2013

Intelligent Feature Selection Techniques for Pattern Classification of Time-Domain Signals

Corey Alexander Miller
College of William & Mary - Arts & Sciences

Recommended Citation

Miller, Corey Alexander, "Intelligent Feature Selection Techniques for Pattern Classification of Time-Domain Signals" (2013). *Dissertations, Theses, and Masters Projects.* Paper 1539623620.
https://dx.doi.org/doi:10.21220/s2-kg5s-e814

Intelligent Feature Selection Techniques for Pattern Classification of Time-Domain Signals

Corey Alexander Miller

Essex Junction, Vermont

B.S. Physics and Mathematics, The College of William and Mary, 2008

A Dissertation presented to the Graduate Faculty
of the College of William and Mary in Candidacy for the Degree of
Doctor of Philosophy

Department of Applied Science

The College of William and Mary
February 2013

APPROVAL PAGE

This Dissertation is submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

Corey Alexander Miller

Approved by the Committee, February 2013

Committee Chair
Professor Mark Hinders, Applied Science
The College of William and Mary

Assistant Professor Hannes Schniepp, Applied Science
The College of William and Mary

Associate Professor Sarah Day, Mathematics
The College of William and Mary

Dr. William Fehlman II
US Army Capabilities Integration Center

ABSTRACT

Time-domain signals form the basis of analysis for a variety of applications, including those involving variable conditions or physical changes that result in degraded signal quality. Typical approaches to signal analysis fail under these conditions, as these types of changes often lie outside the scope of the domain's basic analytic theory and are too complex for modeling. Sophisticated signal processing techniques are required as a result. In this work, we develop a robust signal analysis technique that is suitable for a wide variety of time-domain signal analysis applications. Statistical pattern classification routines are applied to problems of interest involving a physical change in the domain of the problem that translate into changes in the signal characteristics. The basis of this technique involves a signal transformation known as the Dynamic Wavelet Fingerprint, used to generate a feature space in addition to features related to the physical domain of the individual application. Feature selection techniques are explored that incorporate the context of the problem into the feature space reduction in an attempt to identify optimal representations of these data sets.

Contents

Acknowledgments

I would first and foremost like to thank my advisor, Professor Mark Hinders, for his guidance throughout this program. I could not have completed this work without his patience and many, many helpful conversations.

I would like to thank Prof. Hannes Schniepp, Prof. Sarah Day, and Dr. William Fehlman II for being on my defense committee and for taking the time to read my dissertation and provide feedback.

I would like to thank the faculty and staff in the Applied Science Department at the College of William and Mary for making my graduate school experience an enjoyable one. I would like to thank the Naval Research Lab, Valkyrie, Northrop Grumman, Airbus, and the Virginia Space Grant Consortium for funding. This work was performed [in part] using computational facilities at the College of William and Mary which were enabled by grants from Sun Microsystems, the National Science Foundation, and Virginia's Commonwealth Technology Research Fund.

I would like to thank Crystal Bertoncini and Bryan Nousain of the Naval Research Lab for their assistance with the RFID data collection and for their many helpful discussions. I would also like to thank Stephanie Severe for all of her hard work in collecting much of the Lamb wave damage characterization data. Thanks to former students Dr. Cara Leckey and Dr. Jill Bingham for their technical insight on a variety of topics related to my research. Additionally, many thanks are given to the current and former lab members for making my time here fun.

Finally, I would like to thank my friends and family for all of their love and support. I have wonderful friends that have helped me keep focused over the years, and I have a truly amazing family that I cannot thank enough for helping me become who I am today.

This dissertation is dedicated to my parents, David and Pamela Miller.

List of Tables

List of Figures

Chapter 1

Introduction

Time-domain signals are the basis of analysis in a wide variety of useful real-world applications. Often times there exist variable conditions within these applications that degrade signal quality, resulting in the features of interest being buried in noise. In this work we investigate the effect that physical changes within the domain of a problem have on the resulting signals. This is studied through the implementation of advanced signal processing techniques, developing an analysis routine that applies pattern classification to feature sets extracted from time-domain signals.

Typical approaches to signal analysis involve cross-correlation measures, amplitude thresholding, and baseline subtraction. As variation is introduced into the signals through variation in the signal's underlying physical mechanics, these approaches begin to fail. Often these types of changes lie outside the scope of the domain's basic analytic theory and are too complex for modeling. As a result, sophisticated signal processing techniques are required to identify the features of interest from the resulting degraded signals.

In this work we explore the development of a robust signal analysis technique, suitable for a wide variety of time-domain signal applications. Pattern classification is a subset of artificial intelligence that assigns labels to raw data based on multivariate statistical measures of that data set. Since these algorithms consider multi-dimensional feature spaces, many characteristics of a signal can be considered simultaneously in the decision process. The classification abilities are therefore directly related to the choice of feature representation for a data set. The *a priori* choice of optimal features is often unknown, and in practice, features are chosen based on intuition of the problem at hand. Feature selection techniques are designed to reduce the size of the feature space by identifying which subsets produce the best classification performance. This dissertation applies pattern classification techniques to a variety of applications involving a physical change in the data collection that translates to change in the signal characteristics. We explore feature selection techniques that incorporate the context of the problem into the feature space reduction.

1.1 Pattern Classification

Pattern classification is the subset of machine learning that involves taking in raw data and grouping it into categories. Many excellent textbooks exist on the subject [1–6], while several review papers have explored the topic as well [7–9]. Emerging applications in the fields of biology, medicine, financial forecasting, signal analysis, and database organization have resulted in the rapid growth of pattern classification algorithms. We focus our research here on statistical pattern recognition, but other approaches exist including template matching, structural classification, and neural networks [9].

Statistical pattern classification represent data by a series of measurements, forming a one-dimensional feature vector for each individual data point. A general approach for statistical pattern classification includes the following steps: preprocessing.

feature generation, feature extraction/selection, learning, and classification. Preprocessing involves any segmentation and normalization of data that leads to a compact representation of the pattern. Feature generation involves creating the feature vector from each individual pattern, while feature extraction and selection are optional steps that reduce the dimension of the feature vector using either linear transformations or the direct removal of redundant features. The learning step involves training a given classification algorithm, which then outputs a series of decision rules based on the data supplied. Finally, new data points are supplied to the trained classifier during the classification step, where they are categorized based on their own feature vector relative to the defined decision rules.

There exists a theorem in pattern classification known as the *Ugly Duckling Theorem*, which states that in the absence of assumptions there is no 'best' feature representation for a data set, since assumptions about what 'best' means are necessary for the choice of features [10]. Appropriate features for a given problem are usually unknown *a priori*, and as a result, many features are often generated without any knowledge of their relevancy [11]. This is frequently the motivation behind generating a large number of features in the first place. If a specific set of features is known that completely defines the problem and accurately represents the input patterns, then there is no need for any reduction in the feature space dimensionality. In practice, however, this is often not the case, and an intelligent feature reduction technique can simplify the classifiers that are built, resulting in both increased computational speed and reduced memory requirements. Aside from performance gains, there is a well-known phenomenon in pattern classification affectionately called the *Curse of Dimensionality* that occurs when the number of objects to be classified is small relative to the dimension of the feature vector. A generally accepted practice in classifier design is to use at least ten times as many training samples per class as the number of features [12]. The data set becomes sparse when represented in too high of dimension feature space, degrading classifier performance.

It follows that an intelligent reduction in the dimension of the feature space is needed. There are two general approaches to reducing the feature set: feature extraction and feature selection. Feature extraction reduces the feature set by creating new features through transformations and combinations of the original features. Principal component analysis, for example, is a commonly used feature extraction technique. Since we are interested in retaining the original physical interpretation of the feature set, we opt not to use any feature extraction techniques in our analysis.

There are three general approaches for feature selection: wrapper methods, embedded methods, and filter methods [13]. Wrapper methods use formal classification to rank individual feature space subsets, applying an iterative search procedure that trains and tests a classifier using different feature subsets for accuracy comparison. This continues until a given stopping criterion is met [14]. This approach is computationally intensive, and there is often a trade-off among algorithms between computation speed and the quality of results that are produced [15–17]. Additionally, these methods have a tendency to over-train themselves, where data in the training set is perfectly fitted and results in poor generalization performance [18]. Similar to wrapper methods, an embedded method performs feature selection while constructing the classification algorithm itself. The difference is that the feature search is intelligently guided by the learning process itself. Filter methods perform their feature ranking by looking at intrinsic properties of the data without the input of a formal classification algorithm. Traditionally, these methods are univariate and therefore don't account for multi-feature dependencies.

Regardless of feature selection, once a feature set has been finalized, the selection of an appropriate classification algorithm depends heavily on what is known about the domain of the problem. If the class-conditional densities for the problem at hand are known, then Bayes decision theory can be applied directly to design a classifier. This, however, is rarely the case for experimental data. If the training data set has known labels associated with it, then the problem is one of supervised learning. If not, then the underlying structure of the data set is analyzed through unsupervised classification techniques such as cluster analysis. Within supervised learning, a further dichotomy exists based on whether the *form* of the class conditional densities is known. If it is known, parametric approaches such as Bayes plug-in classifiers can be developed that estimate missing parameters based on the training data set. If the form of the densities is unknown, nonparametric approaches must be used, often constructing the decision boundaries geometrically from the training data.

1.1.1 Feature Generation

In this dissertation we include a variety of methods for feature generation. First order statistics are commonly used as features for pattern classification. Lambrou et al. use statistical measures of audio signals for genre classification by identifying statistical characteristics common among like-genres of music [19]. We include several common first-order statistical measures from each time-domain signal in many of our feature sets, including the mean of the raw waveform, the variance, the Shannon entropy, the second central moment, the skewness, and the kurtosis [20, 21]. Additionally, we include features extracted from the Mellin domain of each signal. These are generated by applying a scale-invariant integral transform known as the Mellin transform. Mellin transform features have been successfully used to identify defects in pipes under varying pressure loads [22, 23], and are useful in identifying uniform time-scaling effect on the recorded signals.

Wavelets are very useful for analyzing time series data because wavelet transforms allow us to keep track of the time localization of frequency components [24]. Unlike the Fourier transform, which breaks a signal down into sine and cosine components to identify frequency content, the wavelet transform measures local frequency features in the time domain. One direct advantage the wavelet transform has over the fast Fourier transform is that the time information of signal features can be taken directly from the transformed space without an inverse transform required. One common wavelet-based feature used in classification is generated by Wavelet Packet Decomposition (WPD) [25]. Yen and Lin successfully classify faults in a helicopter gearbox using WPD features generated from time-domain vibration analysis signals [26]. The use of wavelet analysis for feature extraction has also been explored by Jin et al. [27], where wavelet-based features have been used for damage detection in polycrystalline alloys. Gaul and Hurlebaus also used wavelet transforms to identify the location of impacts on plate structures [28].

Wavelets were first applied to ultrasonic signal analysis by Abbate, who was able to perform peak detection on signals dominated by added white noise through the application of a wavelet transform [29]. Since then, wavelets have been used commonly in the field of structural health monitoring. Sohn et al. review the statistical pattern classification models currently being used in structural health monitoring [30]. Most implementations involve identifying one specific type of 'flaw', including loose bolts and small notches, and utilize only a few specific features to separate the individual flaw classes within the feature space [31–33]. Biemans et al. detected crack growth in an aluminum plate using wavelet coefficient analysis generated from guided waves [34]. Legendre et al. found that even noisy electromagnetic acoustic transducer sensor signals were resolvable using the multi-scale method of the wavelet transform [35]. The wavelet transform has also been shown to outperform other traditional time-frequency representations in many applications. Zou and Chen compared the wavelet transform to the Wigner-Ville distribution for identifying a cracked rotor through changes in stiffness [36], identifying the wavelet transform to be more sensitive to variation and generally superior to the Wigner-Ville distribution.

We employ a particular implementation of the wavelet transform known as the Dynamic Wavelet Fingerprinting (DWFP) technique [37]. This approach renders time-domain data in a two-dimensional time-scale binary image, from which subtle signal characteristics can be identified and tracked through standard image processing techniques. This technique has been applied to a wide variety of applications with success, including acoustic microscopy [38], a periodontal probing device [39–41], time-domain reflectometry for wiring flaw detection [42], the analysis of multi-mode Lamb wave signals [43,44], and Lamb wave tomography [45,46].

