARES format: horizontal component followed by vertical components, and grouped according to increasing frequency.

All images are displayed after initial background line subtraction. Noise in the free span and ovalization effects are removed using median filters. Localized signals in the TSP region may not be appropriate for this type of preprocessing.

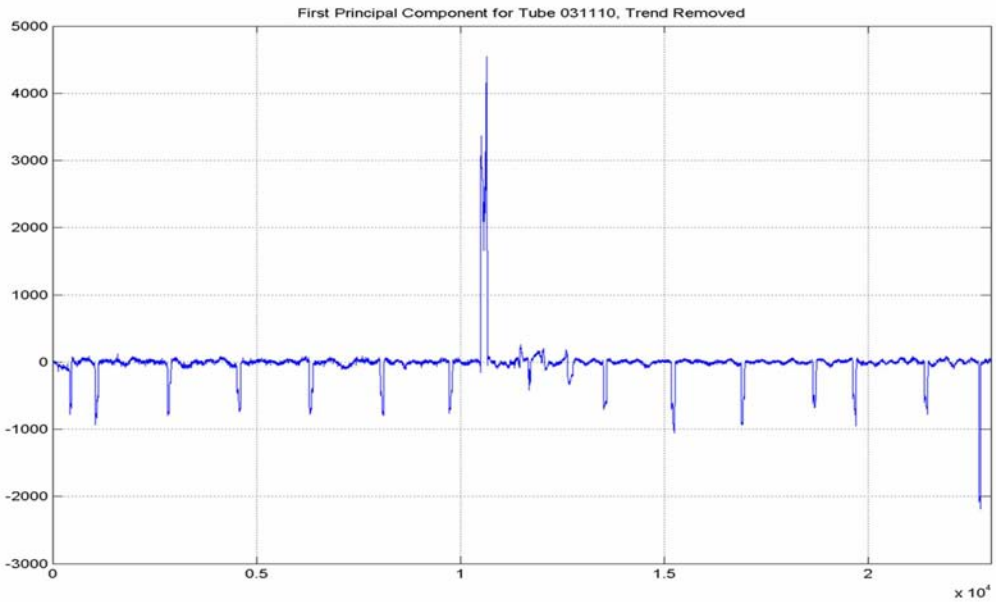

Figure 8-20
First Principal Component Used for Segmentation

We have performed the initial detection analysis of this tube using a large uniform threshold, which resulted in indications in the U-bend region only. Detection results for the axial scan are given in Figure 8-24. Indications at the same positions are observed in the circumferential direction.

The positions of indications are near "bat wings" and the central support, which correspond approximately to those found during previous inspections. Figure 8-25 gives amplitude and phase behavior of the largest indication. Two other sharp peaks have similar amplitude and phase patterns, which are presented in Figure 8-26. A characteristic of these indications is that the phase remains constant with frequency variation. On the other hand, the fourth indication, represented by smoother response, has a different amplitude and phase pattern, presented in Figure 8-27. It is likely that this indication is due to the false positive signal introduced during median filtering in the vicinity of a TSP. Compared to previously observed patterns for notches, all these patterns are rather unusual. This may be due either to response specific of large wear, or to phase distortions during filtering. More extensive testing is needed to show whether these patterns are consistent and whether they can be used for defect and TSP discrimination.

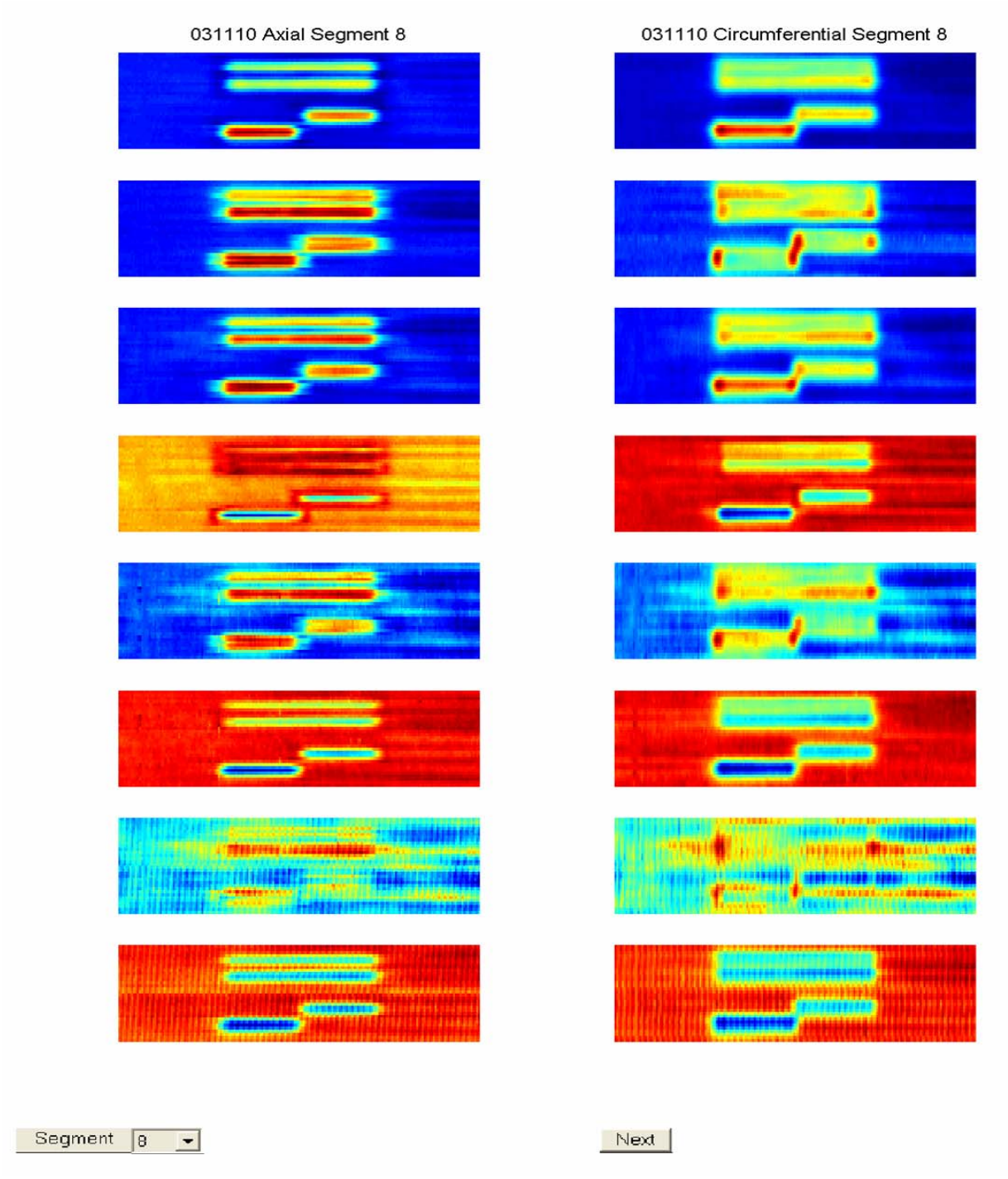

Figure 8-21
TSP Region (Horizontal and Vertical Components Grouped by Frequencies)

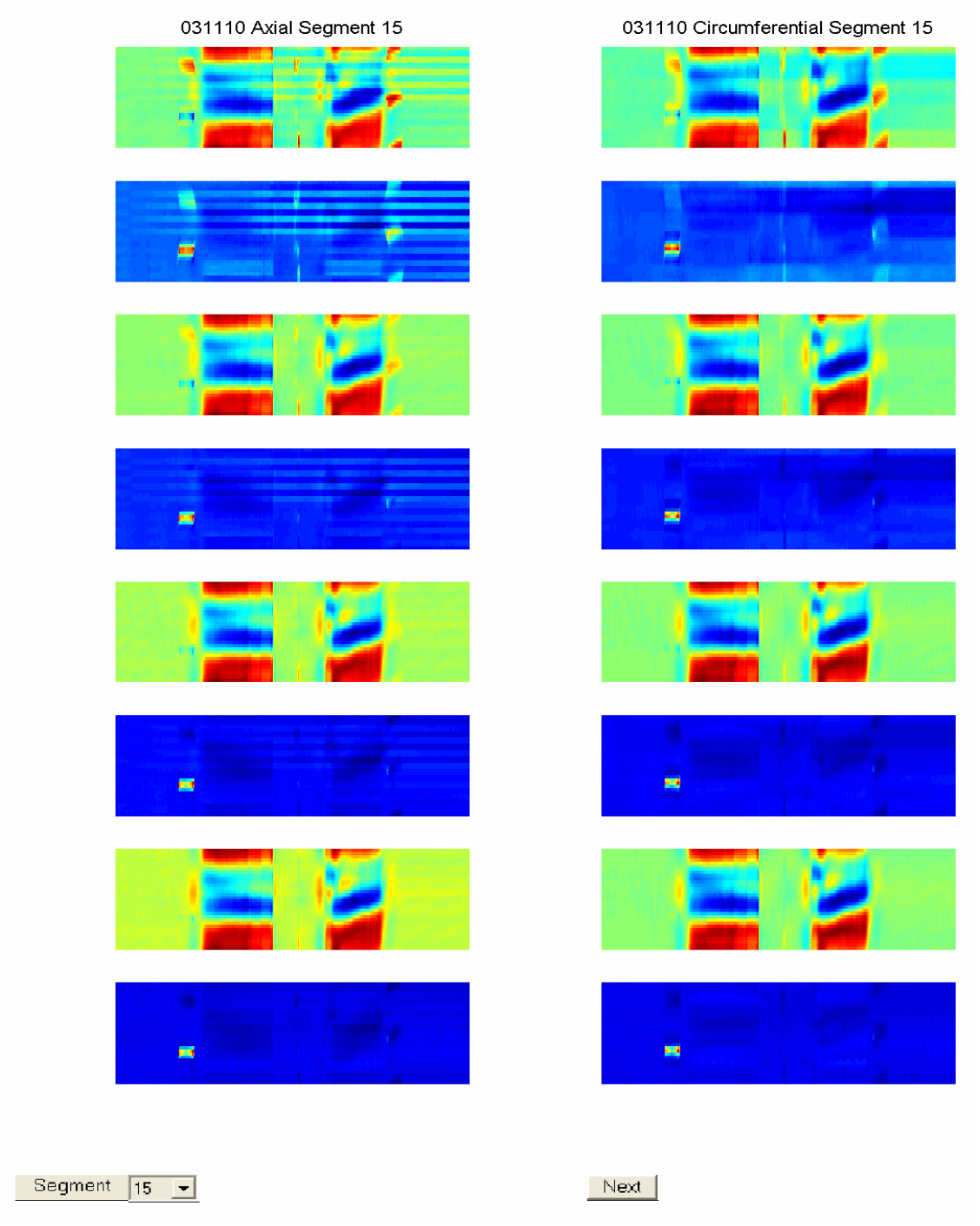

Figure 8-22
U-Bend Region (Horizontal and Vertical Components Grouped by Frequencies)

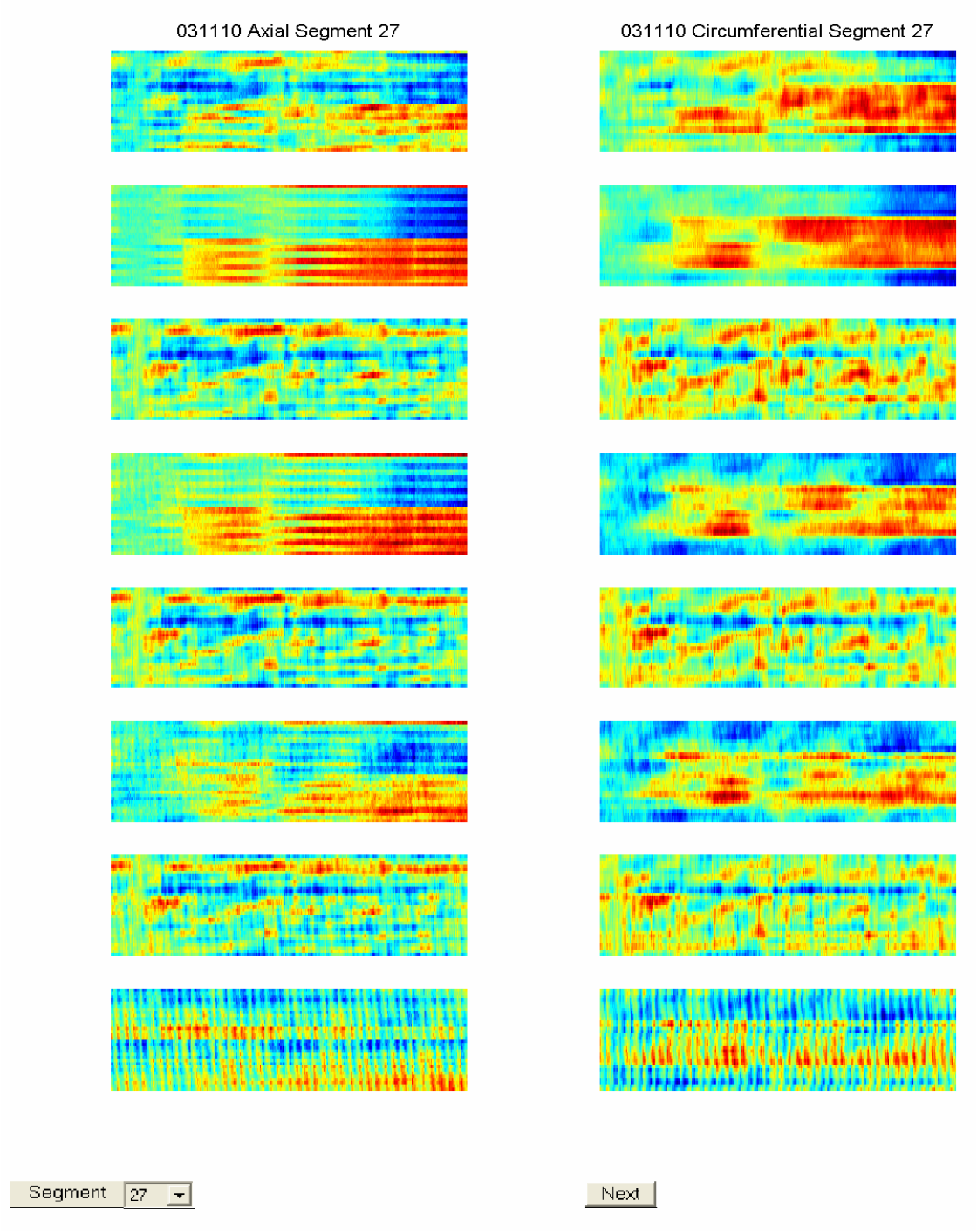

Figure 8-23
Free Span Region (Horizontal and Vertical Components Grouped by Frequencies)

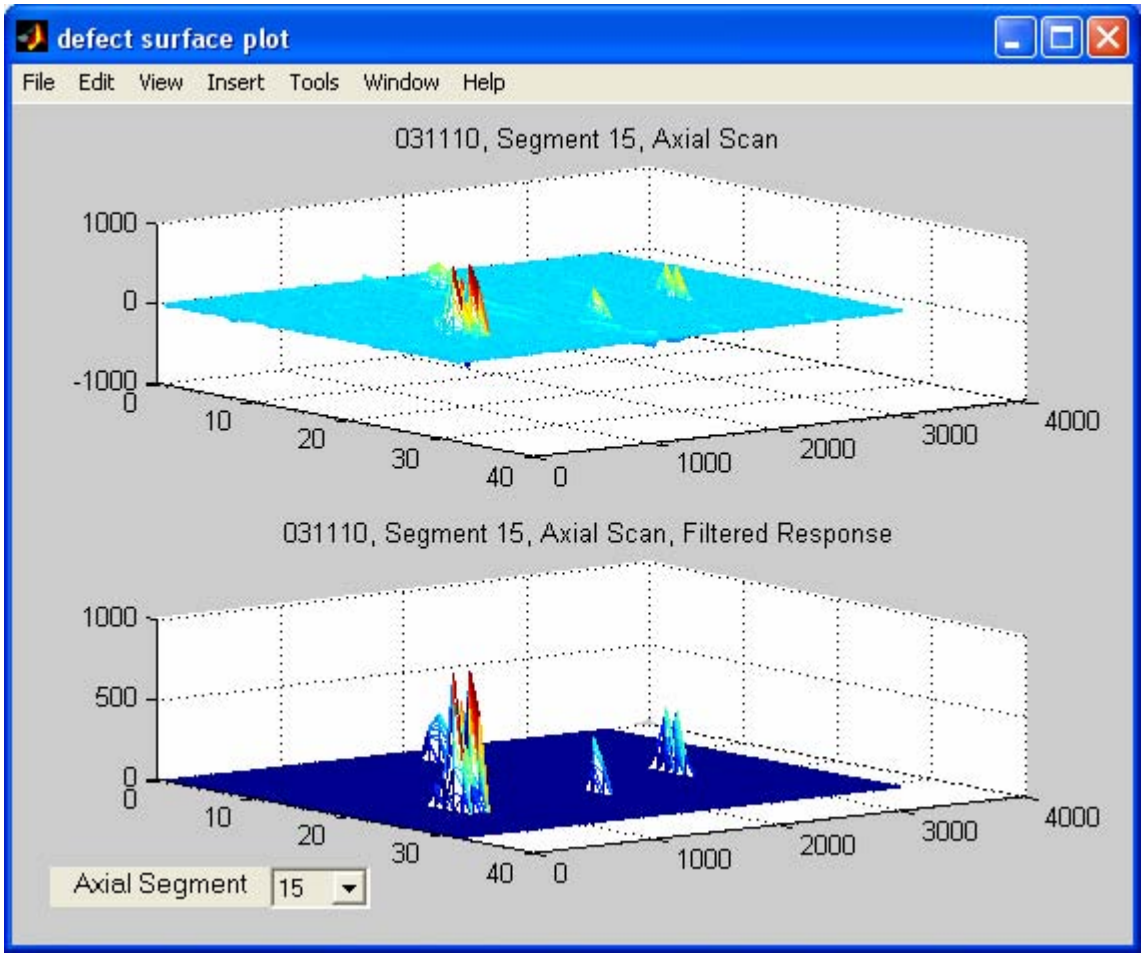

Figure 8-24
Detection in U-bend Region, Tube 031110

To further investigate amplitude and phase patterns due to incomplete TSP removal, another tube, numbered 086053, was analyzed. In the preliminary manual analysis in EddyView™ by project personnel, no defect was found, since the data was very noisy. The position designation available in the results of previous inspections by independent sources was not clear either. The only fact that was clear was that only one axial indication was found in previous inspections. ANTARES, however, found three axial indications in the U-bend region, presented in Figure 8-28. Indications were also found in a region with high noise, but were eliminated during characterization. The amplitude and phase patterns for indication in the U-bend region, presented in Figure 8-29, are very similar to those in Figure 8-27. The general phase trend is similar to OD notches, but the negative phase at 100 kHZ and the very fast amplitude decrease can be used to discriminate incomplete TSP removal from axial OD notches.

Although false alarms were eliminated successfully, the position of the reported single axial indication remained unresolved.

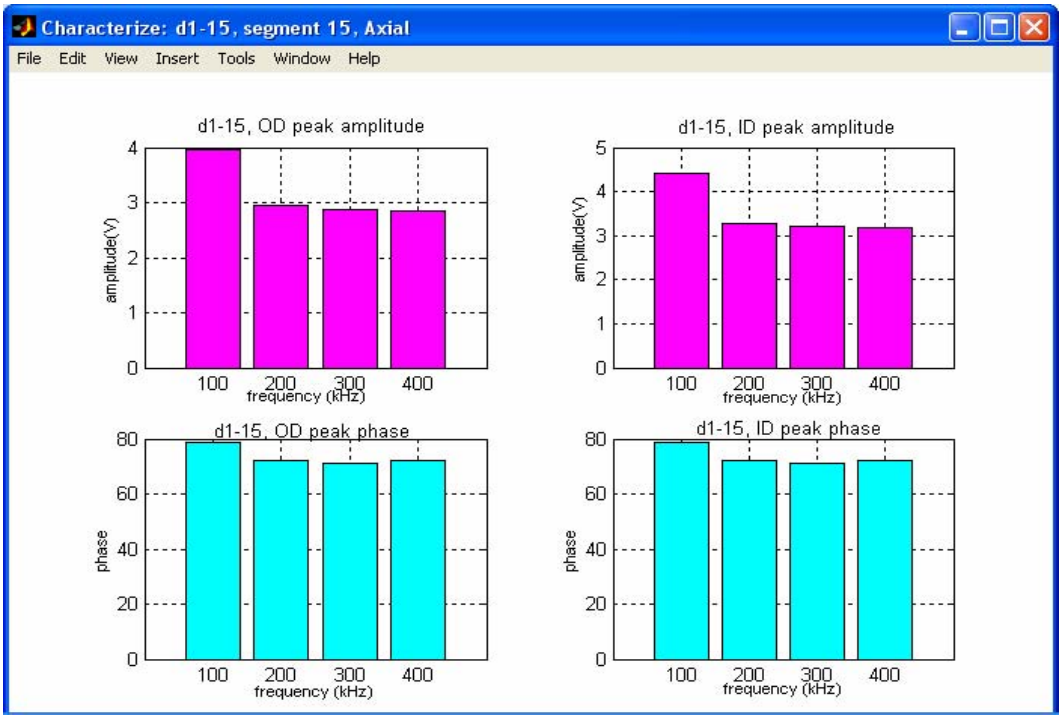

Figure 8-25
Large Volumetric Wear Characterization

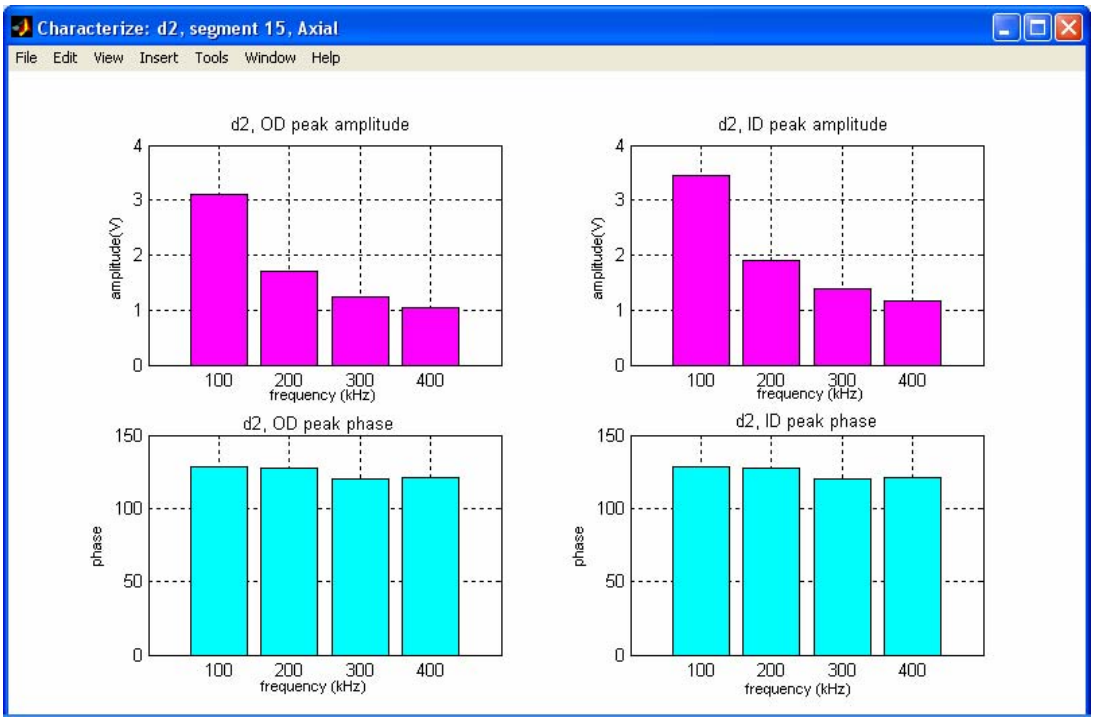

Figure 8-26
Amplitude and Phase Patterns for Smaller Wear

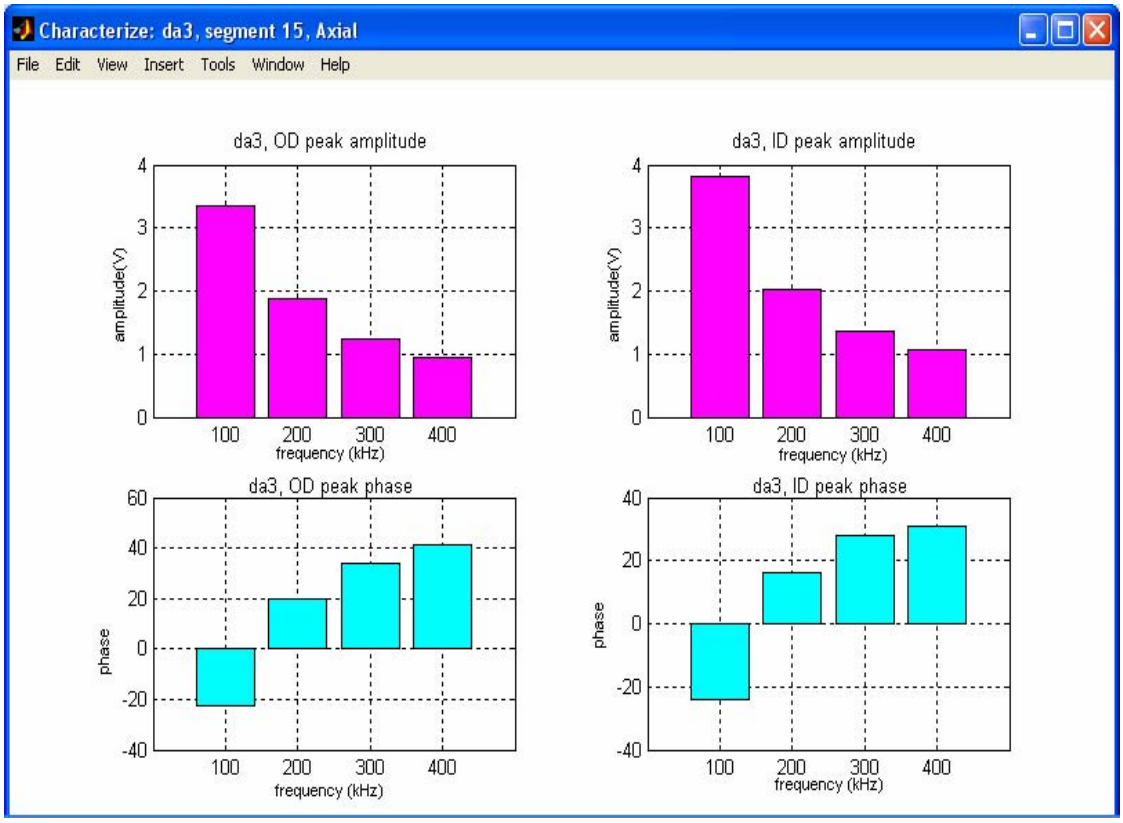

Figure 8-27
Amplitude and Phase Patterns for Incomplete TSP Removal, Tube 031110

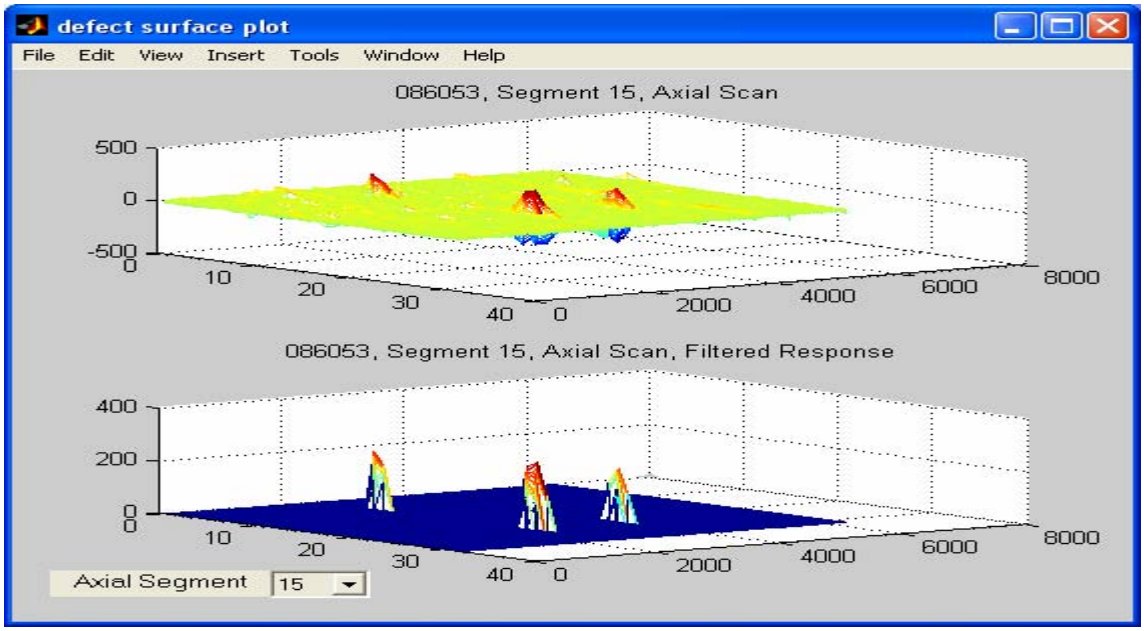

Figure 8-28
Detection in U-Bend Region, Tube 086053

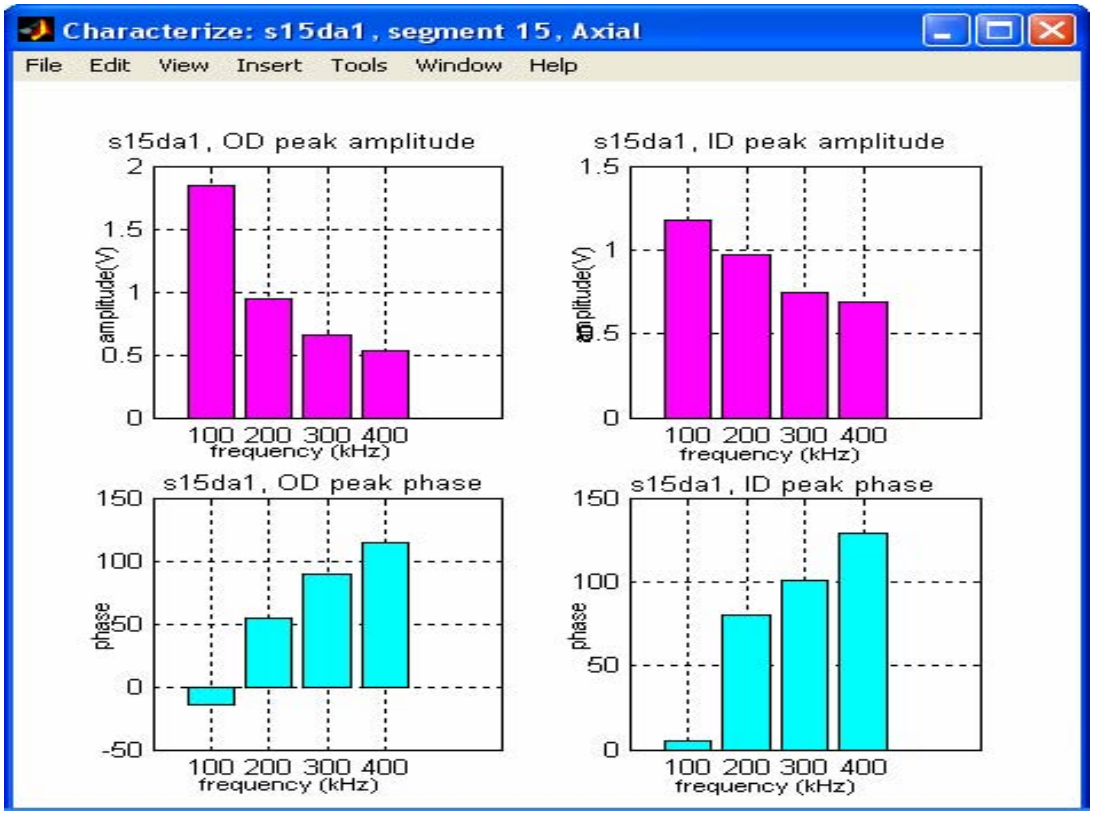

Figure 8-29
Amplitude and Phase Pattern for Incomplete TSP Removal, Tube 086053

9
CHARACTERIZATION ALGORITHM TESTING

The main function of the characterization algorithm in the original version of ANTARES is to determine the profile and the length of the defects found in the detection algorithm. However, this algorithm also serves for further false alarm elimination, as described in the previous section.