1.2 Computational Homology

The algorithms developed within the field of computational homology provide insight into the topological makeup of a space. Combining algebra, combinatorics, and topology, these algorithms are fast and efficient at computing global properties of spaces, specifically a measure of connectedness. Several variations of computational homology exist, including those based on both simplical and cubical complexes. Most experimental data translate well to cubical representations (image pixels, error bounds, etc.), and as a result, many applications of computational homology involve those algorithms based on cubical sets. Computational homology has been previously implemented in a variety of analysis applications. Miller et al. generated isosurface reconstructions of the atomic structure of thermally treated alloys, identifying a correlation between the number of cavities in each surface and the amount of thermal aging undergone by the alloy [47]. More recent implementations span a range of applications including image processing and recognition [48–51], sensor network analysis [52], materials science [53], biomedical analysis [54], and image-based pattern classification [55,56]. The majority of current applications for computational homology theory in the sciences revolves around the analysis of 1-, 2-, or 3-dimensional spaces, most commonly in image analysis.

1.3 Applications

In this dissertation we explore the use of pattern classification routines in real-world applications using wavelet transforms. We then develop feature selection routines based on the underlying structure of a data set's feature space representation. In Chapter 2 we provide a formal introduction to pattern classification. Chapter 3 summarizes the Dynamic Wavelet Fingerprint (DWFP) technique which uses a time-scale wavelet transform to generate two-dimensional binary image representations of time-domain signals. We demonstrate the advantages of this technique through an example in structural health monitoring, where different types of flaws in metals are identified using guided waves. In Chapter 4 we describe an analysis that identifies individual radio-frequency identification (RFID) tags from cloned copies using pattern classification routines. Several sources of variability inherent to RFID usage are discussed and their effects studied, including tag/antenna orientation, communication frequency fluctuation, and environmental damage. A statistical pattern classification routine designed to characterize flaw severity in aluminum plates using Lamb waves is presented in Chapter 5. The concept behind intelligent feature selection is also introduced and explored. Finally, Chapter 6 describes the implementation of a novel feature selection routine involving algorithms from the field of computational homology. Chapter 7 provides conclusions and suggestions for future work.

Chapter 2

Pattern Classification

In the subsequent chapters, we employ statistical pattern classification to analyze complex data sets. This chapter provides an introduction to the mathematics behind the algorithms as well as terminology from the field of pattern classification used throughout this work.

2.1 White Wine Example

As an example data set, we consider the multivariate *white wine* data set [57] composed of results from a chemical analysis of many wines grown in the same region in Italy, but derived from three different grape cultivars. The goal of a classifier developed on this data set is to be able to identify the cultivar of unidentified wines based solely on their chemical makeup.

The white wine data set contains data from 178 different wines (*observations*), originating from three different grape cultivars (*classes*). From each wine sample, thirteen characteristics (*features*) were recorded. The measurements include alcohol concentration, malic acid concentration, ash, alkalinity of the ash, magnesium concentration, total phenols concentration, flavanoids concentration, nonflavanoid phenols concentration, proanthocyanine concentration, color intensity, color hue, OD280/OD315 of diluted wines (the ratio between absorbency at 280 and 315 nm), and proline concentration. Table 2.1 shows a limited selection from the white wine data set.

Table 2.1: The matrix arrangement of the white wine data set. Each the row of the matrix represents a wine sample, while each column represents one of the thirteen variables.

	Alcohol	Malic Acid	Ash	$\cdots$	Proline	Cultivar
Sample 1	14.23	1.71	2.43	$\cdots$	1065	1
Sample 2	13.20	1.78	2.14	$\cdots$	1050	1
Sample 3	13.16	2.36	2.67	$\cdots$	1185	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Sample 59	13.72	1.43	2.50	$\cdots$	1285	1
Sample 60	12.37	0.94	1.36	$\cdots$	520	2
Sample 61	12.33	1.10	2.28	$\cdots$	680	2
Sample 62	13.64	1.36	2.02	$\cdots$	450	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Sample 130	12.04	4.30	2.38	$\cdots$	580	2
Sample 131	12.86	1.35	2.32	$\cdots$	630	3
Sample 132	12.88	2.99	2.40	$\cdots$	530	3
Sample 133	12.81	2.31	2.40	$\cdots$	560	3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
Sample 178	14.13	4.10	2.74	$\cdots$	560	3

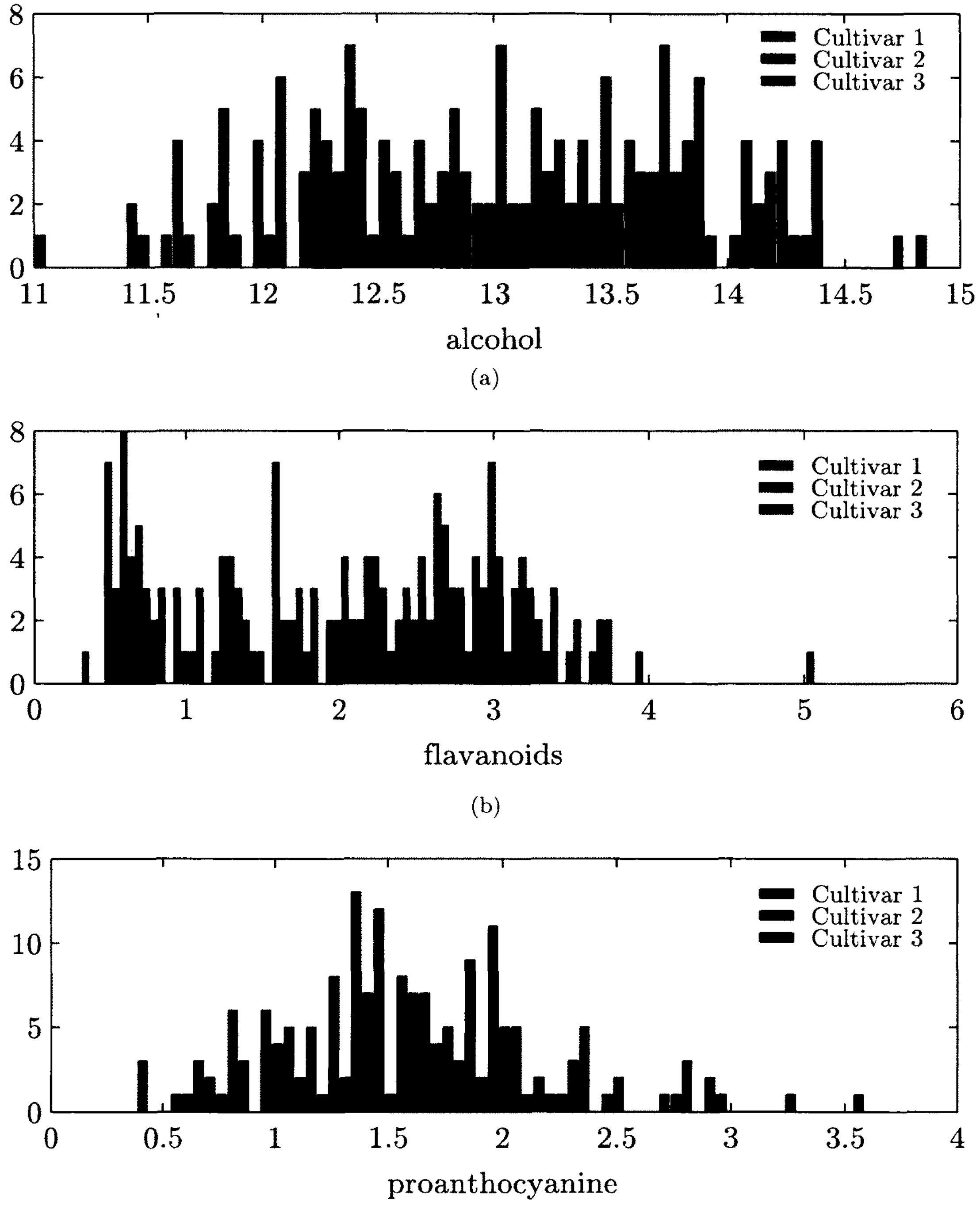
The data matrix in Table 2.1 can then be thought of as a representation of the wine samples in a multi-dimensional space (*feature space*), where each row is a single point in a thirteen-dimensional space and each variable represents an axis in this space. Along these terms, the coordinates of each point in this space can be thought of as that observation's feature vector, where each entry in the feature vector is a single measurement value. Since each wine sample is from one of three cultivars, each sample is given its own class label of cultivar 1, 2 or 3.

Figure 2.1 shows histograms of several of the measurements divided among the three known cultivars. The histograms show that there is significant overlap of proanthocyanine values between the three cultivars. The alcohol concentration values separate the wine samples between grape cultivar 1 and 2, however the samples from cultivar 3 overlap both of these. The flavanoid concentration values similarly do a good job separating samples from cultivar 1 and 3, however those from cultivar 2 are mixed into both other groups. There is no single feature in this data set that completely separates the three cultivars. Similarly, we have no reason to believe that this data set is even representative of the entire population of these three grape cultivars. We therefore require more advanced analysis techniques in order to identify these wines by their chemical makeup, specifically those of statistical pattern classification.

2.2 Statistical Pattern Classification

In short, statistical pattern classification uses statistical measures to learn from a training set of examples. A set of rules are generated, usually defined by a series of decision boundaries within a multi-dimensional feature space, that are then applied to new, unseen observations. This process is not well defined without a sense of context for the problem at hand. For example, if a computer were simply given the white wine data set introduced above, it would have no reason to assume the goal is to identify is the grape cultivar. This concept is summarized by the *ugly duckling theorem* [58], which states that in the absence of assumptions there is no 'best' feature representation for a data set, since assumptions about what 'best' means are necessary for the choice of features. Appropriate features for a given problem are usually unknown *a priori*, and as a result many features are often generated without any true knowledge of their relevancy.

In designing a pattern classification routine, one of two general routes can be taken: supervised classification in which predefined classes are used to identify new patterns,



(a)

(b)

(c)

Figure 2.1: Bar graphs of the wine measurements.

or unsupervised classification where the data's natural structure is used to determine class values for which new patterns are assigned. The choice of which approach to take is usually subjective and depends on the application and the structure of the data at hand. Most of our applications involve data that corresponds to a variety of well-defined classes, making supervised pattern classification a natural choice. Since the goal of a classifier is to determine labels for new, unknown data vectors, we first assign a unique label to each known class, ω_j for $j = 1, \ldots, 3$, for each of the three grape cultivars.

Within supervised pattern classification, there exist several general approaches for applying pattern classification routines. One approach involves template matching, where a similarity measure is used to associate new patterns with a predefined library of templates that is learned from the training set. This often involves matching general properties, such as curves or shapes within the data set. We opt not to use template matching as it tends to fail when large intraclass variations are present among the patterns [9]. Another popular approach uses neural networks, which create weighted directional graphs between artificial neurons. A major disadvantage of neural networks is that they are "black box" in nature, often creating overly complex nonlinear inout-output relationships that conceal the physical interpretation of the data. We choose not to use neural networks because of this loss of physical interpretation. A third approach is that of statistical pattern classification, where each pattern is represented in a multi-dimensional feature space where each class ideally occupies a disjoint region. The final goal in this approach is to determine decision boundaries in this feature space relative to the classes, from which new patterns are assigned labels based on their location in the divided space. Additionally, this approach retains the physical interpretation of the features. Statistical pattern classification is therefore an appropriate apprach to take for our applications. In statistical pattern classification, each pattern is represented as a point in a multi-dimensional feature space (Table 2.1).

A further division of statistical pattern classification lies in whether or not the form of the class-conditional densities is known. Parametric techniques assume a known density function (e.g. Gaussian) for each object class, and in practice replace any unknown parameters in that density function with estimated values. Similar to the ugly duckling theorem, there exists the *no free lunch theorem* which states that in the absence of assumptions about the learning domain, no single classification or learning algorithm should be preferred over another. Most of our experimental data does not have any underlying density function associated with it, however we often include both parametric and nonparametric density estimation classifiers in our analysis, keeping to this general rule.

2.3 Bayes Decision Theory

We next present an introduction to Bayes decision theory, followed by the fundamental concepts behind several of the classifiers used throughout this work. This description is adapted from [1,5].