9.1 Fractography Benchmark

The performance of the characterization algorithm has been evaluated on a set of 8 laboratory grown defects, which were destructed and analyzed using fractography. The comparison of ANTARES estimates, fractography results, and RPC estimates is given in Table 9-1.

Table 9-1
Fractography Benchmark Results

Sample		Depth Sizing TW (%)			Length (mm)		
Number	Type	ANTARES	Fractography	RPC	ANTARES	Fractography	RPC
1	LOD	90	100	90	21	22	22
2	LOD	80	80	75	12	14	14
3	COD	100	100	95	22	14	20
4	LID	42	50	30	13	10	10
5	LODSCC	75	70	60	8	14	12
6	LOD	60	100	70	10	15	15
7	LOD	80	100	87	10	18	8
8	LOD	35	20	40	20	10	20

Figures 9-1 and 9-2 give graphical comparison of characterization results.

Table 9-1 and Figures 9-1 and 9-2 show that, on the average, the maximum depth is estimated with a reasonable accuracy. However, some extreme errors have been observed, mainly due to local artifacts. Similar problems have already been described in defect detection and classification. The results of the maximum depth estimation are in most cases similar to RPC results.

Length estimation is less accurate. Length is usually underestimated, due to conservative thresholding in the detection and filtering steps. The exception to this rule is possible overestimation for small defects, where it is difficult to draw the boundary between the defect and background.

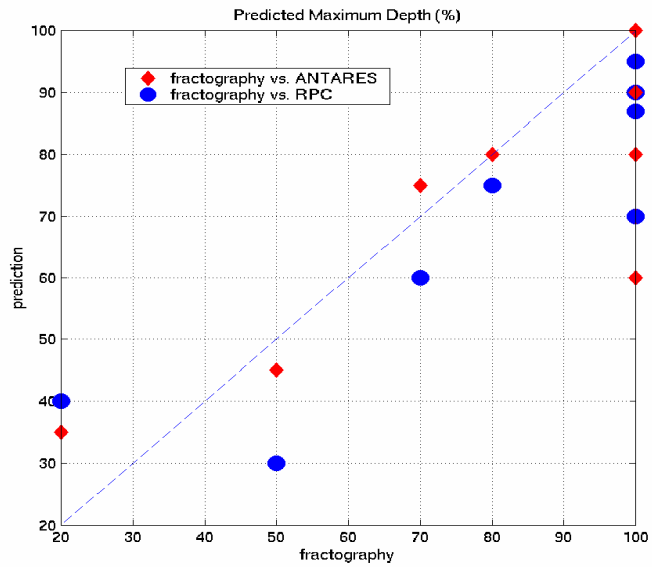

Figure 9-1
Depth Estimation Evaluation

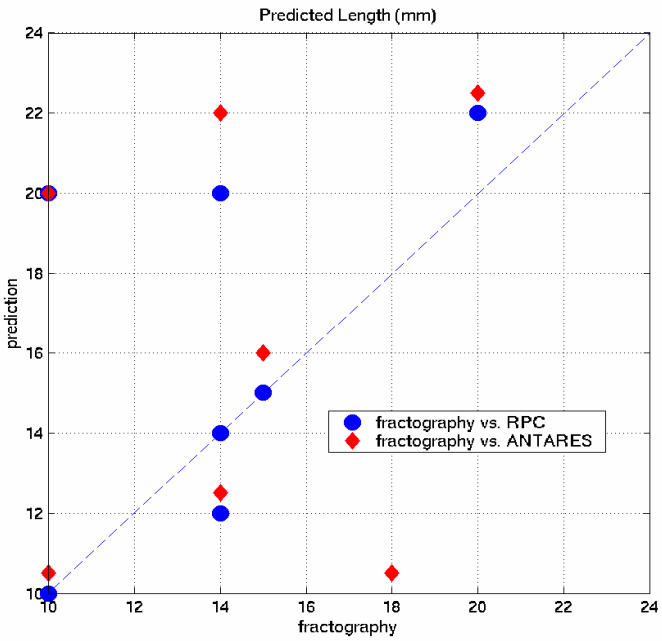

Figure 9-2
Length Estimation Evaluation

The estimated depths are in good agreement with fractography results for the first 5 samples, with a maximum discrepancy of 10%. However, the estimates for the last three samples deviate significantly from measured values. Sample 6 is a through wall axial crack located on the edge of the tube support plate. It is very difficult to subtract background in this region using the masking

technique implemented in ANTARES, since the boundaries of the defect change significantly. As a consequence, a part of the signal is also subtracted, leading to a severe underestimation in both depth and length. The profile of this defect, presented in the following section in Figure 9-8, is very different from the actual defect. Sample 7 is a through wall defect in an artifact that produces significant positive vertical component. Again, direct subtraction of the artifact signal reduces the defect signal and also reduces estimated depth and length. On the contrary, both the depth and length of the small defect in sample 8 are overestimated. This is due to the difficulty in discriminating signal from noise for small defects.

9.2 Profile Comparison

Estimated profiles show good agreement only in few cases, but in majority of cases exhibit only qualitative agreement with fractography data. Detailed point-by-point profiling algorithms usually implemented for RPC probe signals do not seem appropriate for X-probe™ signals, since there is a large uncertainty in phase and amplitude variation. Therefore, the profiling implemented here is very approximate and is based on the rescaled amplitude. The objective is to provide an estimate of the equivalent crack depth to be used in the burst pressure estimation algorithm.

Estimated profiles are compared to fractography data and presented in Figures 9-3 through Figure 9-10. The largest discrepancy is for the last three samples, mainly due to incorrect maximum depth estimation. The difference in the measured and estimated profile shape is large since the defect signal amplitude is not a good estimator of the depth.

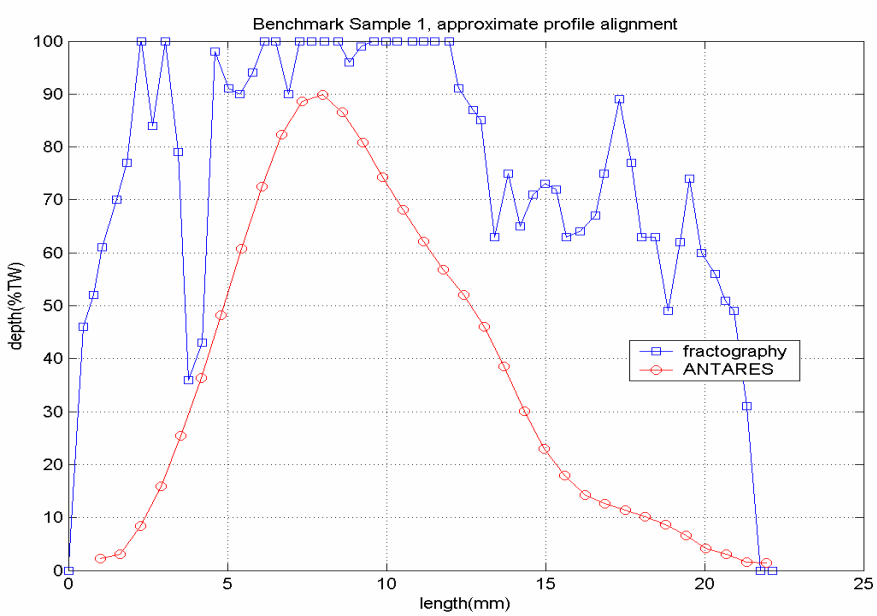

Figure 9-3
Profile Estimation for Sample 1

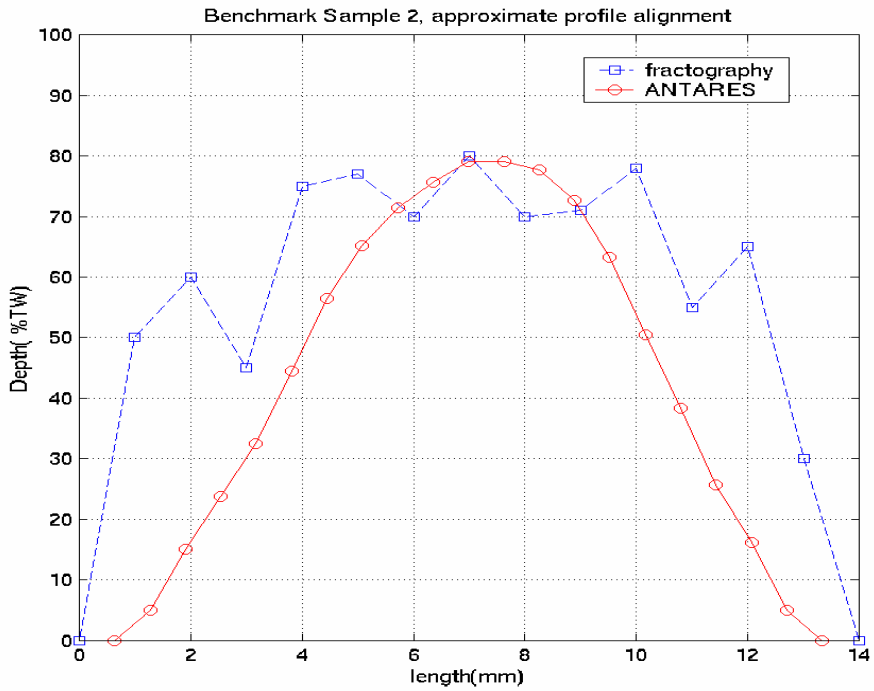

Figure 9-4
Profile Estimation for Sample 2

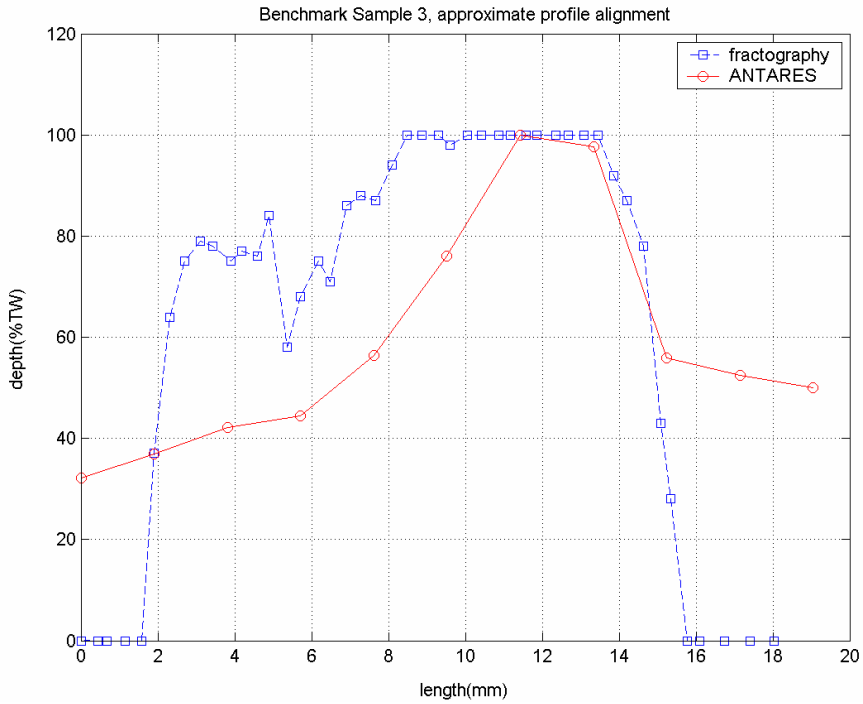

Figure 9-5
Profile Estimation for Sample 3

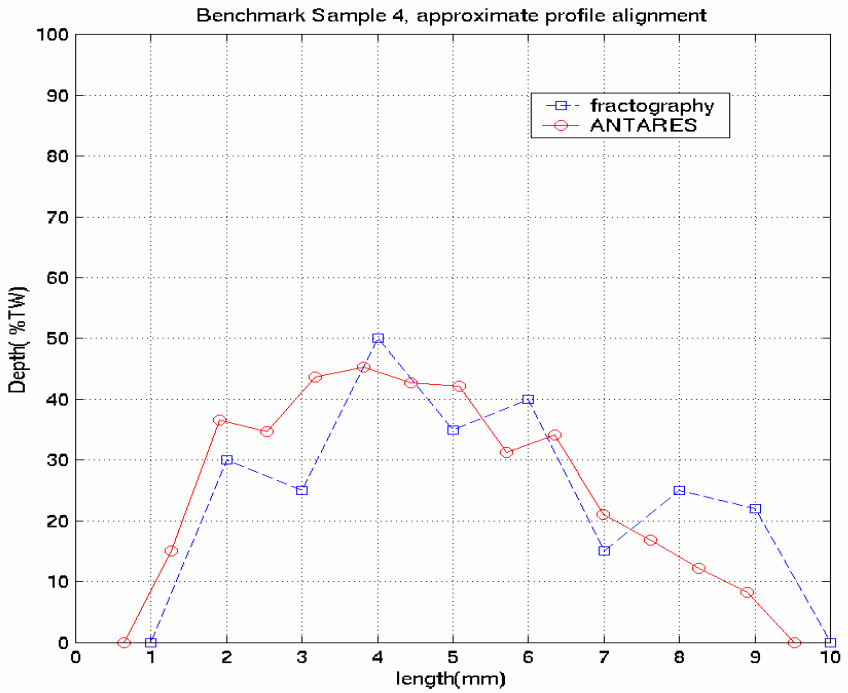

Figure 9-6
Profile Estimation for Sample 4

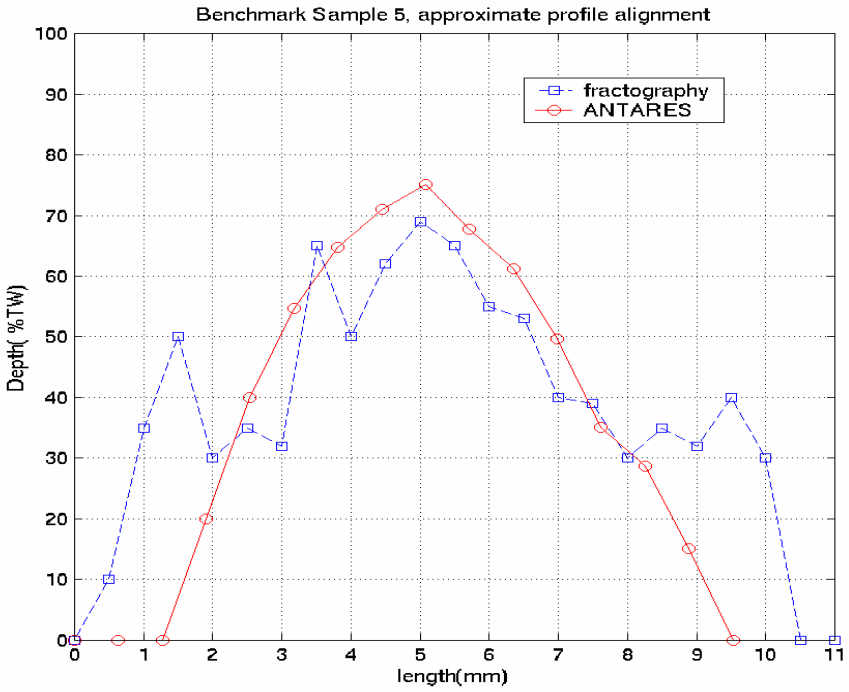

Figure 9-7
Profile Estimation for Sample 5

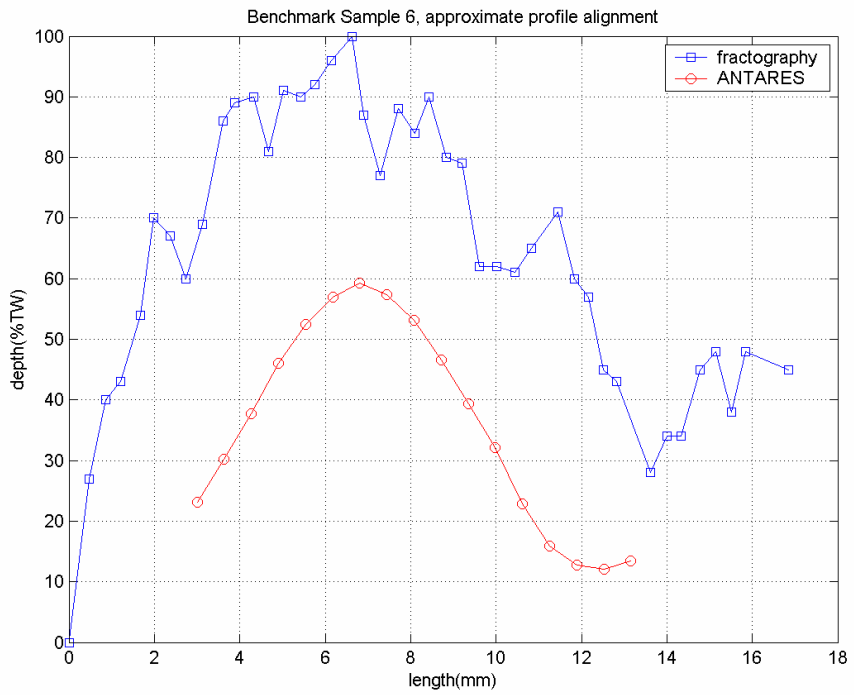

Figure 9-8
Profile Estimation for Sample 6

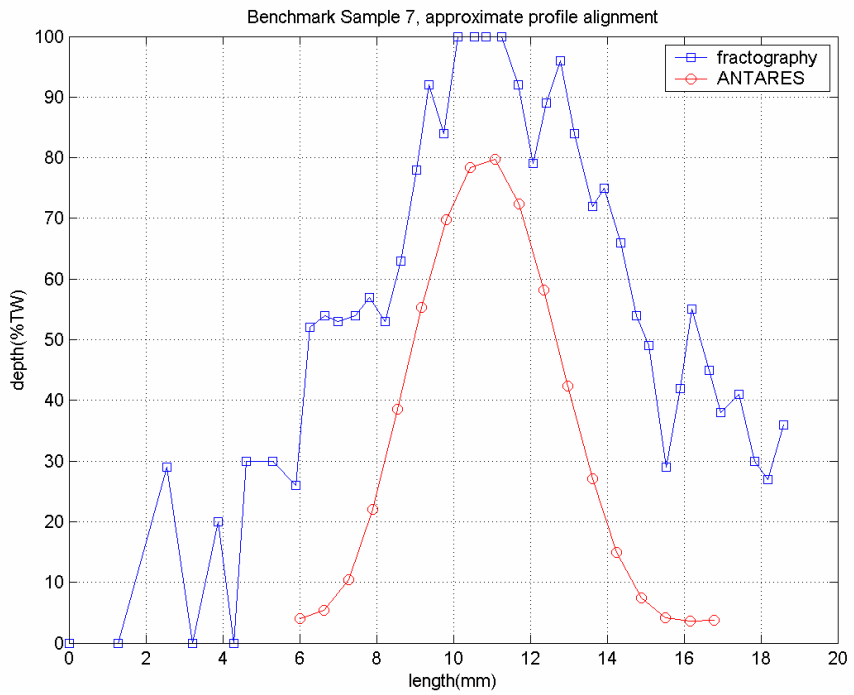

Figure 9-9
Profile Estimation for Sample 7

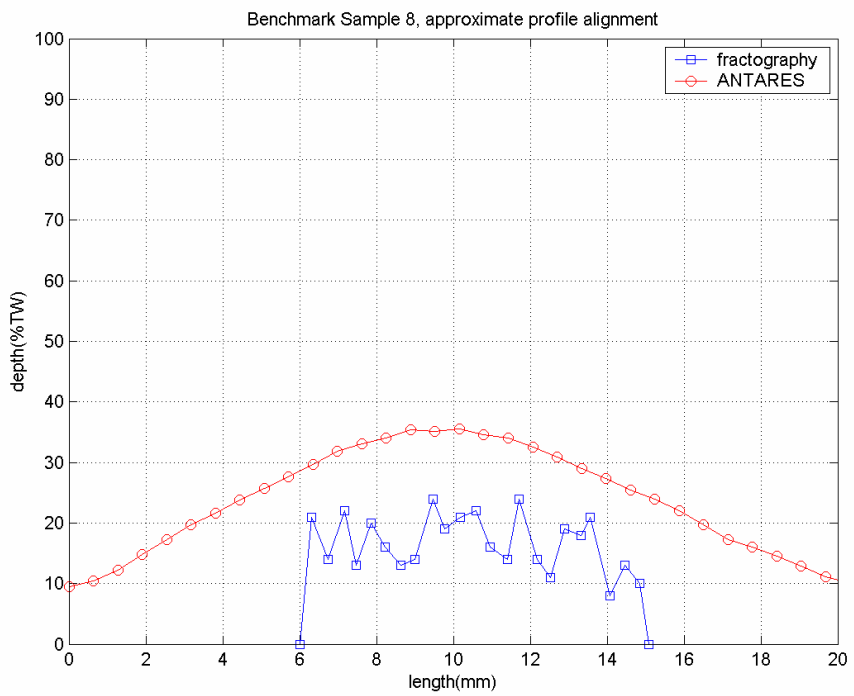

Figure 9-10
Profile Estimation for Sample 8

10
BURST PRESSURE ESTIMATION TESTING

This chapter discusses the results of select verification and validation tests for the Burst Pressure Estimation module. The verification process is performed to provide assurance that the module individual units and functions as well as the interrelationship between different modules meet the design requirements. The validation process tests the functionality of the code as a whole and checks how the code results compare with physical reality.

The first section describes the design and implementation of select verification tests. The second section discusses the validation test results performed for this project.

Experimental data points in the P_l database (see Table 6-1) were used to test the Burst Pressure Estimation module in ANTARES. The first six entries in Table 6-1 correspond to lab-grown stress corrosion crack (SCC) profile P_l experiments performed at ANL [15, 25]. For these experiments, posttest fractography crack profiles were available. As mentioned previously, the fractography profiles were converted into equivalent rectangular crack profiles by the methodology described in Section 5.1. Then, the experimental P_l along with the equivalent rectangular crack dimensions and SGT properties were entered into the corresponding fields as experimental data points in the P_l database.

Similarly, the last entry in Table 6-1 corresponds to an electro-discharge machining (EDM) notch profile P_l experiment also performed at ANL [34]. This experimental data point was also included in the P_l database and used in the V&V process for the Burst Pressure Estimation module.

Note also that the material properties and environmental conditions for the SGT used in these experiments corresponds to the most conservative case discussed previously, for which the SVM algorithm was trained using the data generated by the analytical model described in Section 5.2.

10.1 Module Verification

During module verification, it was determined whether the Burst Pressure Estimation module was correctly implemented in ANTARES and produced expected results. Selected results are presented here with the focus on comparison of SVM estimates with analytical model estimates.

An independent driver that loads the crack profiles and reports the analytical P_l and P_b estimations was developed in MATLAB. In addition, independent SVM calculations were performed using as input the corresponding equivalent rectangular crack dimensions and the optimum parameters and coefficient files produced in the SVM training phases. The results of the independent driver and SVM computations are summarized in the last four columns in Table 10-1. The corresponding Burst Pressure Estimation module results for the same inputs are shown in columns 2 through 5 in Table 10-1.

The first seven entries in Table 10-1 correspond to the SCC experimental data points shown in Table 6-1. For the first six data points, posttest fractography crack profiles were available and were used to compute the corresponding equivalent rectangular crack dimensions. These fracrography profiles were input to both the Burst Pressure Estimation module and the independent driver. As shown above, the independent SVM computation were performed by using the equivalent rectangular crack dimensions as one of the inputs.

Table 10-1
Test Results

SGT ID	ANTARES Results				Independent Calculations			
	P_l, MPa		P_b, MPa		P_l, MPa		P_b, MPa	
	Anal	SVM	Anal	SVM	Anal	SVM	Anal	SVM
413*	24.3	26.2	36.9	36.9	24.3	26.2	36.9	36.9
480*	8.9	18.3	33.4	33.3	8.9	18.3	33.4	33.3
494*	35.9	36.5	22.9	22.9	35.9	36.5	22.9	22.9
493*	15.9	18.8	31.0	30.9	15.9	18.8	31.0	30.9
412*	27.6	26.9	31.6	31.6	27.6	26.9	31.6	31.6
732*	14.5	14.5	21.8	21.9	14.5	14.5	21.8	21.9
T3EA80X15**	13.6	13.3	10.1	10.0	13.6	13.3	10.1	10.0
aaa***	27.5	26.8	24.4	24.4	27.5	26.8	24.4	24.4
bbb***	33.8	33.1	33.4	33.4	33.8	33.1	33.4	33.4

Legend:

* For the analytical calculations, the input for the Burst Pressure Estimation module and independent driver was the posttest fractography crack profile. For SVM calculations, one of the inputs was the corresponding equivalent rectangular crack dimensions calculated from the fractography profile.

** The input for all calculations was the EDM notch profile.

*** For the analytical calculations, the input for the Burst Pressure Estimation module and independent driver was a crack profile estimated by the flaw characterization module of ANTARES. For SVM calculations, one of the inputs was the corresponding equivalent rectangular crack dimensions.

For the seventh entry in Table 10-1, the input for all equivalent calculation discussed above was the EDM notch crack dimensions.

Finally, the last two entries in Table 10-1 are the tests performed to check the ANTARES code functionality as a whole. Two SGT with defects were chosen for this test. ANTARES analyzed these two SGT and the Burst Pressure Estimation module results were recorded. In addition, the crack profiles estimated during the ANTARES runs were extracted (aaa and bbb). As before, these profiles were input to the independent driver and the corresponding equivalent rectangular crack dimensions represented one of the inputs for the equivalent SVM computations.

As expected, the analytical calculations produced by the ANTARES Burst Pressure Estimation module and those produced by the independent driver were identical since the same input was used (see columns 2 and 6 and columns 4 and 8 in Table 10-1). The equivalent SVM computation produced identical results (see columns 3 and 7 and columns 5 and 9 in Table 10-1).

By further analyzing the results shown in Table 10-1 it is seen that the equivalent case results for the SVM and analytical results are generally similar (see columns 2 and 3; 4 and 5; 6 and 7; and 8 and 9 respectively in Table 10-1). This result is expected since the SVM models were trained with data generated by the analytical model.

However, there are 2 cases (see entries for SGL-480 and 493, i.e. entries 2 and 4 in Table 10-1) for which the SVM P_l estimations differ significantly from similar analytical results. This is explained by the fact that the SVM training range was limited to values in between 10 and 90 % TW for equivalent rectangular depth and up to 1.5 inch for the equivalent rectangular length. For these two cases, the equivalent rectangular depth was calculated to be 99 and 93 % TW, respectively. The equivalent rectangular depth value for SGL-480 is significantly outside the SVM training range. The extrapolation introduced by the SVM algorithm performed reasonably for SGL-493. Nevertheless, if the rules discussed in Table 7-1 were to be enforced (see Section 10.2), the SVM estimation results would have never been reported (see Case 5 in Table 7-1).

Currently, a fair comparison between the SVM and analytical results can be made by analyzing the results produced by the data points in the SVM training data set. Thus, the analytical P_l estimates were plotted against the corresponding SVM estimates in Figure 10-1. Similarly, the analytical P_b estimates were plotted against the corresponding SVM estimates in Figure 10-2. The solid line in these figures represents the "perfect" estimates.

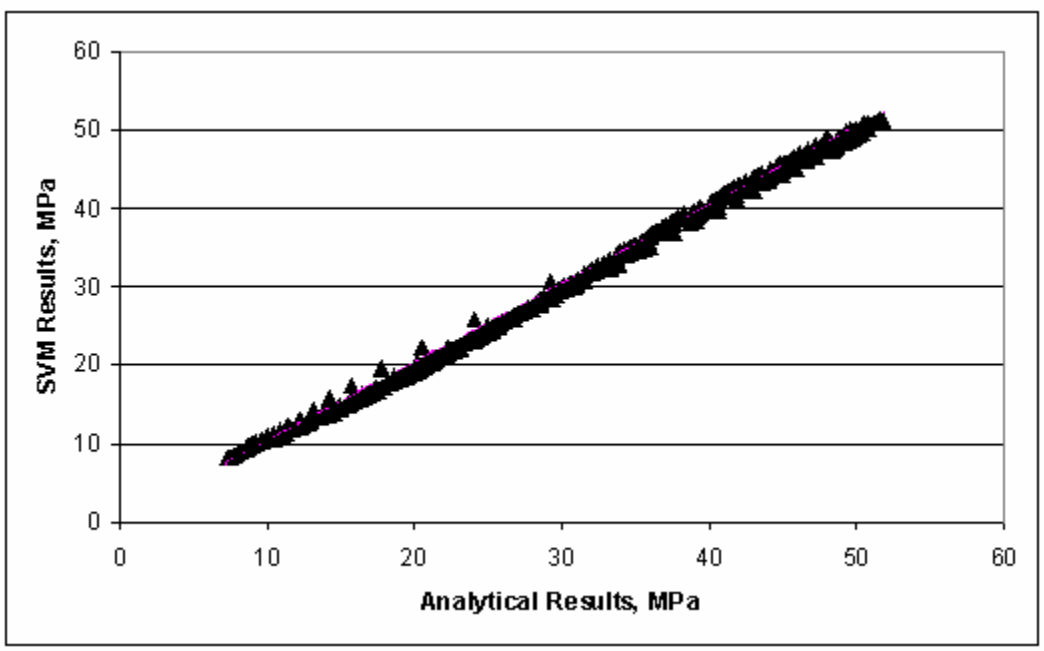

Figure 10-1
SVM vs. Analytical Estimation for SVM Training Data in the P_l Database

It can be seen from Figures 10-1 and 10-2 that an appropriately trained SVM performed very well for the SVM training data sets included in the P_l and P_b databases. Thus it can be concluded that the SVM algorithm is appropriate for this application. Similar performance is expected if sufficient experimental data points are found to properly train an experimentally-based SVM model.