For a given data set, each $n = 1, \ldots, N$ observation has M-many feature values $x_{n,j}$ for $j = 1, \ldots, M$. Together, these feature values make up a single M-dimensional feature vector $\mathbf{x}_n = \{x_{n,1}, \ldots, x_{n,M}\}$. The overall goal of pattern classification is thus: given a set of measurements collected through an observation, i.e. feature vector $\mathbf{x}$ (dropping subscript n for generality), we wish to assign that observation to one of C possible classes ω_j, where $j = 1, \ldots, C$. For the white wine data set, we have $N = 178$, $M = 13$, and $C = 3$ corresponding to the 178 individual wine samples collected, each of which was used to measure 13 properties of the wine. from the 3 different grape cultivars. Let $P(\omega_j)$ represent the *a priori* probability of class ω_j. In practice, the prior probabilities are usually unknown and must be estimated from a set of correctly classified samples. For example, we can use the population distribution of the three classes in the white wine example,

$$P(\omega_1) = \left(\frac{59}{178}\right) = 0.332, \quad P(\omega_2) = \left(\frac{71}{178}\right) = 0.399, \quad P(\omega_3) = \left(\frac{48}{178}\right) = 0.270.$$

Since we have both an observation as well as a feature vector, we wish to assign $\mathbf{x}$ to class ω_j if the probability of class ω_j given the observation $\mathbf{x}$, denoted $p(\omega_j|\mathbf{x})$, is greatest over all classes $\omega_1, \ldots, \omega_C$,

$$p(\omega_j|\mathbf{x}) > p(\omega_k|\mathbf{x}) \quad k = 1, \ldots, C; \quad k \neq j, \tag{2.1}$$

effectively splitting the feature space into C regions.

Bayes' theorem allows us to express the *a posteriori* probabilities $P(\omega_j|\mathbf{x})$ in terms of the *a priori* probabilities and the class-conditional density functions $p(\mathbf{x}|\omega_j)$. The probability that the object belongs to class ω_j given the feature vector $\mathbf{x}$ is:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j)P(\omega_j)}{\sum_{j=1}^{C} p(\mathbf{x}|\omega_j)P(\omega_j)}. \tag{2.2}$$

The parametric subset of pattern classification assumes a knowledge of the class-conditional probability density functions. If this assumption is violated, by using bad density estimates for example, the classifier may be far from optimal. If there is no known theoretical probability density function for the features, then nonparametric approachs can be used. For example, the experimental distribution of the features in each class may be used (such as in Figure 2.1 where the probabilities sum to unity).

Bayes' theorem allows us to rewrite the decision rule in Equation 2.1. We assign

$\mathbf{x}$ to ω_k if

$$p(\mathbf{x}|\omega_j)P(\omega_j) > p(\mathbf{x}|\omega_k)P(\omega_k) \quad k = 1, \ldots, C; \quad k \neq j. \tag{2.3}$$

Equation 2.3 allows a decision to be made on the feature vector under consideration given its measured features. We now present several classifiers, both parametric and nonparametric, that are used in the following chapters.

Quadratic and Linear Discriminant Classifiers

As the probabilistic structure of a given problem is rarely known for real applications, a solution involves estimating the unknown probabilities and probability densities. In supervised pattern classification, estimating the prior probabilities $P(\omega_j)$ is straightforward to do (using the known sample distributions), however accurate estimation of the class-conditional densities $p(\mathbf{x}|\omega_j)$ is less straightforward. If it is assumed that $p(\mathbf{x}|\omega_j)$ is a normal (Gaussian) density with mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$, the problem simplifies from one of estimating an unknown function $p(\mathbf{x}|\omega_j)$ to one of estimating the parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$.

One way of representing a pattern classifier is in terms of a set of discriminant functions, $g_j(\mathbf{x})$, for $j = 1, \ldots, C$. The classifier is said to assign a feature vector $\mathbf{x}$ to class ω_j if

$$g_j(\mathbf{x}) > g_k(\mathbf{x}) \quad \text{when} \quad k \neq j. \tag{2.4}$$

Using this, Equation 2.3 can be reproduced by allowing $g_j(\mathbf{x}) = p(\mathbf{x}|\omega_j)P(\omega_j)$. By assuming a Gaussian distribution for $p(\mathbf{x}|\omega_j)$, a discriminant function $g_j(\mathbf{x})$ can be derived that assigns $\mathbf{x}$ to ω_j if $g_j(\mathbf{x}) > g_k(\mathbf{x})$ for all $j \neq k$ [5], where

$$g_j(\mathbf{x}) = \log(P(\omega_j)) - \frac{1}{2}\log(|\boldsymbol{\Sigma}_j|) - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j). \tag{2.5}$$

The estimative approach replaces the quantities $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ with estimates based on a training set, given by

$$\begin{aligned} \mathbf{m} &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}\mathbf{x}_i \\ \hat{\boldsymbol{\Sigma}} &= \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T. \end{aligned} \tag{2.6}$$

Plugging these estimates into Equation 2.5 results in the *quadratic discriminant classifier* (QDC), given by

$$g_j(\mathbf{x}) = \log(P(\omega_j)) - \frac{1}{2}\log(|\hat{\boldsymbol{\Sigma}}_j|) - \frac{1}{2}(\mathbf{x} - \mathbf{m}_j)^T\hat{\boldsymbol{\Sigma}}_j^{-1}(\mathbf{x} - \mathbf{m}_j). \tag{2.7}$$

This form of classifier development is often referred to as Bayes plug-in method. Figure 2.2 shows the resulting decision boundaries defined by the quadratic discriminant classifier using a two-dimensional feature subset of the white wine data set, with alcohol and malic acid as the two features considered.

Problems can occur in this analysis if any of the $\hat{\boldsymbol{\Sigma}}_j$ matrices are singular, as these matrices are inverted in the computation of the QDC. Several alternatives are employed, one of which assumes that the class covariance matrices $\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_C$ are all the same. This allows the class covariance matrices to be substituted with the common group covariance matrix $\mathbf{S}_W$, resulting in the *linear discriminant classifier* (LDC):

$$g_j(\mathbf{x}) = \log(P(\omega_j)) - \frac{1}{2}\mathbf{m}_j^T\mathbf{S}_W^{-1}\mathbf{m}_j + \mathbf{x}^T\mathbf{S}_W^{-1}\mathbf{m}_j \tag{2.8}$$

where $\mathbf{S}_W$ is given by

$$\mathbf{S}_W = \frac{N}{N-C}\sum_{j=1}^{C}\frac{N_j}{N}\hat{\boldsymbol{\Sigma}}_j \tag{2.9}$$

and N_j is the number of observations in class ω_j. Figure 2.3 shows the resulting decision boundaries defined by the linear discriminant classifier using a two-dimensional

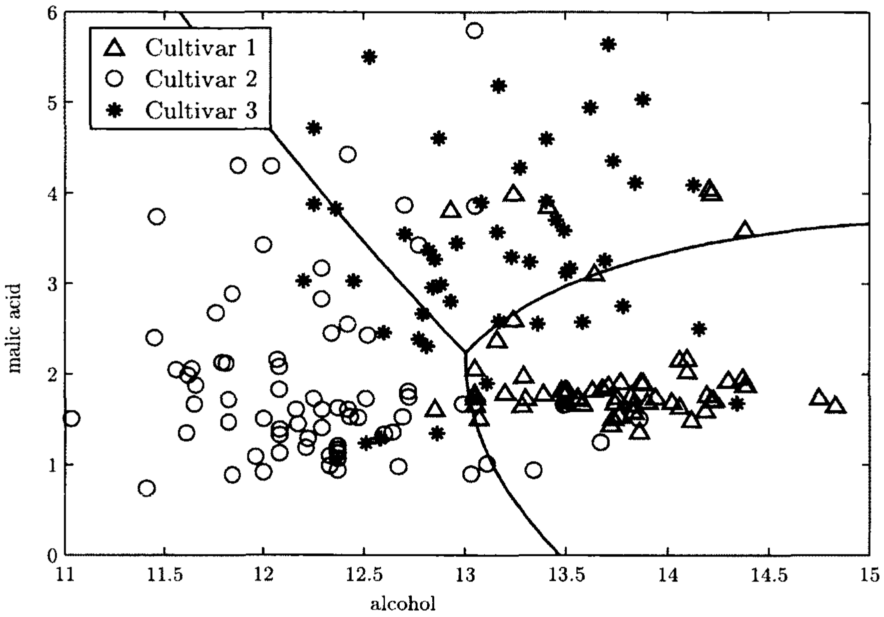


Figure 2.2: Decision boundary generated by the quadratic discriminant classifier (QDC) for a two-dimensional feature subset of the white wine data set, with alcohol and malic acid as the two features considered.

feature subset of the white wine data set, with alcohol and malic acid as the two features considered.

Near Mean Classifier

A special case of Equation 2.8 occurs when class priors $P(\omega_j)$ are equal for all C classes, and the matrix $\mathbf{S}_W$ is taken to be the identity. In this case, known as the *near mean classifier*, we assign $\mathbf{x}$ to class ω_j if

$$-2\mathbf{x}^T\mathbf{m}_k + \mathbf{m}_k^T\mathbf{m}_k > -2\mathbf{x}^T\mathbf{m}_j + \mathbf{m}_j^T\mathbf{m}_j \quad \text{when} \quad j \neq k. \tag{2.10}$$

In other words, to classify a feature vector $\mathbf{x}$, measure the Euclidean distance $\|\mathbf{x}-\boldsymbol{\mu}_j\|$ from each $\mathbf{x}$ to each of the C mean vectors $\boldsymbol{\mu}_j$, and assign $\mathbf{x}$ to the class of the nearest

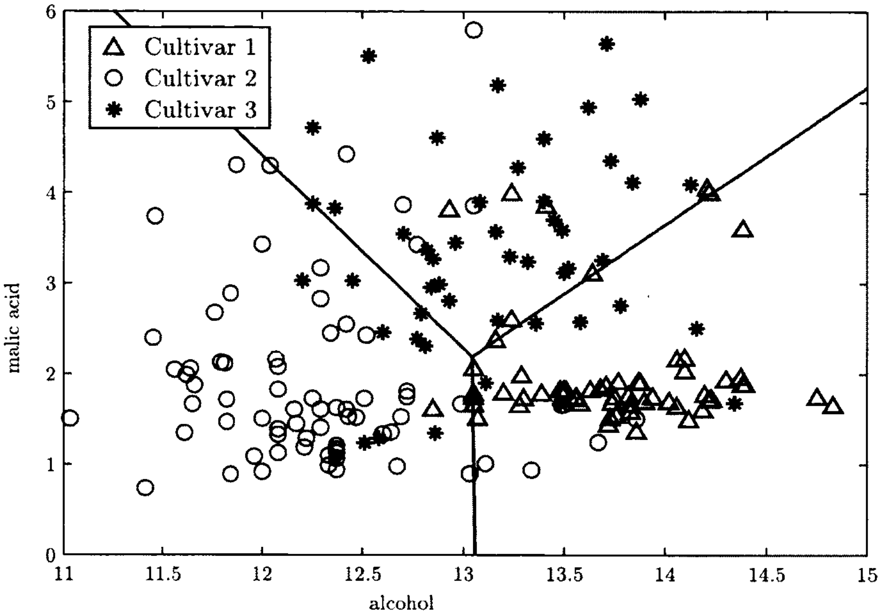


Figure 2.3: Decision boundary generated by the linear discriminant classifier (LDC) for a two-dimensional feature subset of the white wine data set, with alcohol and malic acid as the two features considered.

mean. Figure 2.4 shows the resulting decision boundaries defined by the near mean classifier using a two-dimensional feature subset of the white wine data set, with alcohol and malic acid as the two features considered.