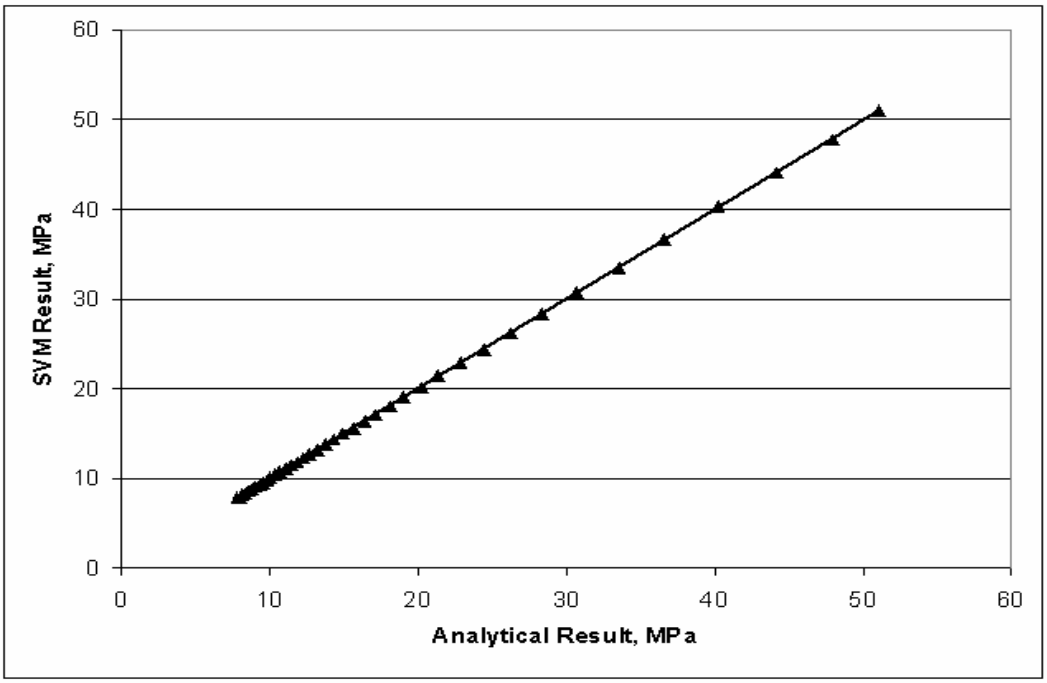

Figure 10-2
SVM vs. Analytical Estimation for SVM Training Data in the P_b Database

10.2 Module Validation

This section discusses the performance of the Burst Pressure Estimation algorithm in ANTARES when compared with experimental results. The validation was performed only for the experimental cases summarized in Table 6-1 and discussed at the beginning of this section. As shown previously, these experimental data points represent the case that theoretically would produce the most conservative P_l and P_b estimations due to the SGT material properties. This conservative case was used in the analytical data generation for the SVM training.

The validation results are summarized in Table 10-2. Column 4 in Table 10-2 shows the experimental P_l reported in references [15, 25 and 34]. No data relevant to the most conservative case was found in these references for P_b experiments (see column 7 in Table 10-2). Columns 5 and 6 and 8 and 9 in Table 10-2 present the ANTARES estimations for P_l and P_b respectively. Finally, columns 2 and 3 in Table 10-2 show the computed or real equivalent rectangular crack lengths and depths, respectively.

Table 10-2
Comparison between Burst Pressure Estimation Module and Experimental Results

SGT ID	L_e, in	D_e, in	P_l, MPa			P_b, MPa		
			Exp	Anal	SVM	Exp.	Anal.	SVM
SGL-413	0.25*	0.045*	12.4	24.3	26.2	N/A	36.9	36.9
SGL-480	0.30*	0.049*	6.2	**	**	N/A	33.4	**
SGL-494	0.55*	0.024*	32	35.9	36.5	N/A	N/A***	N/A***
SGL-493	0.34*	0.046*	11.5	**	**	N/A	31.0	**
SGL 412 or 535	0.33*	0.039*	10.3	27.6	26.9	N/A	31.6	31.6
SGL 732 or 855	0.58*	0.043*	13.8	14.5	14.5	N/A	21.8	21.9
T3EA80X1.5	1.5	0.04	13.8	13.6	13.3	N/A	N/A***	N/A***

Legend:

N/A Not Applicable.

* The equivalent rectangular crack dimensions were computed from posttest fractography crack profile.

** Corresponding to Case 5 reporting rule in Table 7-1. The SGT ligament is conservatively declared ruptured by ANTARES.

*** Supplementary reporting rule (see Section 7) applies.

Note: the SGT characteristics material properties and temperature were considered constant in both models (see Table 6-1) and therefore the SVM model was based only on crack or equivalent rectangular crack dimensions.

It can be seen in Table 10-2 that ANTARES performed well in estimating P_l for three cases: SGL-494, SGL-732 and the EDM notch T3EA80X1.5. In two cases (SGL-480 and SGL-493), ANTARES conservatively reports that the SGT ligament has ruptured. Since the experimental pressure for which this actually happens is relatively small, this is not a bad approximation. However, for currently unknown reasons, ANTARES significantly overestimates P_l for SGL-412 and 413.

By further analyzing the results in Table 10-2, it can be seen that usually the analytical model overestimates (sometimes significantly) the experimental data for P_l. Since the SVM model is trained on analytically-generated data, the SVM estimation performs similarly. However, the SVM results have the potential to become closer to the experimental results if the training is performed on experimental data. Further research should be performed in order to populate the corresponding database with new experimental points. Thus, the P_l and P_b estimation will become more reliable and robust since it will be based on two different methodologies, an experiment-based method and an analytical method.

11
CONCLUSIONS AND FUTURE WORK

This document describes the flaw detection and characterization algorithms implemented in the original version of the Automated Non-Destructive Testing Array Evaluation System (ANTARES), as well as modification for implementation in the EPRI AutoAnalysis software. Also included is a description of the machine learning algorithm for the follow-on tube burst pressure estimation. The implementation and performance testing of the original version is based on the currently available X-probe™ data collected at the U.S. NRC mock-up at Argonne. Testing on additional data provided by EPRI, such as ETSS data, U-bend data and some full length tube data is also presented for the modifications. Finally, testing of the machine learning algorithm for the follow-on tube burst pressure estimation is included.

Detection performance of the original algorithm has been tested on two data sets and the results are evaluated separately. Test 1 includes 25 mock-up tubes with axial OD SCCs in the TSP region. Only two defects were missed, with 5 false calls. The average performance is good, with the missed alarm rate of 5% and the false alarm rate of 2.7%. The probability of detection curve based on this data set shows that defects larger than 40% TW could be detected with probability of near 60% or larger at the 90% lower confidence limit (LCL). Exception to this rule is a missed 70% TW crack in a dent. Probability of detection for cracks less than 40% TW is larger than 50% at the 90% LCL. Test 2 includes 25 mock-up tubes with various SCCs. For this data set, 7 defects were missed, with 4 false calls. Therefore, the false alarm rate is only 2%, but the missed alarm rate is 17.5%. This result indicates that the characterization rules are not properly optimized to balance false and missed alarm rates. Although the missed alarm rate is large, most missed defects are shallow, which is evident from the shape of the estimated POD curve. The POD curve indicates that the cracks larger than 30% TW can be detected with the probability larger than 60% at the 90% LCL. Moreover, the probability of detection for cracks larger than 80% TW is 85% at the 90% LCL. The probability of detection for cracks less than 30% appears to be small. Poor detection is also observed for defects in dents and at roll transitions. Some improvement could be accomplished using better characterization rules or filtering. However, in some cases the probe signal is too small to be detected by an automatic algorithm with a reasonable false alarm rate.

The performance of the characterization algorithm in the original ANTARES version has been evaluated on a set of 8 defects, which were examined using fractography. The estimated depths are in good agreement with fractography results for five samples, with the maximum discrepancy of 10%. However, the estimates for three samples deviate significantly from measured values. Two samples were large defects in artifacts. Artifact signal removal also removed a significant part of the signal, resulting in a substantial depth underestimation. The third sample was a very small defect. The depth was overestimated, since a part of the noise was not removed, due to a low signal to noise ratio. In all cases, the difference between the ANTARES estimates and a characterization algorithm based on RPC signals is 10% or less.

Length estimation is less accurate. Length is usually underestimated, due to conservative thresholding in the detection and filtering steps. An exception to this rule is possible overestimation for small defects, where it is difficult to draw the boundary between the defect and background.

Estimated profiles show good agreement only in a few cases, but in majority of cases exhibit only qualitative agreement with fractography data. Detailed point-by-point profiling algorithms usually implemented with RPC probe signals do not seem appropriate for X-probe™ signals, since there is a large uncertainty in phase and amplitude variation. Therefore, the profiling implemented here is very approximate and is based on the rescaled amplitude. The objective is to provide an estimate of the equivalent crack depth to be used in the burst pressure estimation algorithm.

Comparison of burst pressure estimation results with limited experimental data indicates that ANTARES performed well in estimating P_l for three cases, while in two cases ANTARES conservatively reports that the SGT ligament has ruptured. Since the experimental pressure for which this actually happens is relatively small, this is not a poor approximation. However, for currently unknown reasons, ANTARES significantly overestimates P_l for two tubes.

Furthermore, the analytical model overestimates (sometimes significantly) the experimental data for P_l. Since the SVM model is trained on analytically-generated data, the SVM estimation performs similarly. However, the SVM results have the potential to draw closer to the experimental results if the training is performed on experimental data. Further research should be performed in order to populate the corresponding database with new experimental points. The P_l and P_b estimation will become more reliable and robust since it will then be based on two different methodologies, an experiment-based method and an analytical method.

To address challenges posed by automatic analysis of array probe field data, several modifications in ANTARES have been implemented. A new set of calibration functions has been evaluated for the probe frequencies of 100 kHz, 200 kHz, 300kHz, and 400kHz. Filtering algorithms have been introduced to reduce ovalization effects in the U-bend region. A semi-automatic segmentation algorithm has been developed, based on principal component analysis and filtering. Region-specific detection has been implemented.

The modified algorithm was tested on ETSS data. The modified algorithm missed one of fifteen defects provided for testing. Difficulties in discrimination between ID and OD defects have been observed and the classification algorithm has been modified to report two defect classes only: “indication” and “ non-detectable degradation”. The maximum depth of large defects was estimated with good accuracy.

Preliminary testing in the U-bend region indicated that it is feasible to find filtering and detection parameters to control detection sensitivity and false alarm rate. Examples of initial full length data analysis indicated that it is possible to find large degradations, even with incomplete TSP removal. Detection of small defects will probably require better algorithms for TSP removal.

Detailed and reliable information about degradations would enable better training of the detection and classification algorithms with the goal to make the software applicable for field inspection.

Recent software development efforts resulted in improved automation in the detection procedure through automated region of detection extracts and automated implementation of classification rules.

Future development effort may include:

- improved algorithms for defect detection in the TSP regions and U-bends;
- improved artifact removal;
- further evaluation, improvement, and automation of characterization rules, as well as development of new classification and characterization algorithms.

12
REFERENCES

1. L. S. Obrutsky, V. S. Cecco, S. P. Sullivan, and D. Humphrey, "Transmit-Receive Eddy Current Probes for Circumferential Cracks in Heat Exchanger Tubes," Material Evaluation, (January 1996).
2. J. Renaud, C, Fogal, M. Klatt, F. Hardy, and G. Lafontaine, "X-Probe™, The Year in Review," 20th Annual EPRI Steam Generator NDE Workshop, Orlando, FL, (July 2001).
3. J. Kang, J. Renaud, and M. Klatt, "X-Probe™ Experience at Diablo Canyon," 22nd Annual EPRI Steam Generator NDE Workshop, Hilton Head, SC, (July 2003).
4. S. Bakhtiari, J. Y. Park, D.S. Kupperman, S. Majumdar, and W.J. Shack, "Advanced Nondestructive Evaluation for Steam Generator Tubing,", NUREG/CR-6746, (September 2001).
5. J. Benson, S. Mukherjee, S. Soo, U. Parikkar, P. Ramuhalli, L. Udpa, and S. S. Udpa, "Development of Automated Analysis Algorithms for Rotating Probe Data," 23rd Annual EPRI Steam Generator NDE Workshop, Chicago, IL (July 2004).
6. Y. Iwahashi, S. Ishikawa, T. Saito,Y. Asada, T.Kinoshita, M. Kurokawa, K. Kawata, and N. Kawase, "Recent Improvements of Intelligent ECT System," 22nd Annual EPRI Steam Generator NDE Workshop, Hilton Head, SC (July 2003).
7. T. Saito, Y, Asada, T. Kinoshita, M. Kurokawa, K. Kawata and N. Kawase, Y. Iwahashi, "Success Story of Intelligent ECT," 23rd Annual EPRI Steam Generator NDE Workshop, Chicago, IL (July 2004).
8. J. Benson, P. Ramuhalli, G. Arunachalam, P. Jella, U. Parrikar, L. Udpa, and S.S. Udpa, "Development of Automated Analysis Algorithms for Array Probe Eddy Current Data," 23rd Annual EPRI Steam Generator NDE Workshop, Chicago, IL (July 2004).
9. EddyView™, R/D Tech Inc. (2003).
10. MATLAB, the Language of Technical Computing, Using MATLAB Version 6, The MathWorks, Inc. (November, 2000).
11. Advanced NDE System for Detection and Characterization of Defects in Steam Generator Tubes, NEPO Projects 3-2 Report, Argonne National Laboratory (September 2001).
12. Automated Nondestructive Testing Array Evaluation System (ANTARES), Part I – Algorithm Description, NEPO Project 3-2 Final Report, Argonne National Laboratory (September 2003).
13. Automated Nondestructive Testing Array Evaluation System (ANTARES), Part II – Testing Documentation, NEPO Project 3-2 Final Report, Argonne National Laboratory (September 2003).