k-Nearest-Neighbor

Nonparametric classifiers do not assume a form for the class-conditional probability densities. In practice, the common parametric forms rarely fit the densities actually encountered. Nonparametric classifiers can be used with arbitrary distributions, without assuming the form of any underlying densities. One particular density estimation technique can be defined by the k-nearest neighbor (kNN) approximation,

$$\hat{p}(\mathbf{x}) = \frac{k}{NV}, \tag{2.11}$$

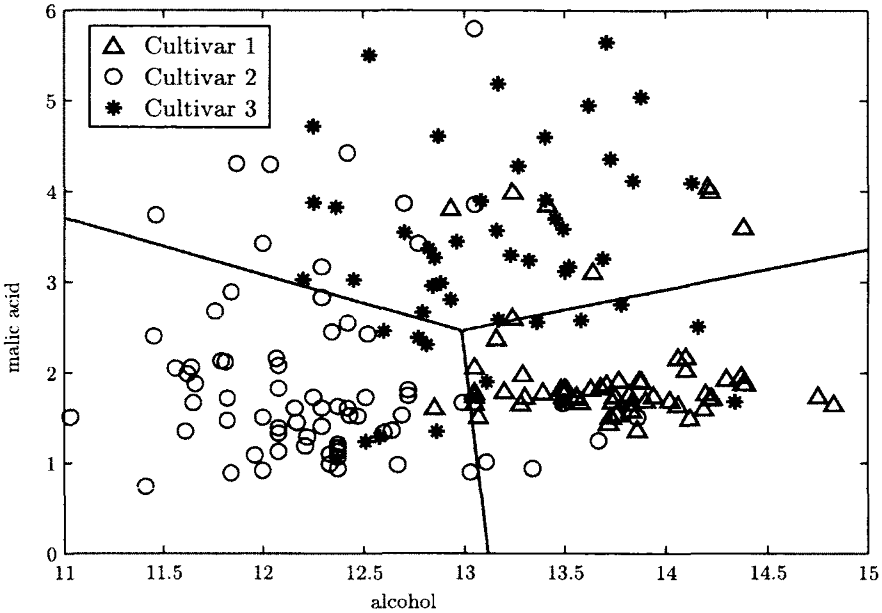


Figure 2.4: Decision boundary generated by the near mean classifier (NMC) for a two-dimensional feature subset of the white wine data set, with alcohol and malic acid as the two features considered.

where $k \in \mathbf{Z}$ is the number of samples, out of N total samples, that fall within a volume V centered at $\mathbf{x}$. If $\mathbf{x}_k$ is the kth nearest-neighbor point to $\mathbf{x}$, then V is the volume of a hyper-sphere centered at $\mathbf{x}$ of radius $\|\mathbf{x} - \mathbf{x}_k\|$.

We can relate this density estimate back to a decision rule. If we let k_j be the first k samples in class ω_j, and let N_j be the total number of samples in class ω_j (out of N total samples), then we can estimate the class-conditional density and the prior probability as

$$\begin{aligned} \hat{p}(\mathbf{x}|\omega_j) &= \tfrac{k_j}{n_j V} \\ \hat{p}(\omega_j) &= \tfrac{N_j}{N}. \end{aligned}$$

Using Bayes' decision rule (Equation 2.3), we can then assign observation $\mathbf{x}$ to ω_j if

$$\frac{k_j}{N_j V}\frac{N_j}{N} \leq \frac{k_k}{N_k V}\frac{N_k}{N} \qquad \forall k. \tag{2.12}$$

In other words, assign **x** to the class that receives the largest vote amongst the k nearest neighbors. Any instances of a tie are broken randomly. Figure 2.5 shows the resulting decision boundaries defined by the k-nearest neighbor classifier for $k = 3$ using a two-dimensional feature subset of the white wine data set, with alcohol and malic acid as the two features considered.

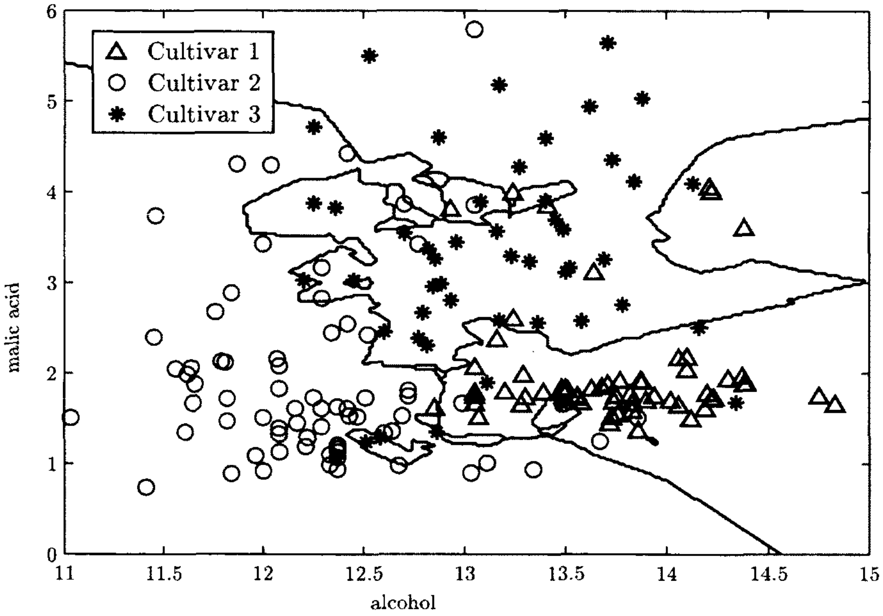


Figure 2.5: Decision boundary generated by the k-nearest-neighbor with $k = 3$ (3NN) for a two-dimensional feature subset of the white wine data set, with alcohol and malic acid as the two features considered. This classifier is prone to over-training, as is evident by the very irregular decision boundary fit to the training data set here.

2.4 Summary

In this chapter we have introduced an example classification problem based on a data set containing various properties of white wine. Bayes formula was provided for use with statistical pattern classification, and several classification routines were presented with example decision boundaries for the supplied example. We will apply these classifiers and more in the following chapters to various time-domain signals.

Features included for each application are specific to the fundamental properties of each signal being considered, but all include several features generated using the DWFP technique, presented in detail in Chapter 3. In what follows, the specifics on how features can be generated from the fingerprint images will be discussed.

Chapter 3

Time-Domain Signals Analysis using Wavelet Transforms

Our motivation is to identify subtle features within time-domain signals collected from real-world applications, where signals degrade due to any number of factors, including physical interactions, aging, decoupling, etc. The often complex raw waveforms are first transformed using the Dynamic Wavelet Fingerprinting (DWFP) technique, taking the one-dimensional time-domain signal and outputting a two-dimensional time-scale binary image. This chapter discusses the fundamentals behind the DWFP technique, and then provides two example applications of how it can be used to extract specific features of interest for structural health monitoring (SHM) using guided waves.

3.1 Wavelet Transforms

Wavelets are ideally suited for analyzing non-stationary signals, originally developed to introduce a local formulation of time-frequency analysis techniques. The *contin-*

uous wavelet transform (CWT) of a square-integrable, continuous function $s(t)$ can be written

$$C(a,b) = \int_{-\infty}^{+\infty} \psi_{a,b}^{*}(t)s(t)dt, \tag{3.1}$$

where $\psi(t)$ is the mother wavelet, * denotes the complex conjugate, and $\psi_{a,b}(t)$ is given by

$$\psi_{a,b}(t) = |a|^{-p}\psi\left(\frac{t-b}{a}\right). \tag{3.2}$$

Here, the constants $a, b \in \mathbb{R}$, where a is a scaling parameter defined by $p \geq 0$, and b is a translation parameter related to the time localization of ψ. The choice of p is dependent only upon which source in the literature is being referred to, much like the different conventions for the Fourier transform, so we choose to implement the most common value of $p = 1/2$ [59]. The mother wavelet can be any square-integrable function of finite energy, and is often chosen based on its similarity to the inherent structure of the signal being analyzed. The scale parameter a can be considered to relate to different frequency components of the signal. For example, small values of a result in a compressed mother wavelet, which will then highlight many of the high-detail characteristics of the signal related to the signal's high-frequency components. Similarly, large values of a result in stretched mother wavelets, returning larger approximations of the signal related to the underlying low-frequency components.

To better understand the behavior of the CWT, it can be rewritten as an inverse Fourier transform,

$$C(a,b) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{s}(\omega)\sqrt{a}\left(\hat{\psi}(a\omega)\right)^{*} e^{j\omega b}d\omega, \tag{3.3}$$

where $\hat{s}(\omega)$ and $\hat{\psi}(\omega)$ are the Fourier transforms of the signal and wavelet, respectively. From Eq. 3.3, it follows that stretching a wavelet in time causes its support in the frequency domain to shrink as well as shift its center frequency toward a lower frequency. This concept is illustrated in Figure 3.1. Applying the CWT with only a single mother wavelet can therefore be thought of as applying a bandpass filter, while a series of mother wavelets via changes in scale can be thought of as a bandpass filter bank.

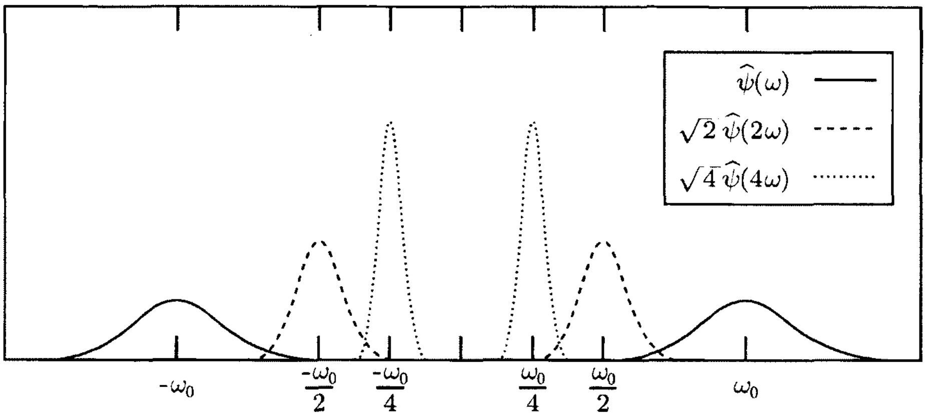


Figure 3.1: Frequency-domain representation of a hypothetical wavelet at scale parameter values of $a = 1, 2, 4$. It can be seen that increasing the value of a leads to both a reduced frequency support as well as a shift in the center frequency component of the wavelet toward lower frequencies. In this sense, the CWT acts as a shifting bandpass filter of the input signal.

An infinite number of wavelets are therefore needed for the CWT to fully represent the frequency spectrum of a signal $s(t)$, since every time the value of the scaling parameter a is doubled, the bandwidth coverage is reduced by a factor of 2. An efficient and accurate discretization of this involves selecting *dyadic* scales and positions based on powers of two, resulting in the *discrete wavelet transform* (DWT). In practice, the DWT requires an additional scaling function to act as a low-pass filter to allow for frequency spectrum coverage from $\omega = 0$ up to the band-pass filter range of the chosen wavelet scale. Together, scaling functions and wavelet functions provide full spectrum coverage for a signal. For each scaled version of the mother wavelet $\psi(t)$, a corresponding scaling function $\phi(t)$ exists.

Just as Fourier analysis can be thought of as the decomposition of a signal into various sine and cosine components, wavelet analysis can be thought of as a decomposition into *approximations* and *details*. These are generated through an implementation of the wavelet and scaling function filter banks. Approximations are the high-scale (low-frequency) components of the signal revealed by the low-pass scaling function filters, while details are the low-scale (high-frequency) components revealed by the high-pass wavelet function filter. This decomposition process is iterative, with the output approximations for each level used as the input signal for the following level, illustrated in Figure 3.2. In general most of the information in a time-domain signal is contained in the approximations of the first few levels of the wavelet transform. The details of these low levels often have mostly high-frequency noise information. If we remove the details of these first few levels and then reconstruct the signal with the inverse wavelet transform, we will have effectively de-noised the signal, keeping only the information of interest. This is the process behind wavelet filtering.

3.1.1 Wavelet Fingerprinting

Once a raw signal has been filtered, we then pass it through the DWFP algorithm. Originally developed by Hou [37], the DWFP applies a wavelet transform on the original time domain data, resulting in an image containing "loop" features that resemble fingerprints. The wavelet transform coefficients can be rendered in an image similar to a spectrogram, except that the vertical axis will be scale instead of frequency. These time-scale image representations can be quite helpful for identifying subtle signal features that may not be resolvable via other time-frequency methods.