14. Automated Nondestructive Testing Array Evaluation System (ANTARES) Part III User Manual, Version 1.0, NEPO Project 3-2 Final Report, Argonne National Laboratory (September 2003).
15. S. Majumdar, S. Bakhtiari, K. Kasza, J. Y. Park, "Validation of Failure and Leak-Rate Correlations for Stress Corrosion Cracks in Steam Generator Tubes", NUREG/CR-6774, ANL-01/34, Argonne, IL, 2002.
16. V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
17. C. Yu and N. Zavaljevski, ActiveSVM User's Manual, ANL, Argonne, IL, 2003.
18. A. Wald, Sequential Analysis, John Wiley and Sons, New York, (1947).
19. J. C. Russ, The Image Processing Handbook, CRC Press, IEEE Press, (1999).
20. I. Pitas, A.N. Venetsanopoulos, Nonlinear Digital Filters, Kluwer, Boston,(1990).
21. MATLAB Image Processing Toolbox, The MathWorks, Inc., Natick, MA, (April 2001).
22. MATLAB Wavelet Toolbox, The MathWorks, Inc., Natick, MA, (September 2000).
23. MATLAB Statistical Toolbox, The MathWorks, Inc., Natick, MA, (July 2002).
24. S. Majumdar, Personal Communication, ANL, Argonne, IL, 2003.
25. S. Majumdar, K. Kasza, and J. Franklin, "Pressure and Leak-Rate Tests and Models for Predicting Failure of Flawed Steam Generator Tubes", NUREG/CR-6664, ANL-99/23, Argonne, IL, 2000.
26. G. T. Hahn, M. Sarrate, and A. R. Rosenfield, "Criteria for Crack Extension in Cylindrical Pressure Vessels", Int. J. Fracture Mech., **5**, No. 3, pp. 187-210, 1969.
27. F. Erdogan, "Ductile Failure Theories for Pressurized Pipes and Containers", Int. J. of Pres. Ves. & Piping, **4**, pp. 253-283, 1976.
28. V. Cherkassky and F. Mulier, Learning from Data, John Wiley & Sons, 1998.
29. G. Strang, Introduction to Applied Mathematics, Wellesley, MA: Wellesley-Cambridge Press, 1986.
30. R. Fletcher, Practical Methods of Optimization, John Wiley &Sons, New York, 1998.
31. W. Karush, "Minima of Functions of Several Variables with Inequalities as Side Constraints", Master's Thesis, Dept. of Mathematics, Univ. of Chicago, Chicago, 1939.
32. H. W. Kuhn and A. W. Tucker, "Nonlinear Programming", Proc. 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics, University of California Press., Berkeley, 1951.
33. A. J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression", NeuroCOLT2 Technical Report Series, NC2-TR-1998-030, 1998.
34. K. Kasza, S. Majumdar, J. Park, J. Franklin, Results from Pressure and Leak-Rate Testing of Laboratory-Degraded Steam Generator Tubes, NUREG/CR-6789, ANL-01/31, 2002.
35. U. Parrikar, Personal Communication, MSU, 2004.

A
MODIFIED SOFTWARE DESCRIPTION

The original ANTARES code has been modified significantly to satisfy software design requirements specified by the EPRI AutoAnalysis development team. Main modifications consist of the following:

- all MATLAB scripts have been converted to functions;
- all global variable have been removed;
- all GUI functions have been removed;
- algorithms are accessed through standardized interfaces;
- input and output format has been adapted for the standard AutoAnalysis format;
- parameter setup is performed through external functions.

The ANTARES detection and classification/characterization modules have been modified according to the specified requirements. Modified interfaces are described below.

A.1 Detection Interface

In addition to coding changes needed to satisfy general requirements, a prefiltering function has been added to the detection algorithm. A new function for automated region of interest extraction has also been developed. The new functions are **prefiltImage** and **getROIC,** presented in the flow diagram in Figure A-1. The detection algorithm is encapsulated in the interface **DoDetection**. The functions **setPrefiltParameters, setDetectionParameters**, and **setDetectionThresholds** have been 1developed for external definition of prefiltering and detection parameters. Detailed documentation for the functions comprising the detection interface is given below.

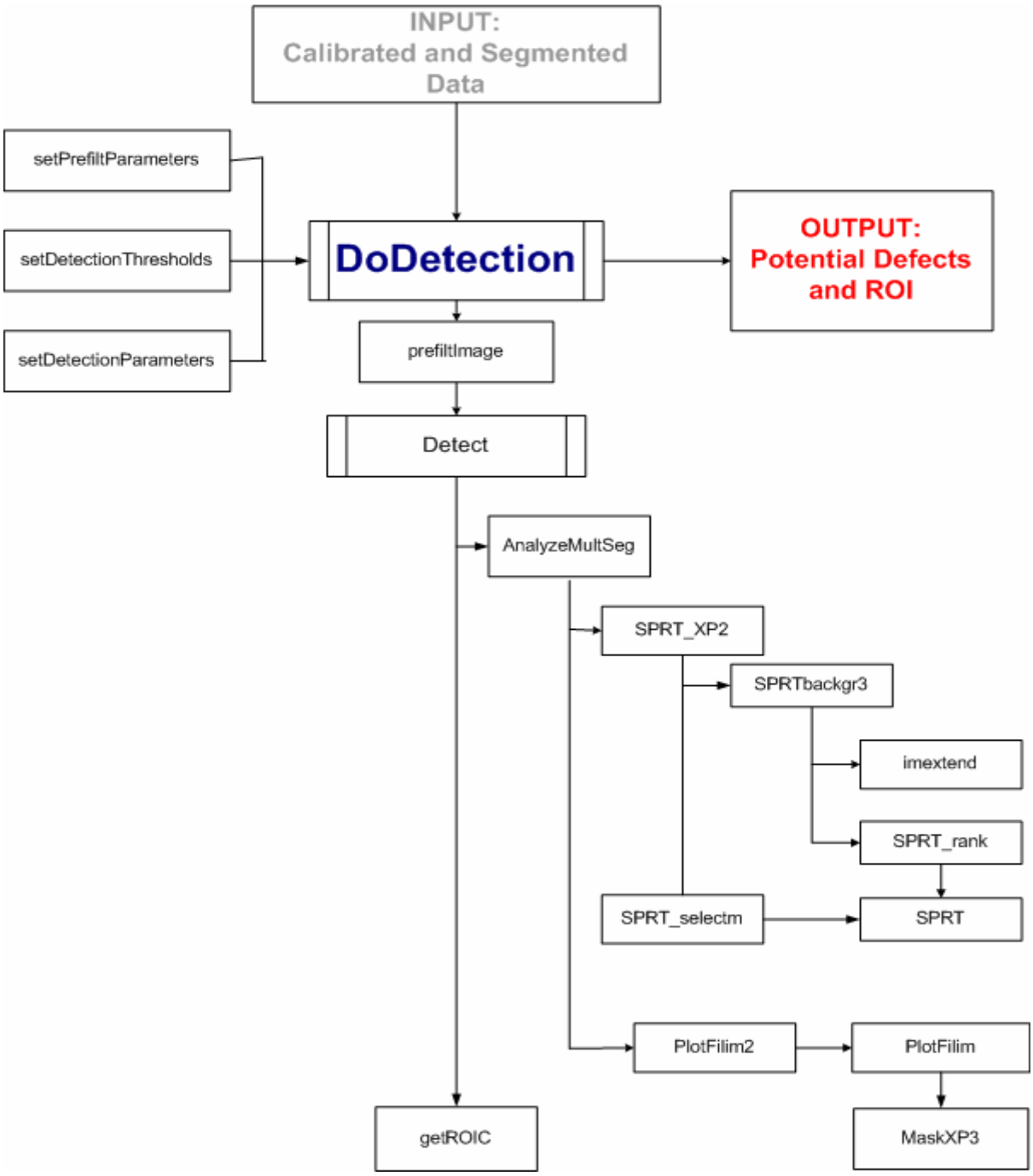

Figure A-1
Detection Interface Flow Diagram

A.1.1 Detection Interface Documentation

Module	**Detection**
Interface	**eiDetection**

nhood =setPrefiltParameters (nhood_val)

Description: This function assigns a sliding window neighborhood size for 2-D median filter.

Inputs: nhood_val – Integer vector with two elements to specify the filtering window along rows and columns, respectively. If set to -999 then internal value is used.

Outputs: nhood – Assigned neighborhood.

IMA0 = prefiltImage (IMA0, num_freq, nhood)

Description: This function performs median filter subtraction for the multidimensional image structure specified below.

Inputs: IMA0 – MATLAB 4-dimensional intensity image of the size (nrows, ncols, 1,2*num_freq) to be filtered. Tube images are concatenated in the following order: horizontal-AF1, vertical-AF1, horizontal-AF2…

num_freq – Integer. Number of frequencies.

nhood – Integer vector with two elements of the specified filtering window along rows and columns, respectively.

Outputs: IMA0 – Filtered image in the same format as input image.

[rangeBack, rangeSenDet, erosionSize, dilationSize, minsum]= setDetectionParameters (rangeBack_val, rangeSD_val, erosionSize_val dilationSize_val, minsum_val)

Description: This function assigns detection parameters.

Inputs: rangeBack_val – MATLAB cell array with two cells, one for each scan direction. Each cell is an integer vector [nrowb ncolb] for the sliding window used in the background estimation. If a cell in the cell array is set to -999 then internal values are used for the corresponding scan direction.

rangeSD_val – MATLAB cell array with two cells, one for each scan direction. Each cell is an integer vector [nrowsd ncolsd] for the sliding window used in the sensitive detection algorithm. If a cell in the cell array is set to -999 then internal values for the corresponding direction are used.

erosionSize_val – MATLAB cell array with two cells, one for each scan direction. Each element is an integer vector [nrowe ncole] for the erosion size window. 'Rectangle' structural element is assumed. If a cell in the cell array is set to -999 then internal values for the corresponding direction are used.

dilationSize_val – MATLAB cell array with two cells, one for each scan direction. Each cell is an integer for the dilation size. 'Diamond' structural element type is assumed. If a cell in the cell array is set to -999 then internal values for the corresponding direction are used.

minsum_val – Integer. The minimum number of pixels per segment to declare potential defects. If set to -999 then internal value is used.

Note: The default filtering range and morphology parameters defined internally are dependent on the number of axial channels (NAC). NAC =32 is assumed.

Outputs: rangeBack – Assigned background filtering widow.

rangeSenDet – Assigned filtering window for sensitive detection.

erosionSize – Assigned erosion size.

dilationSize – Assigned dilation size.

minsum – Assigned minimum number of pixel for defective segments.

FinThresh = setDetectionThresholds (FinThresh_val)

Description: This function assigns final detection thresholds to remove segments with small peaks. Final thresholds are dependent on the plant type, the segment type, and the scan direction.

Inputs: FinThresh_val – Real matrix of dimensions 5-by-2, for 5 segment types defined in the program and 2 scan directions. The first column is for the axial scans, the second column is for the circumferential scan. Segment types are:

1- free span; 2 –tube sheet, 3- TSP (full circumference);
4-TSP ("egg crate"); 5 –U-bend.

If any element of the matrix is set to -999 then internal value is used.

Outputs: FinThresh – Assigned threshold values.

[DAF1, DAF2, DAF3, DAF4, DCF1, DCF2, DCF3, DCF4, defsegA, defsegC, ROICTAx, NDAx, NtotAx, ROICTCir, NDCir, NtotCir, status, ercode, errmess]= DoDetection(AF1,AF2,AF3,AF4,CF1,CF2,CF3,CF4, cropStart, cropEnd, NSEG,segBounds, segType, preFiltOpt, preFiltNhood, status, num_freq, FinThresh, rangeBack, rangeSenDet, erosionSize, dilationSize, minsum)

Description: This function takes as input 8 matrices with calibrated or filtered axial and circumferential scan data in the standard format and transforms them into the specific format used in the ANL detection algorithm. It then call the detection algorithm and upon successful completion transform the processed data back to the standard format.

Inputs: AF1, AF2, ...CF3, CF4 – Data matrices in the standard input standard format for raw or filtered data. Each matrix is a floating point matrix having (2 * N) columns corresponding to horizontal and vertical values of all the channels in a particular frequency, while rows equal to number of data points in a channel.

cropStart – Integer. Stating point for the cropped region in the original tube coordinates.

cropEnd – Integer. Ending point for the cropped region in the original tube coordinates.

NSEG – Integer. Number of segments.

segBounds – Integer vector of segment boundary positions (NSEG +1) in the cropped tube coordinates systems (starts with 1).

segType – Integer vector of size NSEG. – segment type for each segment

1 – free span; 2 – tube sheet; 3 – TSP (full circumference);
4 – TSP (“egg crate”); 5 – U-bend.

preFiltOpt – Integer. Prefiltering option. preFiltOpt =1 performs prefiltering inside the detection algorithm, preFiltOpt =0 no prefiltering.

preFiltNhood – Integer vector [n1 n2]. Prefiltering neighborhood.

status – Integer. Program status on input. Status =0 – data in the initial format imported.

num_freq – Integer. Number of frequencies. Assumed num_freq =4.

FinThresh – Matrix of dimensions (5x2), for 5 possible segment types and 2 scan directions. The first column is for the axial scans, the second column is for the circumferential scan. Segment types are:

1- free span; 2- tube sheet, 3- TSP (full circumference);
4- TSP ("egg crate"); 5- U-bend.

rangeBack – MATLAB cell array with two elements, one for each scan direction. Each element is an integer vector [nrowb ncolb] for the sliding window used in the background estimation.

rangeSenDet – MATLAB cell array with two elements, one for each scan direction. Each element is an integer vector [nrowsd ncolsd] for the sliding window used in the sensitive detection algorithm.

erosionSize – MATLAB cell array with two elements, one for each scan direction. Each element is an integer vector [nrowe ncole] for the erosion size window. 'Rectangle' structural element is assumed.

dilationSize – MATLAB cell array with two elements, one for each scan direction. Each element is an integer for the dilation size. 'Diamond' structural element type is assumed.

minsum – Integer. The minimum number of pixels per segment to declare potential defects.

Outputs: Processed data DFA1, DFA2, DFA3, ... DCF3, DCF4, which include only possible defects. The format is the same as input.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects.

ROICTAx – Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDAx(iseg)-by-4, where NDAx(iseg) is the number of axial defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function: column-min, row-min, width, length. All coordinates are transformed to the full length data coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices and should be wrapped around boundary, if needed.

NtotAx – Integer. Total number of axial defects.

ROICTCir- Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDCir(iseg)-by-4, where NDCir(iseg) is the

number of circumferential defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function: column-min, row-min, width, length. All coordinates are transformed to the full length data coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices and should be wrapped around boundary, if needed.

NDCir – Integer vector of size NSEG with number of circumferential defects for each segment.

NtotCir – Integer. Total number of potential circumferential defects.

status – Integer. Program status. 1- internal image structure initialized; 2 – segmentation completed; 3 – detection completed

ercode – Integer. Error code. 1- successful completion; 2- incorrect input parameters; 3- errors in tube segmentation; 4 –error in detection

errmess – Cell array. Each cell is a string with an error message.

[filt_img, defsegA, defsegC, ROICTAx, NDAx, NtotAx, ROICTCir, NDCir, NtotCir,ercode] = Detect (segType, NSEG, seg_imgA, seg_imgC, num_freq, FinThresh, rangeBack, rangeSD, erosionSize, dilationSize, minsum))

Description: This function is the main driver for the detection algorithm. It calls the detection routines and upon successfully completed detection calls a function, which determines the coordinates of the potential defect regions.

Inputs: segType – Integer vector of size NSEG. Assigns a segment type to each segment. Segment types are: 1 – free span; 2 – tube sheet; 3 – TSP (full circumference); 4 – TSP (“egg crate”); 5 – U-bend.