Combining Eq. 3.1 and 3.2, the CWT of a continuous square-integrable function

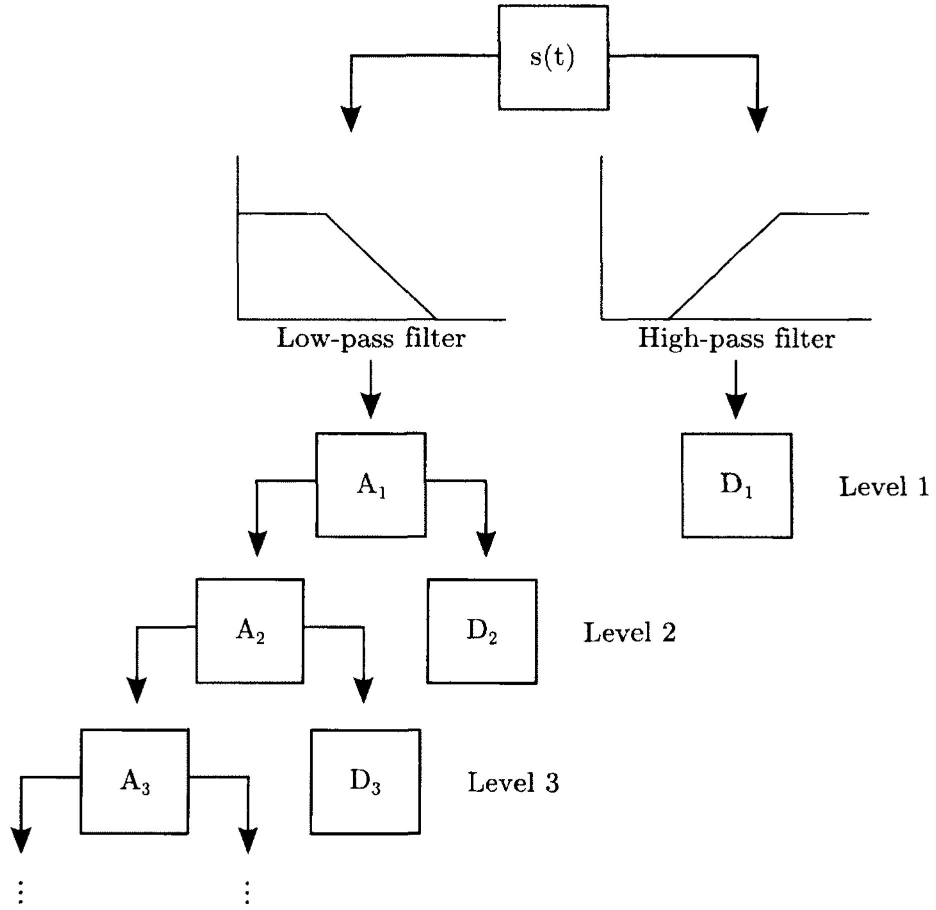


Figure 3.2: The signal is decomposed into approximations (A_1) and details (D_2) at the first level [60]. The next iteration then decomposes the first-level approximation coefficients into second level approximations and details, and this process is repeated for the desired number of levels. For wavelet filtering, the first few levels of details can be removed, effectively applying a low-pass filter to the signal.

$s(t)$ can be written

$$C(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} s(t)\psi^{*}\left(\frac{t-b}{a}\right) dt. \tag{3.4}$$

Unlike the DWT, where scale and translation parameters are chosen according to the dyadic scale ($a = 2^m, b = n2^m$, $n, m \in \mathbb{Z}^2$), the MATLAB implementation of the CWT used here utilizes a range of real numbers for these coefficients. A normal range of scales includes $a = 1, \ldots, 50$ and $b = 1, \ldots, N$ for a signal of length N. This results in a two-dimensional array of coefficients, $C(a,b)$, which are normalized to the range of $[-1, 1]$ (Fig. 3.3(b)). These coefficients are then sliced in a “thick” contour manner, where the number of slices and thickness of each slice is defined by the user. To increase efficiency, the peaks ($C(a,b) \geq 0$) and valleys ($C(a,b) < 0$) are considered separately. Each slice is then projected onto the time-scale plane (Fig. 3.3(c)). The resulting slice projections are labeled in an alternating, *binary* manner, resulting in a binary “fingerprint” image, $I(a,b)$ (Fig. 3.3(d)):

$$s(t) \xrightarrow{DWFP(\psi_{a,b})} I(a,b). \tag{3.5}$$

The values of slice thickness and number of slices can be varied to alter the appearance of the wavelet coefficients, as can changing which mother wavelet is used. The process of selecting mother wavelets for consideration is application-specific, since certain choices of $\psi(t)$ will be more sensitive to certain types of signal features. In practice, mother wavelets used are often chosen based on preliminary analysis results as well as experience [37,39,42,46,61 64].

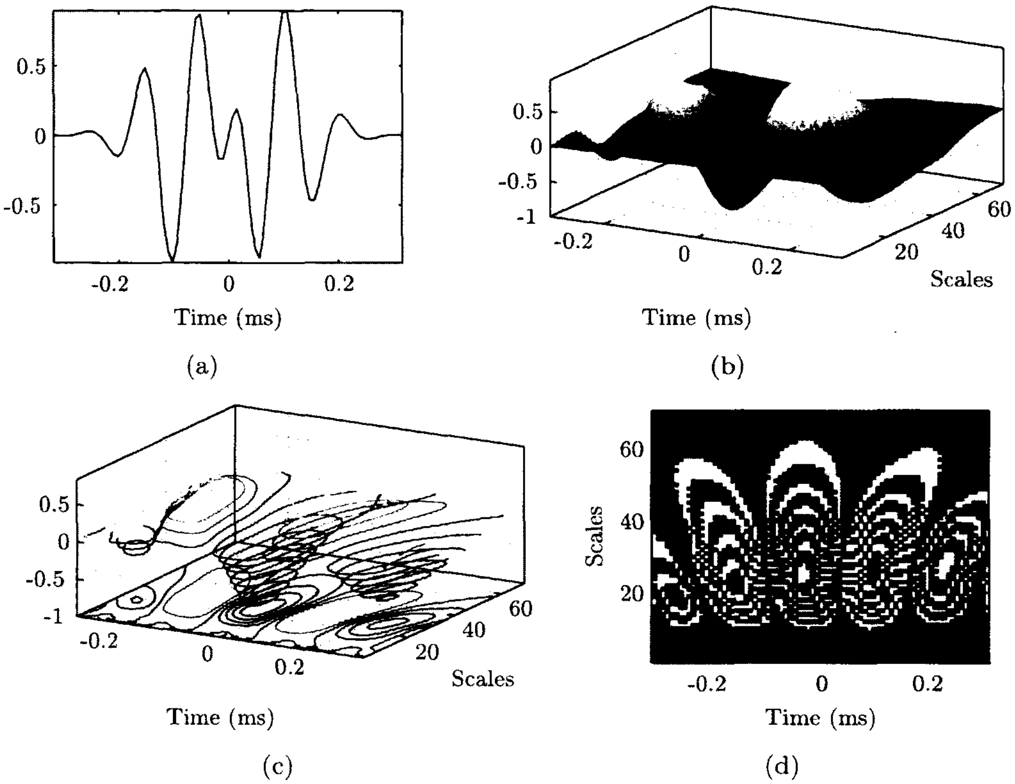


Figure 3.3: A visual summary of the DWFP algorithm [37]. A time-domain signal (a) for which a set of wavelet coefficients (b) is generated via the continuous wavelet transform. The coefficients are then "thickly" sliced and projected onto the time-scale plane (c), resulting in two-dimensional binary images (d), shown here with white peaks and gray valleys for distinction.

3.1.2 Feature Extraction from Wavelet Fingerprints

The problem of time-domain signal analysis has thus been transformed from a 1D signal identification problem to a 2D image recognition scenario. This concept is the underlying advantage behind the DWFP technique, as the output is a binary matrix that is easily stored and transferred. The ability to select specific mother wavelets to highlight different features in the signals provides a robustness to the algorithm as well.

The last step in applying the DWFP technique to signal analysis is identifying features within the binary images that correspond to the waveform features of interest. A ridge counting algorithm applied to the 2D images is a straightforward way to compare fingerprint images and identify some of the features of interest. An example of ridge counting is shown in Figure 3.4. For each column of the binary image matrix, the number of connected "on" regions corresponds to the number of ridges for that point in time. In Figure 3.5 we show two longer fingerprint images generated from ultrasonic waveforms with and without flaw interaction. For the majority of the signal, the number of ridges between the flawed and unflawed waveforms match. The signature of the flaw is easily identified here in the region indicated by the dashed rectangle. A simple threshold can be applied to the ridge-counting metric for damage detection.

Representing a one-dimensional time-domain signal as a two-dimensional time-frequency image is an inherent advantage of time-frequency representations, allowing for the application of image processing methods to assist in the analysis. Full false-color images are both computationally expensive and generally difficult to analyze when the goal is to automatically identify features of interest within a signal. Image segmentation is usually a hurdle with automated processing, especially when the flaw's signature is identified by a specific feature shape or size within the im-

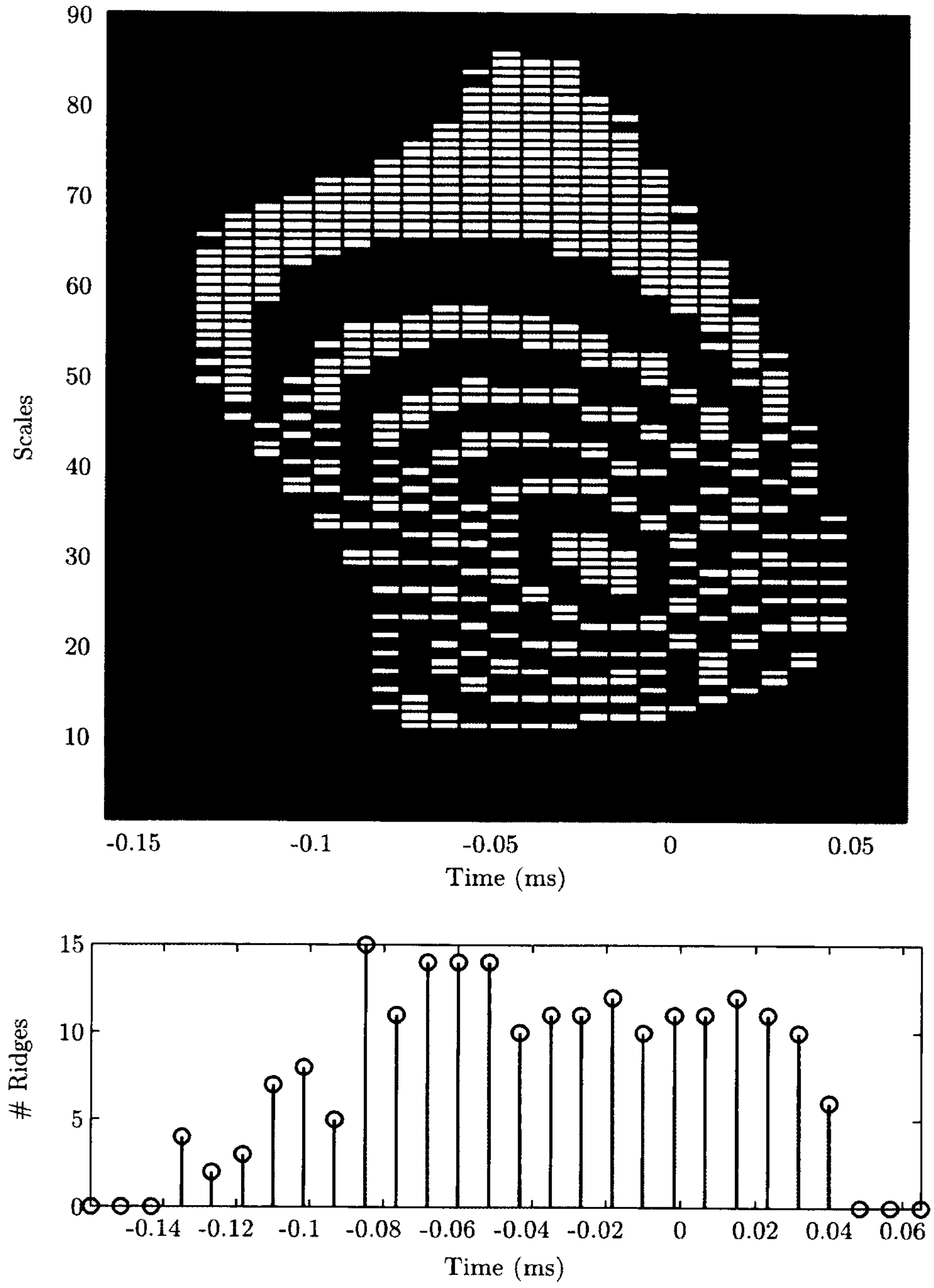


Figure 3.4: Example of ridge counting in a fingerprint image. The number of connected "on" pixel regions in each column corresponds to the number of ridges for that point in time.