NSEG – Integer. Number of segments.

seg_imgA – MATLAB cell array with NSEG cells with segmented axial scan images. Each cell is a MATLAB 4-dimensional intensity image of the size (nrows, ncols,1,2*num_freq).

seg_imgC – MATLAB cell array with NSEG cells with segmented circumferential scan images. Each cell is a MATLAB 4-dimensional intensity image of the size (nrows, ncols,1,2*num_freq).

num_freq – Integer. Number of frequencies.

FinThresh – Matrix of dimensions 5-by-2, for 5 possible segment types and 2 scan directions. The first column is for the axial scans, the second column is for the circumferential scan. Segment types are:

1- free span; 2- tube sheet, 3- TSP (full circumference);
4- TSP ("egg crate"); 5- U-bend.

rangeBack – MATLAB cell array with two elements, one for each scan direction. Each element is an integer vector [nrowb ncolb] for the sliding window used in the background estimation.

rangeSD – MATLAB cell array with two elements, one for each scan direction. Each element is an integer vector [nrowsd ncolsd] for the sliding window used in the sensitive detection algorithm.

erosionSize – MATLAB cell array with two elements, one for each scan direction. Each element is an integer vector [nrowe ncole] for the erosion size window. 'Rectangle' structural element is assumed.

dilationSize – MATLAB cell array with two elements, one for each scan direction. Each element is an integer for the dilation size. 'Diamond' structural element type is assumed.

minsum – Integer. The minimum number of pixels per segment to declare potential defects.

Outputs: filt_img – MATLAB cell array of processed images for each segment, consisting of NSEG-by-2 cells. First column holds processed 4-D images for axial scans, second column holds 4-D processed images for circumferential scans. Each cell is a MATLAB 4-dimensional intensity image of the size (nrows, ncols,1,2*num_freq).

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects.

OICTAx – Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDAx(iseg)-by-4, where NDAx(iseg) is the number of axial defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function: col-min, row-min, width, length. All coordinates are in the local segment coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices and should be wrapped around boundary, if needed.

NDAx – Integer vector of size NSEG with number of axial defects for each segment.

NtotAx – Integer. Total number of axial defects.

ROICTCir- Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDCir(iseg)-by-4, where NDCir(iseg) is the number of circumferential defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function: col-min, row-min, width, length. All coordinates are in the local segment coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices and should be wrapped around boundary, if needed.

NDCir – Integer vector of size NSEG with number of circumferential defects for each segment.

NtotCir – Integer. Total number of potential circ defects.

ercode – Integer. Error code. 1- successful completion.

[ROICTAx, NDAx, NtotAx, ROICTCir, NDCir, NtotCir,filt_img] = getROIC (filt_img,defsegA,defsegC,ImgThresh,segType)

Description: This function determines the coordinates of the potential defect regions and produces the final processed images by removing morphologically small potential defects.

Inputs: filt_img – MATLAB cell array of processed images for each segment, consisting of NSEG-by-2 cells. First column holds processed 4-D images for axial scans, second component holds 4-D processed images for circumferential scans. Each cell is a MATLAB 4-dimensional intensity image of the size (nrows, ncols,1,2*num_freq).

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects.

ImgThresh – Threshold for MATLAB img2bw function (intensity image to binary image conversion). In the current implementation, ImgThresh = FinThresh on input, but is internally reduced to min(0.01,0.1*FinThresh).

segType – Integer vector of size NSEG. – segment type for each segment

1 – free span; 2 – tube sheet; 3 – TSP (full circumference);
4 – TSP (“egg crate”); 5 – U-bend.

Outputs: ROICTAx – Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDAx(iseg)-by-4, where NDAx(iseg) is the number of axial defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function: column-min, row-min, width, length. All coordinates are in the local segment coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices. Should be wrapped around boundary, if needed.

NDAx – Integer vector of size NSEG with number of axial defects for each segment.

NtotAx – Integer. Total number of axial defects.

ROICTCir- Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDCir(iseg)-by-4, where NDCir(iseg) is the number of circumferential defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function: column-min, row-min, width, length. All coordinates are in the local segment coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices. Should be wrapped around boundary, if needed.

NDCir – Integer vector of size NSEG with number of circumferential defects for each segment.

NtotCir – Integer. Total number of potential circ defects.

filt_img – MATLAB cell array of processed images for each segment, consisting of NSEG-by-2 cells. First column holds processed 4-D images for axial scans, second component holds 4-D processed images for circumferential scans. Each cell is a MATLAB 4-dimensional intensity image of the size (nrows, ncols,1,2*num_freq). Updated processed image.

[filt_img, defsegA, defsegC] = AnalyzeMultSeg (im1, STYPE, NSEG, filt_img, defsegA, defsegC, segType, FinThresh, rangeBack, rangeSD, erosionSize, dilationSize, minsum)

Description: This function performs potential defect detection for a segmented image. The function is called separately for axial and circumferential scan directions. Processed image structure is updated after each function call.

Inputs: im1 – MATLAB cell array with NSEG cells. Each cell contains a 4-D image for the corresponding segment. Placeholder for axial and circumferential scan data evaluated in two calls of this function.

STYPE – Character. Scan type (‘A’ or ‘C’)

NSEG – Integer. Number of segments.

filt_img – MATLAB cell array of processed images for each segment, consisting of NSEGx2 cells. First column holds processed 4-D images for axial scans, second column holds 4-D processed images for circumferential scans. Previous values.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects. Previous values.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects. Previous values.

segType – Integer vector of size NSEG. – segment type for each segment

1 – free span; 2 – tube sheet; 3 – TSP (full circumference);
4 – TSP (“egg crate”); 5 – U-bend.

FinThresh – Matrix of dimensions 5-by-2, for 5 possible segment types and 2 scan directions.

rangeBack – MATLAB cell array with two cells, one for each scan direction. Each cell is an integer vector [nrowb ncolb] for the sliding window used in the background estimation.

rangeSD – MATLAB cell array with two cells, one for each scan direction. Each cell is an integer vector [nrowsd ncolsd] for the sliding window used in the sensitive detection algorithm.

erosionSize – MATLAB cell array with two cells, one for each scan direction. Each cells is an integer vector [nrowe ncole] for the erosion size window. ‘Rectangle’ structural element is assumed.

dilationSize – MATLAB cell array with two cells, one for each scan direction. Each cell is an integer for the dilation size. ‘Diamond’ structural element type is assumed.

minsum – Integer. The minimum number of pixels per segment to declare potential defects.

Outputs: filt_img – MATLAB cell array of processed images for each segment, consisting of NSEGx2 cells. First column holds processed 4-D images for axial scans, second column holds 4-D processed images for circumferential scans. Updated value.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects. Updated value.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects. Updated value.

[BINDI, POS] = SPRTbackgr3 (IMA, f0, alpha0, beta0, range)

Description: This function performs initial background assessment. 200 kHz vertical component is used for testing.

Inputs: IMA – MATLAB 4-D image of the size (nrow,ncols,1,2*num_freq)

alpha0 – Specified false alarm probability.

beta0 – Specified missed alarm probability.

f0 – Alternative hypothesis parameter (factor or SDM in **SPRT**)

range – Integer vector with 2 components. Rectangular filtering neighborhood.

Outputs: BINDI – MATLAB cell array with nr cells, where nr is the number of image rows. Each cell contains a list of background pixels for the corresponding row.

POS – Array with nr elements. Each element gives the number of background pixels for the corresponding row.

[defseg, TOTAL] = SPRT_XP2 (IMA0, fback, alpha0, beta0, rangeback, f0, MAG, IW, range, strela, minsum, defseg, imseg,TOTAL)

Description: This function performs defect detection in the specified image segment imseg. Uses 2 step SPRT algorithm for coarse and sensitive detection.

Inputs: IMA0 – MATLAB 4-D image, defined for each segment and scan type in the calling function.

fback – SPRT SDM factor for initial background estimation .

alpha0 – SPRT setup for false alarm probability.

beta0 – SPRT setup for missed alarm probability.

rangeback – Integer vector with 2 components for background filtering range.

f0 – SDM factor for fine detection.

MAG – Magnification factor for coarse detection.

IW – Integer for filter type indicator, IW=1 – Wiener filter, IW =2, median filter.

range – Integer vector for sensitive SPRT filtering range.

strela – MATLAB structure for erosion structural element.

minsum – Minimum number of nonzero pixels per segment to declare a defect.

defseg – Integer vector of segments with defects. Previous value.

imseg – Current segment index.

TOTAL – Cell array with alarm maps for segments with defects, previous value. Note that each alarm map is of the size ncolumns-by-nrows, due to transpositions in SPRT function. Alarms are actually Boolean complement, for display purposes. 0 represents a defect pixel.

Outputs: defseg – Updated vector of indices of segments with defects.

TOTAL – Cell array with alarm maps for segments with defects, update. Boolean complement, for display purposes. 0 is a defect pixel.

ALA = SPRT_rank (residuals, backgr, alpha0, beta0, f0, indi, ALA)

Description: This function defines SPRT test parameters for background estimation.

Inputs: residuals – Matrix ncols-by nrows. Note image transposition to adjust for the format of SPRT function calls.

backgr – Matrix ncols-by nrows. Transposed background image.

alpha0 – SPRT setup for false alarm probability.

beta0 – SPRT setup for missed alarm probability.

f0 – SDM factor for background detection.

indi – Indices of tested components in the multidimensional image. For this function, indi = 4 is defined in the calling function for background estimation

ALA – MATLAB uint8 image of size ncols-by-nrows for SPRT alarms. Previous alarm map.

Outputs: ALA – MATLAB uint8 image of size ncols-by-nrows for SPRT alarms. Updated alarm map.

ALA = SPRT_selectm (residuals, BINDI, alpha0, beta0, f0, ALA, indi)

Description: This is a function for sensitive SPRT test with variance smoothing for image residual data.

Inputs: residuals – Transposed image of the size ncols-by-nrows, adjusted for SPRT function calls.

BINDI – Cell array with nrows cells, where nrows is the number of image rows. Each cell contains a list of background pixels for the corresponding row.

alpha0 – SPRT setup for false alarm probability.

beta0 – SPRT setup for missed alarm probability.

f0 – SDM factor for fine detection.

ALA – MATLAB uint8 image of the size ncols-by-nrows for SPRT alarms. Previous alarm map.

indi – Tested components of multidimensional image.

In this function indi = 4,6,8. Positive SPRT used (only positive SPRT used for X-probes).

Outputs: ALA – MATLAB uint8 image of size ncols-by-nrows for SPRT alarms. Updated alarm map.

[ALARM1, ALARM2, SPRT1, SPRT2, ASN1, ASN2] = SPRT(residuals, alpha, beta, factor, var, mv_flag, p_flag)

Description: This is a general detection function, which detects either the change in the mean (mean SPRT) or variance (variance SPRT tests) on an array of residual signals. The function is used here with the positive mean test, which corresponds to SPRT1.

Inputs: residuals – M by N array of input signals. There are N signals, one signal to each column in the array.

alpha – N element vector with the false alarm probability for each signal.

beta – N element vector with the missed alarm probability for each signal.

factor – N element vector of multiplicative factors.

var – N element vector containing the variances of the signals.

mv_flag – Perform mean tests if mv_flag = ‘m’ or ‘M’. Perform variance tests if mv_flag = ‘v’ or ‘V’ (not used in this program) .

p_flag – Print/don’t print run progress.

Outputs: ALARM1 – M by N array of alarm values for SPRT test #1.

ALARM2 – M by N array of alarm values for SPRT test #2.

SPRT1 – M by N array of SPRT indices for SPRT test #1.

SPRT2 – M by N array of SPRT indices for SPRT test #2.

ASN1 – N element array containing the average sample numbers for SPRT test #1.

ASN2 – N element array containing the average sample numbers for SPRT test #2.

[filt_img, defsegA, defsegC] = PlotFilim(TOTAL, IMA0seg, nhood, defseg, STYPE, FinThresh, NSEG, filt_img, defsegA, defsegC,segType)

Description: Identifies segments with defects and calls function which performs final masking and filtering to produce images with defects only.

Inputs: TOTAL – MATLAB cell array with uint8 alarm maps for segments with defects, update. Boolean complement, for display purposes. 0 is the defect pixel.

IMA0seg – MATLAB cell array with NSEG cells for the segmented input image.

nhood – MATLAB structure for dilation neighborhood.

defseg – Working vector with indices of segments with defects.

STYPE – Character. Scan type (A or C).

FinThresh – Matrix of dimensions 5-by-2, for 5 segment types and 2 scan directions.

NSEG – Number of segments.

filt_img – MATLAB cell array of processed images for each segment, consisting of NSEG-by-2 cells. First column holds processed 4-D images for axial scans, second column holds 4-D processed images for circumferential scans. Previous value.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects. Previous value.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects. Previous value.

segType – Integer vector of size NSEG. Segment type for each segment.

Outputs: filt_img – MATLAB cell array of processed images for each segment, consisting of NSEG-by-2 cells. First column holds processed 4-D images for axial scans, second column holds 4-D processed images for circumferential scans. Update.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects. Update.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects. Update.

[filt_img, defsegA, defsegC] = PlotFilim2 (TOTAL, IMA0seg, nhood, defseg, STYPE, FinThresh, NSEG, filt_img, defsegA, defsegC, segType)

Description: Final thresholding, masking, and image plotting.

Inputs: TOTAL – MATLAB cell array with alarm maps for segments with defects. Boolean complement, for display purposes. 0 is a defect pixel.

IMA0seg – MATLAB cell array with NSEG cells. The segmented input image for the scan direction defined by STYPE.

nhood – MATLAB structure for dilation neighborhood.

defseg – Working vector with indices of segments with defects.

STYPE – Character. Scan type (A or C).

FinThresh – FinThresh – Matrix of dimensions 5-by-2, for 5 possible segment types and 2 scan directions.

NSEG – Number of segments.

filt_img – MATLAB cell array of processed images for each segment, consisting of NSEGx2 cells. First column holds processed 4-D images for axial scans, second column holds 4-D processed images for circumferential scans. Previous value.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects. Previous value.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects. Previous value.

segType – Integer vector of size NSEG. Segment type for each segment

Outputs: filt_img – MATLAB cell array of processed images for each segment, consisting of NSEGx2 cells. First column holds processed 4-D images for axial scans, second column holds 4-D processed images for circumferential scans. Update.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects. Update.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects. Update.

[IMA, has_defect] = MaskXP3 (ALA, IMA, nhood, STYPE, FinThresh_comp)

Description: Background and defect subtraction using SPRT mask, performed on individual segment images for each scan. Final masking and boundary effec fixes.

Inputs: ALA – MATLAB uint8 image of size nrows-by-ncols for SPRT alarms. Note that here the alarm map is transposed to correspond to the standard image format.

IMA – MATLAB 4-D image for specified segment and scan type. Previous value.

nhood – MATLAB structural element for mask dilation.

STYPE – Character. scan type (A or C).

FinThresh_comp – Component of the FinThresh matrix for specified segment and scan type.

Outputs: IMA – MATLAB 4-D image for specified segment and scan type. Update.

has_defect – Boolean defect indicator.

A.1.2 Auxiliary Functions

nelinfilter – The same as MATLAB function **nlfilter**, except that a GUI message is removed

mkconstarray – MATLAB library function, called from the modified unction **nelinfilter**

minimag – Finds minimum value of a matrix; used in **nelinfilter.**

imextend – Function which extends images along columns (circumferential direction) using periodic boundaries; m pixels are added at each side.