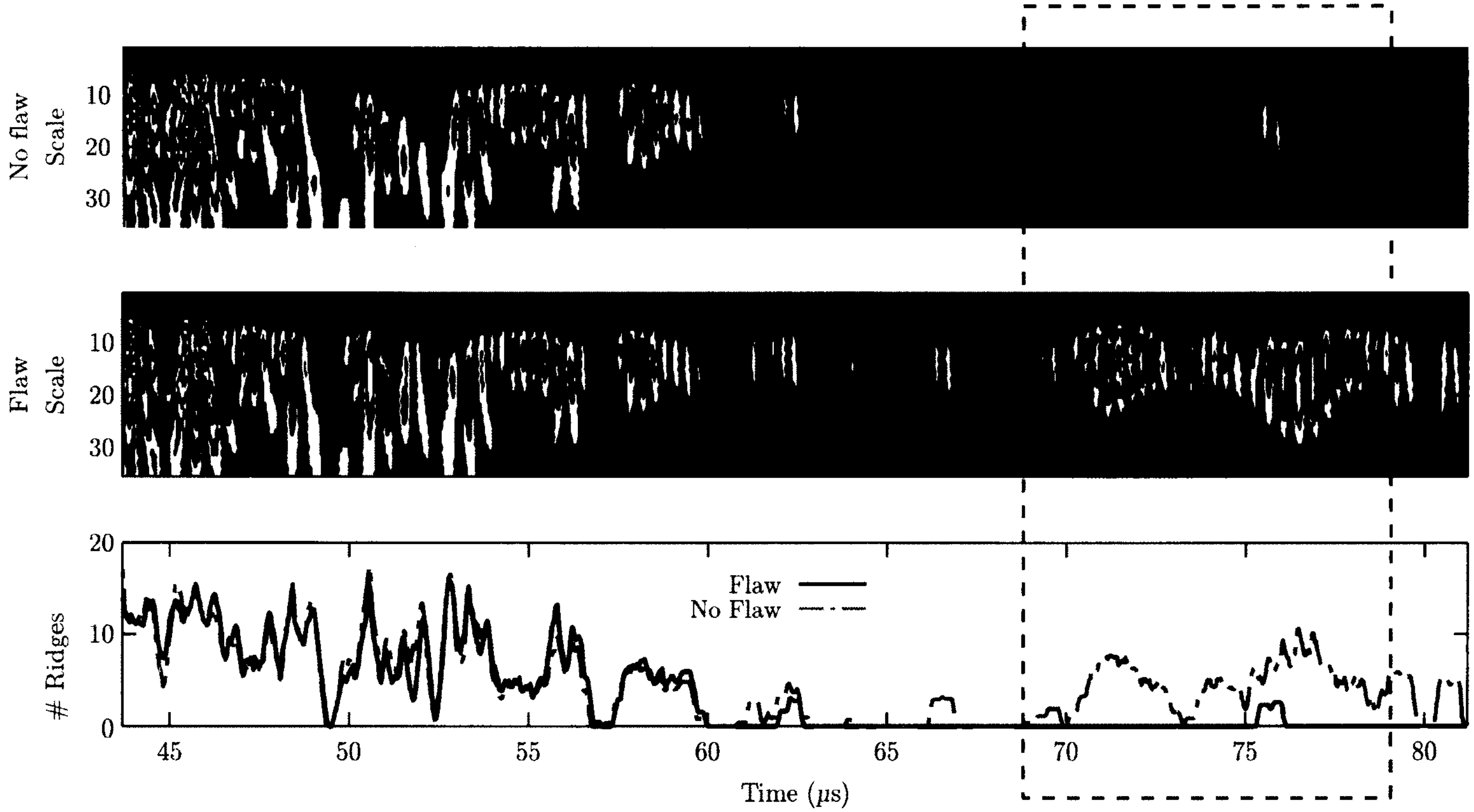


Figure 3.5: Fingerprint images from two waveforms without (top) and with (middle) a flaw. The result of a ridge counting algorithm is plotted (bottom), with the region identifying the flaw highlighted by the dashed rectangle.

age, as is often the case. A binary image requires significantly less computer storage than a corresponding gray-scale or color image, and segmentation is trivial. While this sort of fingerprint image can be formed from any time-frequency representation, wavelets have been shown to work quite well for a wide variety of applications previously investigated, including an ultrasonic periodontal probe [39–41], structural health monitoring with Lamb waves [43, 44], RFID signal analysis [64], and robot navigation [65, 66].

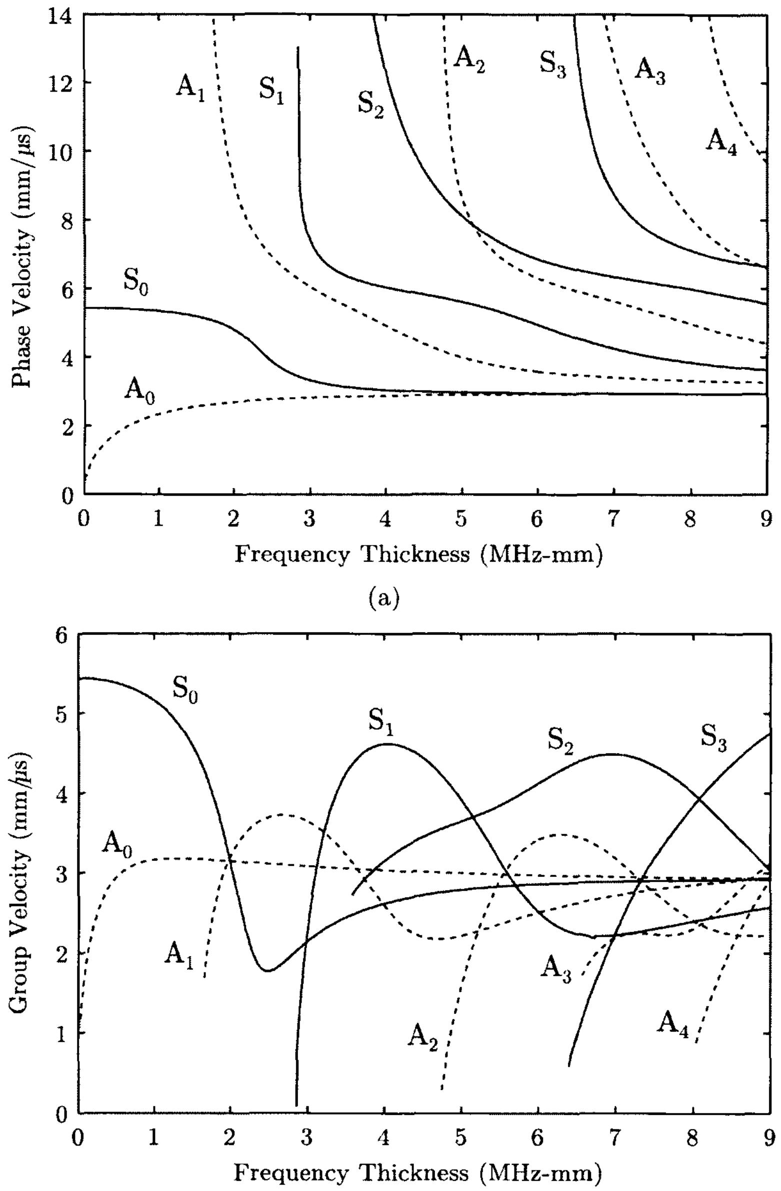
We have introduced the principles of using the DWFP wavelet transform technique for time-domain signal analysis. We next apply this technique to two real-world industrial structural health monitoring (SHM) applications. First, we explore how the DWFP technique can be utilized to distinguish dents and the resulting rear surface cracks generated in aircraft grade aluminum plates. We then show how wavelet fingerprints can be used to identify corrosion in marine structures beneath layers of insulation.

3.2 Applications in Nondestructive Evaluation

Originally explored by Lord Rayleigh in 1885 while investigating the propagation of surface waves in a solid, e.g. as in earthquakes, the study of guided waves has been advanced over the years [67–71]. Guided waves occur when vibrations propagate at stress-free boundaries, or boundaries between two differing media. Deriving from the same fundamental equations as bulk waves, guided waves experience mode conversion and interference due to the guiding boundaries, which result in an infinite number of dispersive guided wave modes [72]. Lamb waves are a subset of guided wave found in plate-like structures. The basic derivations and discussions of Lamb waves can be found in various texts [43, 72–81], while a more thorough discussion of current techniques can be found in literature reviews [82, 83]. Lamb waves are confined by a structure's boundaries and so follow its shape and curvature, with sensitivity to material discontinuities at either surface as well as in the interior of the solid. The Rayleigh-Lamb wave equations, derived in Appendix A, can be numerically solved to generate plots of the modal group velocity as a function of a combined frequency-thickness product. Also known as dispersion curves, these plots relate a combined frequency-thickness parameter to the phase and group velocities of both the symmetric and antisymmetric families of modes, and are shown in Figure 3.6 for aluminum.

The key technical challenge to using Lamb waves effectively is automatically identifying which modes are which in very complex waveform signals. As seen in the dispersion curves, there are usually several guided wave modes present at any given frequency-thickness value, often having overlapping mode velocities. Since each mode propagates with its own modal structure, modes traveling with the same group velocity result in a superposition of the individual displacement and stress components. The resulting signals are inherently complex, and sophisticated analysis techniques need to be applied to make sense of the signals. Contrary to traditional bulk-wave ultrasonics, standard peak-detection approaches often fail for Lamb wave analysis.

Numerous methods have been developed in an attempt to simplify Lamb wave analysis. One often used approach is to use a statistical comparison in which a baseline measurement is compared to the damaged state signal. The main problem with baseline-dependent data is that it is usually not practical for real-world situations where environmental and operational conditions result in deviations from baseline data that do not correlate to damage. Also, the dependence on a previously established baseline makes it difficult to implement with current technologies. Alternative inspection techniques to a baseline method exist, but often include an assumption or restriction that renders them impractical. Examples include assuming uniform



(a)

(b)

Figure 3.6: Dispersion curves for an aluminum plate. Solutions to the Rayleigh-Lamb wave equations are plotted here for both symmetric (solid lines) and antisymmetric (dashed lines) for both phase (a) and group (b) velocity.

thickness in the plate of interest, greatly reducing the number of possible real-world applications [84], restricting the inspection frequency by exciting a specific Lamb wave mode and analyzing the reflected and mode-converted waves [85], and expanding the ray coverage to damaged as well as undamaged areas for an instantaneous cross-correlation difference analysis [86,87].

3.2.1 Dent and Surface Crack Detection in Aircraft Skins

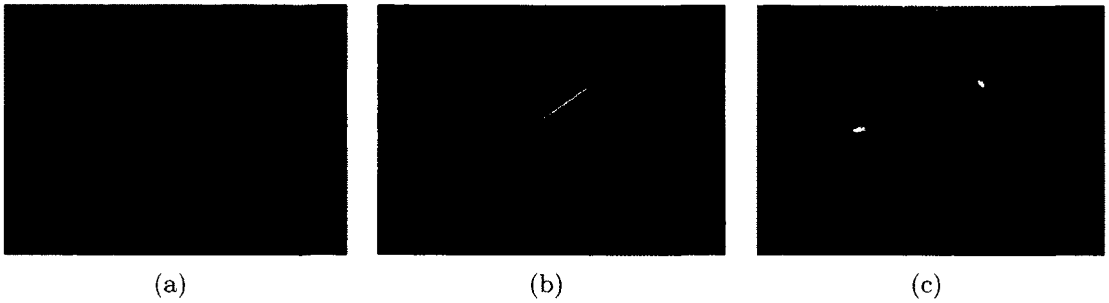
Aircraft manufacturers and operators have concerns over fuselage damage commonly called "ramp rash". This damage occurs in the lower fuselage sections, and can be caused by incidental contact with ground equipment such as air stairs, baggage elevators, and food service trucks [88]. Ramp rash costs airlines billions of dollars per year in both damage repair costs and downtime [89]. Even minor damage can become the source of serious mechanical failure if left unattended [90]. The formation of dents in the fuselage is common, but the generation of rear cracks can lead to serious complications. Repairing this subsequent damage is critical for aviation safety, however the hidden extent of the damage is initially unknown. In order to accurately estimate repair requirements, and therefore reduce unnecessary downtime losses, an accurate assessment of the damage is required.