A.2 Classification Interface

New nonlinear regression parameters have been determined based on the 18 EDM notch standard inspected using modified X-probe frequencies. The modified parameters are stored in the MATLAB data files AXIDfitparu, AXODfitparu, CIRIDfitpatu, and CIRODfitparu.

In addition to coding modifications needed to satisfy general requirements, a new function for automated defect classification **ClassRules** has been developed. The classification algorithm is encapsulated in the interface **Classify** presented in Figure A-2. The function **setClassificationThresholds** has been developed for external definition of classification parameters. Detailed documentation for the functions comprising the classification interface is given below.

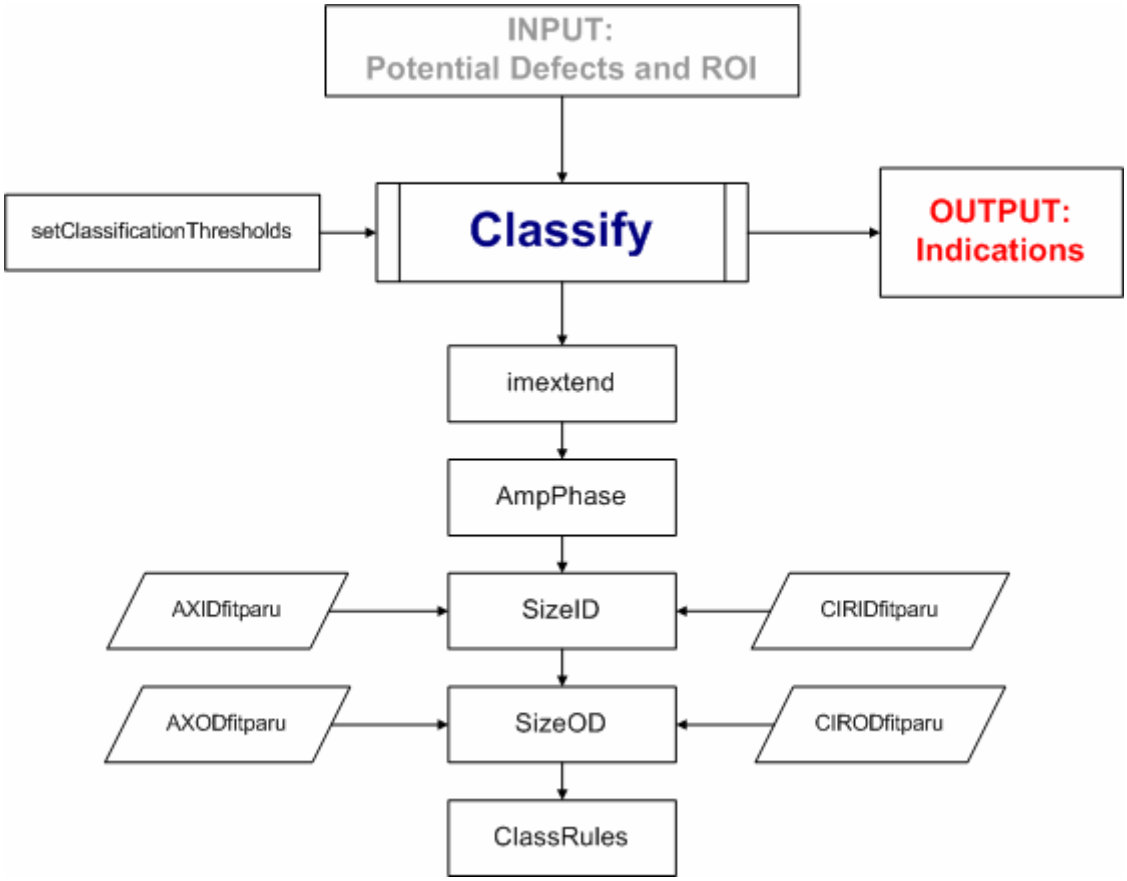

Figure A-2
Classification Interface Flow Diagram

A.2.1 Classification Interface Documentation

Module	**Classification**
Interface	**eiClassification**

[AmpThr, DepthThr] = setClassificationThresholds (AmpThr_val, DepthThr_val)

Description: This function assigns values for classification thresholds.

Inputs: AmpThr_val – Amplitude threshold (V). If maximum amplitude (for all

frequencies) is less than the specified threshold, degradation is declared non-detectable. Default value used if set to -999.

DepthThr_val – TW depth threshold (%). If estimated depth value is less than the specified threshold, degradation is declared non-detectable. Default value used if set to -999.

Outputs: AmpThr, DepthThr – Assigned threshold values.

[DAF1, DAF2, DAF3, DAF4, DCF1, DCF2, DCF3, DCF4, ROICTAx, NDAx, ROICTCir, NDCir, ercode, errmess] =Classify (DAF1, DAF2, DAF3, DAF4, DCF1, DCF2, DCF3, DCF4, num_freq, NSEG, defsegA, defsegC, segBounds, cropStart, cropEnd, ROICTAx, NDAx, ROICTCir, NDCir, AmpThr, DepthThr)

Description: This function classifies preliminary indications into defects and non-defects and reports indications which are classified as defects.

Inputs: DAF1,DAF2, ..., DCF3,DCF4 – Output from the detection algorithm, which contains preliminary indications. Data matrices are in the standard input format for filtered data. Each matrix is a floating point matrix having (2 * N) columns corresponding to horizontal and vertical values of all the channels in a particular frequency, while rows equal to the number of data points in a channel.

num_freq – Integer. Number of frequencies. Assumed num_freq =4.

NSEG – Integer. Number of segments.

defsegA – Integer vector of size NSEGDA with indices of segments with axial defects. NSEGDA – total number of segments with axial defects.

defsegC – Integer vector of size NSEGDC with indices of segments with circumferential defects. NSEGDC is the total number of segments with circumferential defects.

segBounds – Integer vector of segment boundary positions (NSEG +1) in the cropped tube coordinates systems (starts with 1).

cropStart – Integer. Stating point for the cropped region in the original tube coordinates.

cropEnd – Integer. Ending point for the cropped region in the original tube coordinates.

ROICTAx – Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDAx(iseg)-by-4, where NDAx(iseg) is the number of axial defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function: column-min, row-min, width, length. All coordinates are transformed to the full length data coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices and should be wrapped around boundary, if needed.

NDAx – Integer vector of size NSEG with number of axial defects for each segment.

ROICTCir- Cell array of size NSEG (number of segments) with region of interest coordinates for each segment. Each cell iseg contains the matrix of the region of interest coordinates of the size NDCir(iseg)-by-4, where NDCir(iseg) is the number of circumferential defects in the segment iseg. Region coordinates correspond to the image coordinate system as used in **imcrop** MATLAB function : column-min, row-min, width, length. All coordinates are transformed to the full length data coordinate system. Regions that cross the boundary are accounted for at the right boundary to prevent negative indices and should be wrapped around boundary, if needed.

NDCir – Integer vector of size NSEG with number of circumferential defects for each segment.

AmpThr – Amplitude threshold (V). If maximum amplitude (for all frequencies) is less than the specified threshold, degradation is declared non-detectable. Default value used if set to -999.

DepthThr – TW depth threshold (%). If estimated depth value is less than the specified threshold, degradation is declared non-detectable. Default value used if set to -999.

Outputs: DFA1, DFA2, DFA3, ... DCF3, DCF4 are updated matrices after classification, which include confirmed defects only. The format is the same as input.

ROICTAx – Cell array of size NSEG (number of segments) with region of interest coordinates for each segment (axial scan). Update after classification.

NDAx – Integer vector of size NSEG with number of axial defects for each segment. Update after classification.

ROICTCir – Cell array of size NSEG (number of segments) with region of interest coordinates for each segment (circumferential scan). Update after classification.

NDCir – Integer vector of size NSEG with number of circumferential defects for each segment. Update after classification.

ercode – Integer. Error code. 1- successful completion; 2- incorrect input parameters.

errmess – Cell array. Each cell is a string with an error message.

defect_indices = ClassRules (defect_list, NSEG, NDAx, NDCir, INDNZ, AmpThr, DepThr)

Description: This function uses a limited set of empirical rules based on the amplitude and phase behavior at different frequencies as well as approximate depth estimation to determine whether an indication is an ID defect, OD defect, or non-detectable degradation. Since multi-class discrimination is not very accurate, all classified defects are reported as one class. Conservative classification is adopted, i.e. all defects of an unspecified class are classified as defects. New rules could be added as the training base increases.

Inputs: defect_list – MATLAB cell array. Each cell is a MATLAB structure with the following fields:

class – ‘UN’-unknown (initialization), ‘ID’, ‘OD’, ‘ND’ (non-detectable).

mabs -Vector with peak amplitudes for all frequencies, starting with the lowest frequency.

mplo -Vector with peak phases for all frequencies, starting with the lowest frequency.

amps – Amplitude slope for increasing frequency (linear LS estimate).

angs – Phase angle slope for increasing frequency (linear LS estimate).

depth – Approximate TW depth assessment. Insufficient accuracy for reporting. Used only in rules based on sizing. depID and depOD – Temporary fields.

NSEG – Number of segments.

NDAx- NDAx(1:NSEG)vector with number of axial defects for each segment

NDCir – NDCir(1:NSEG)vector with number of circ defects for each segment.

INDNZ – Indices of frequencies with nonzero amplitudes.

AmpThr – Amplitude threshold (V). If maximum amplitude (for all frequencies) is less than the specified threshold, degradation is declared non-detectable.

DepThr – TW depth threshold (%). If estimated depth value is less than the specified threshold, degradation is declared non-detectable.

Outputs: defect_indices – MATLAB cell array of the size NSEG-by-2. Each cell contains a vector of indices of degradation classified as defects. Indices correspond to the numbering of the preliminary indications. Degradations declared as non-detectable are removed from the ROI list in the function Classify.

[mabs, mplo] = AmpPhase (IMA, num_freq, RECT)

Description: This function finds peak and computes amplitude and phase for each potential defect.

Inputs: IMA – MATLAB intensity image of the dimensions (nrows, ncols, 1, 2*num_freq)

num_freq – Number of frequencies.

RECT – ROI coordinates in the **imcrop** MATLAB function format: column-min, row-min, width, length. Coordinates are extended by few pixels to enable better evaluation.

Outputs: mabs – Vector with peak amplitudes for all frequencies, starting with the lowest frequency.

mplo – Vector with peak phases for all frequencies, starting with the lowest frequency.

[DEPRED, YP, INDNZ] = SIZEID (mabs, mplo, STYPE, AmpThr, DepThr)

Description: This function is used for preliminary maximum depth estimation for ID defects.

Inputs: mabs – Vector with peak amplitudes for all frequencies, starting with the lowest frequency.

mplo – Vector with peak phases for all frequencies, starting with the lowest frequency.

STYPE – Character. Scan type (‘A’ or ‘C’).

AmpThr – Amplitude threshold (V). If maximum amplitude (for all frequencies) is less than the specified threshold, degradation is declared non-detectable.

DepThr- TW depth threshold (%). If estimated depth value is less than the specified threshold, degradation is declared non-detectable.

Outputs: DEPRED – Maximum depth estimation.

YP – vector of estimations based on amplitude and phase

YP(1) – estimation based on amplitude at 200 kHz

YP(2) – estimation based on amplitude at 400 kHz

YP(3) – estimation based on phase at 200 kHz

YP(4) – estimation based on phase at 400 kHz

INDNZ – index of frequencies with nonzero amplitudes

[DEPRED, YP] = SIZEOD (mabs, mplo, STYPE, AmpThr, DepThr)

Description: This function is used for preliminary maximum depth estimation for OD defects.

Inputs: mabs -Vector with peak amplitudes for all frequencies, starting with the lowest frequency.

mplo -Vector with peak phases for all frequencies, starting with the lowest frequency.

STYPE – Character. Scan type (‘A’ or ‘C’).

AmpThr – Amplitude threshold (V). If maximum amplitude (for all frequencies) is less than the specified threshold, degradation is declared non-detectable.

DepThr- TW depth threshold (%). If estimated depth value is less than the specified threshold, degradation is declared non-detectable.

Outputs: DEPRED – Maximum depth estimation.

YP – Vector of estimations based on amplitude and phase.

YP(1) – Estimation based on amplitude at 200 kHz.

YP(2) – Estimation based on amplitude at 300 kHz.

YP(3) – Estimation based on phase at 200 kHz.

YP(4) – Estimation based on phase at 300 kHz.

INDNZ – Indices of frequencies with nonzero amplitudes.

A.2.2 Auxiliary Functions

imextend – Function which extends images along columns (circumferential direction) using periodic boundaries; m pixels are added at each side.

IDamfit – Analytical function to fit ID amplitudes.

ODamfit – Analytical function to fit OD amplitudes.

A.2.3 MATLAB Data Files

AXIDfitparu – Fitting parameters based on axial ID notches.

AXODfitparu– Fitting parameters based on axial OD notches.

CIRIDfitparu– Fitting parameters based on circumferential ID notches.

CIRODfitparu– Fitting parameters based on circumferential OD notches.

WARNING: This Document contains information classified under U.S. Export Control regulations as restricted from export outside the United States. You are under an obligation to ensure that you have a legal right to obtain access to this information and to ensure that you obtain an export license prior to any re-export of this information. Special restrictions apply to access by anyone that is not a United States citizen or a Permanent United States resident. For further information regarding your obligations, please see the information contained below in the section titled "Export Control Restrictions."

Export Control Restrictions
Access to and use of EPRI Intellectual Property is granted with the specific understanding and requirement that responsibility for ensuring full compliance with all applicable U.S. and foreign export laws and regulations is being undertaken by you and your company. This includes an obligation to ensure that any individual receiving access hereunder who is not a U.S. citizen or permanent U.S. resident is permitted access under applicable U.S. and foreign export laws and regulations. In the event you are uncertain whether you or your company may lawfully obtain access to this EPRI Intellectual Property, you acknowledge that it is your obligation to consult with your company's legal counsel to determine whether this access is lawful. Although EPRI may make available on a case by case basis an informal assessment of the applicable U.S. export classification for specific EPRI Intellectual Property, you and your company acknowledge that this assessment is solely for informational purposes and not for reliance purposes. You and your company acknowledge that it is still the obligation of you and your company to make your own assessment of the applicable U.S. export classification and ensure compliance accordingly. You and your company understand and acknowledge your obligations to make a prompt report to EPRI and the appropriate authorities regarding any access to or use of EPRI Intellectual Property hereunder that may be in violation of applicable U.S. or foreign export laws or regulations.

About EPRI

EPRI creates science and technology solutions for the global energy and energy services industry. U.S. electric utilities established the Electric Power Research Institute in 1973 as a nonprofit research consortium for the benefit of utility members, their customers, and society. Now known simply as EPRI, the company provides a wide range of innovative products and services to more than 1000 energy-related organizations in 40 countries. EPRI's multidisciplinary team of scientists and engineers draws on a worldwide network of technical and business expertise to help solve today's toughest energy and environmental problems.

EPRI. Electrify the World

Program: 1009540

Nuclear Power

© 2004 Electric Power Research Institute (EPRI), Inc. All rights reserved. Electric Power Research Institute and EPRI are registered service marks of the Electric Power Research Institute, Inc. EPRI. ELECTRIFY THE WORLD is a service mark of the Electric Power Research Institute, Inc.

Printed on recycled paper in the United States of America

EPRI • 3412 Hillview Avenue, Palo Alto, California 94304 • PO Box 10412, Palo Alto, California 94303 • USA
800.313.3774 • 650.855.2121 • askepri@epri.com • www.epri.com