A continuously monitoring inspection system at fuselage areas prone to impact would provide an alternative to conventional point-by-point inspections. Guided waves have previously shown potential for damage detection in metallic aircraft components [62]. We test a guided wave-based SHM technique for identifying potential flaws in metallic plate samples. We employ the DWFP to generate time-scale signal representations of the complex guided waveforms, from which we extract subtle features to assist in damage identification.

Experimental Setup

The samples tested were aircraft-grade 1.62 mm thick aluminum 2024 plates (with primer), approximately 45 cm x 35 cm in size. Some of them had dents and some also had simulated cracks of varying length and orientation, images of which are shown in Figure 3.7, including the following flaw types:

- Dent without crack
- Dent with crack between 2-40 mm
- Dent with crack larger than 40 mm
- Crack between 2-40 mm
- Crack larger than 40 mm



(a) (b) (c)

Figure 3.7: A plate with no flaw present (a), a plate with a shallow, large crack present (b), and a plate with both a dent as well as a crack present (c).

Angle beam transducers were used to generate surface waves that produce shear waves normal to the surface of the sample. These waves propagate exclusively along the surface of thicker materials and are known as Rayleigh waves. In thin plates, when they are guided by boundary layers as they are in aircraft skins, they are referred to as Rayleigh-Lamb waves. Angle beam transducers focus their propagation energy in one direction using Snell's law, unlike contact transducers that spread their energy in all directions. This focused, directed energy can be favorable when specific areas are being considered that are known in advance to be prone to damage.

Experimental data was collected using transducers manufactured by Olympus[1]. Signals were generated and received using a Nanopulser 3 (NP3) from Automated Inspection Systems[2]. The NP3 integrates both a pulser/receiver and a digitizer into a portable USB-interfaced unit, emitting a spike excitation instead of a traditional tone burst often used in NDE. The spike excitation then relies on the resonant frequency of the transducers in order to generate the ultrasonic waves with a broad frequency content to maximize scattering interaction with the surface cracks. The compact size of the NP3 makes it favorable for in-field use for aircraft damage analysis. A graphical user interface was developed in MATLAB to control the NP3 system.

We explore both a pitch/catch scanning configuration, where one transducer is the transmitter while a second transducer is the receiver, as well as a pulse/echo configuration where one transducer acts as both the transmitter and receiver. We control the position of each transducer by attaching two individual transducers to linear slides, controlled by stepper-motors that advance them along the edges of the test area. The test area was centered around the flaw, with the transducers placed 215 mm apart from each other and advancing through 25 individual locations in 8.6 cm increments for a total scan length of 215 mm. An illustration of this is shown in Figure 3.8. For the pitch/catch configuration, two transducers are stepped in parallel. For the pulse/echo configuration, only one transducer is present.

Each waveform is run through the DWFP algorithm to generate a fingerprint representation. This process is demonstrated for a typical waveform in Figure 3.9, and involves windowing the full signal 3.9(a) around the first arriving wave mode 3.9(b), which is then used as an input signal for the DWFP transformation 3.9(c).

[1] Waltham, MA (`http://www.olympus-ims.com/en/`)

[2] Martinez, CA (`http://www.ais4ndt.com/`)

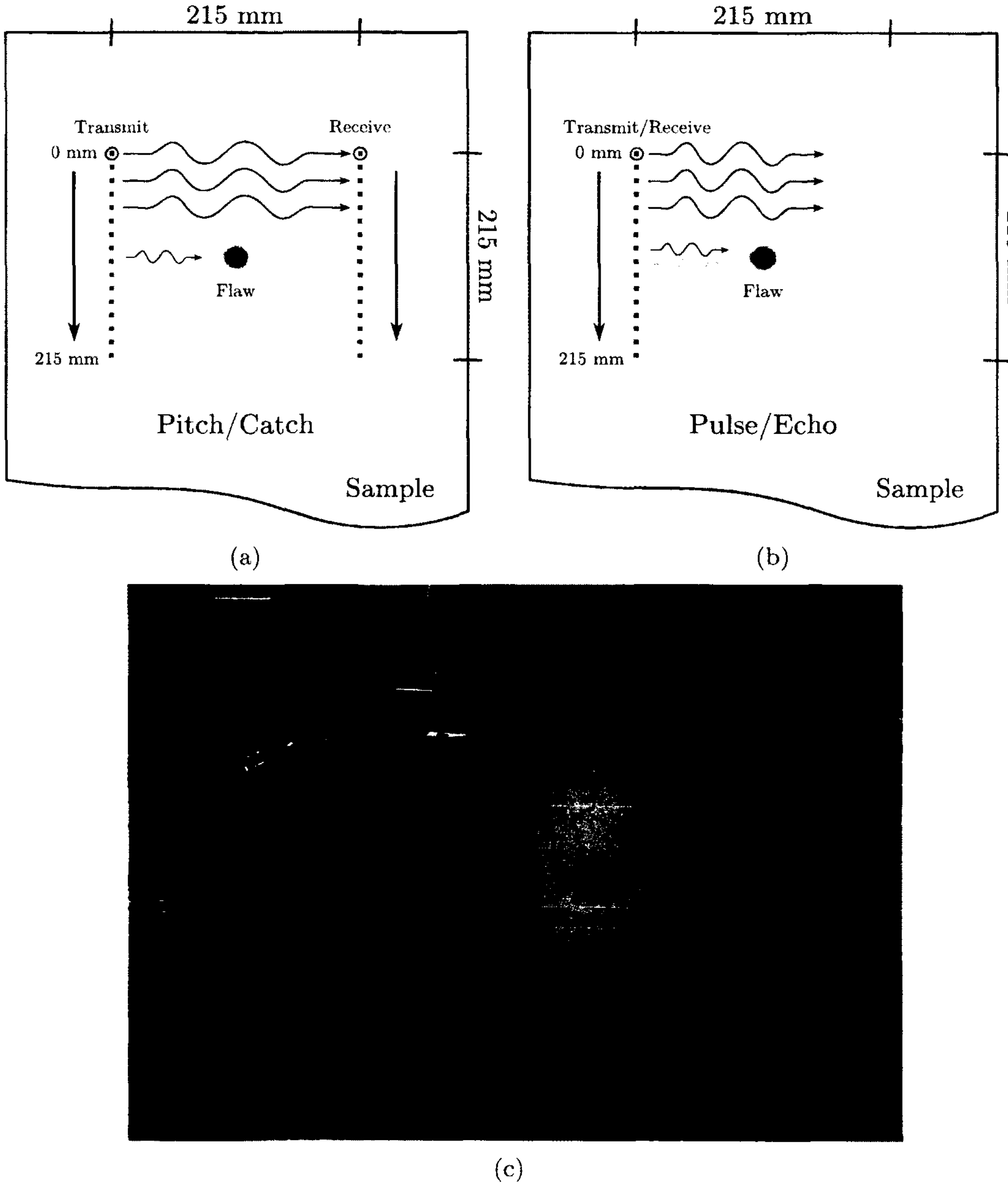
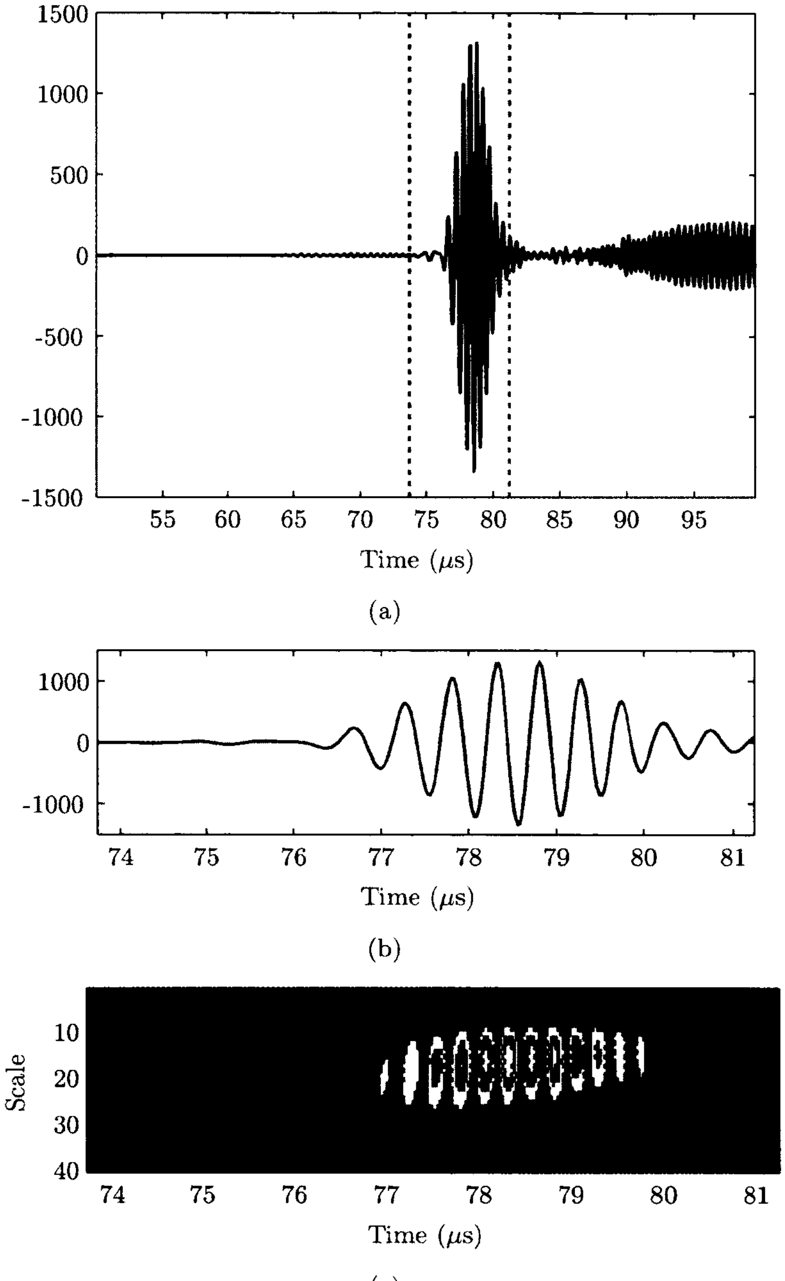


Figure 3.8: Scanning apparatus for both a pitch/catch configuration (a) as well as a pulse/echo configuration (b). Transducers are attached to linear slides controlled by stepper motors on either side of the sample, allowing for systematic scanning to be performed (c).



(a)

(b)

(c)

Figure 3.9: A raw Rayleigh-Lamb waveform collected from an unflawed plate sample (a). The signal is first windowed to the region of interest, here the first mode arrival (b), from which a DWFP image is generated (c).

Dent Detection - Pitch/Catch Scanning

We first present results here from the pitch/catch scanning configuration. The first sample considered contained a crack only, and the raw waveforms can be seen in Figure 3.10. It can be seen that these signals are all very similar to each other, and little information is readily available to identify the crack in the time-domain. Figure 3.11 provides the DWFP representations of these same signals. For each individual DWFP image, a specific feature is identified and automatically tracked throughout the scan progression. The feature in this case is the first arriving mode, indicated by the first "peak" (in white) in time. For each following DWFP image, this feature is identified and highlighted by a red star ($*$), with the corresponding position in time given to the right of each fingerprint image. This first plate sample again shows no variation in the DWFP representations as the transducers are moved along the edges of the plate. The feature of interest highlighted varies little in its position. This indicates that the pitch/catch scanning configuration is inadequate for identifying surface cracks in the material.

The second plate scanned contained a dent only. Raw waveforms collected as the pair of transducers progress along the sample edges can be seen in Figure 3.12. It can be seen again that these signals are all very similar to each other, with the time-domain representations unable to highlight the dent. Figure 3.13 provides the DWFP representations of these same signals. The same feature described above, the first arriving mode indicated by the first DWFP peak, is identified and tracked again. It can be seen that there is a region of the sample where this feature shifts, indicating the presence of a discontinuity in the propagation caused by the dent. It follows that the pitch/catch scanning configuration is adequate for identifying dents in the material.

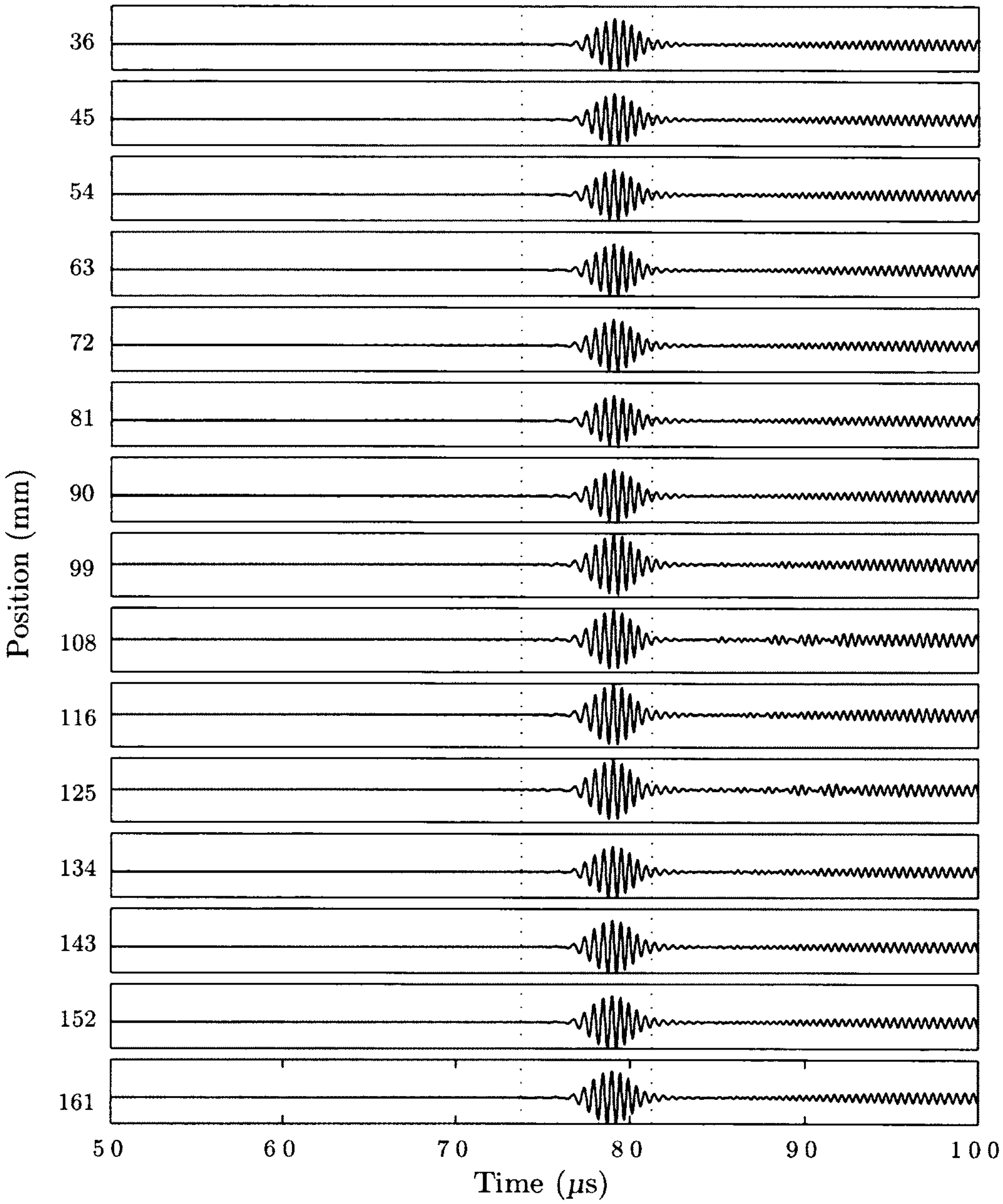


Figure 3.10: Raw waveforms collected from a plate sample with a crack only, using a pitch/catch scanning configuration. Little information is readily available from the time-domain representation of the signal. The dotted lines indicate the windowed region used for the DWFP image generation.

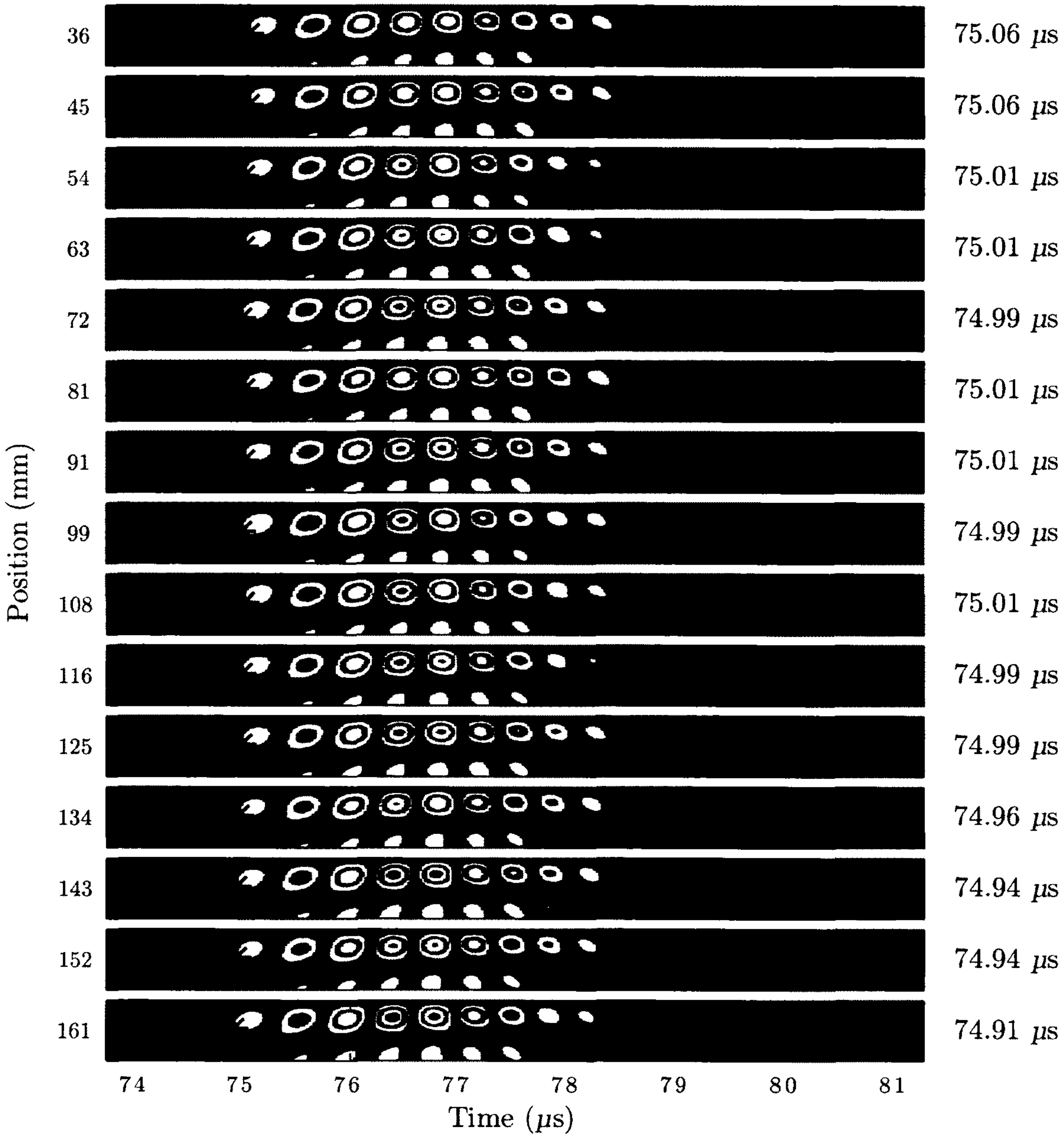


Figure 3.11: DWFP representations of the pitch/catch raw waveforms shown in Figure 3.10 from a sample with a crack only. For each individual DWFP image, a specific feature is identified and automatically tracked throughout the scan progression. The feature in this case is the first arriving mode, indicated by the first "peak" (in white) in time. For each following DWFP image, this feature is identified and highlighted by a red star (∗), with the corresponding position in time given to the right of each fingerprint image.

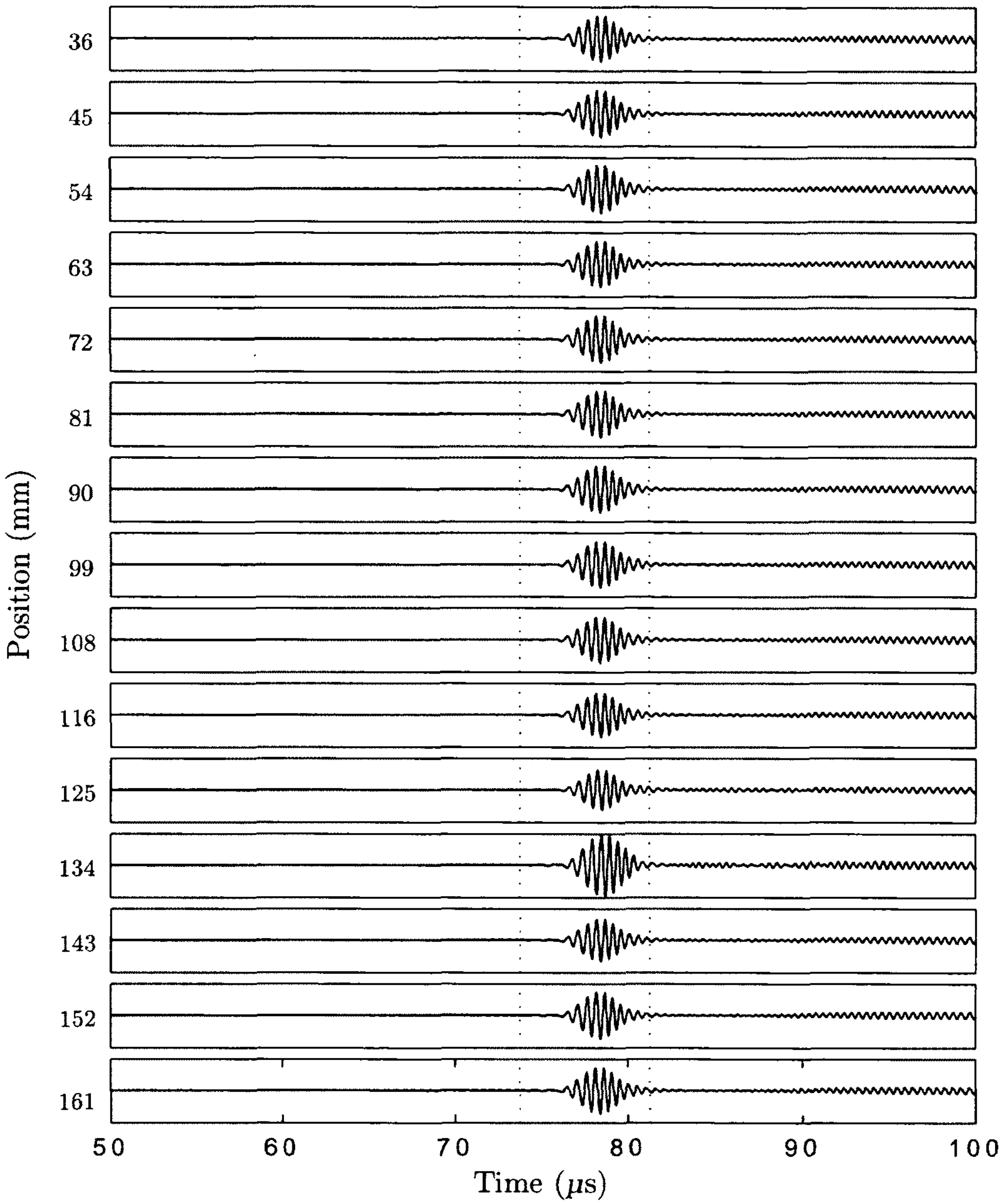


Figure 3.12: Raw waveforms collected from a plate sample with both a dent as well as a crack, using a pitch/catch scanning configuration. Little information is readily available from the time-domain representation of the signal. The dotted lines indicate the windowed region used for the DWFP image generation.

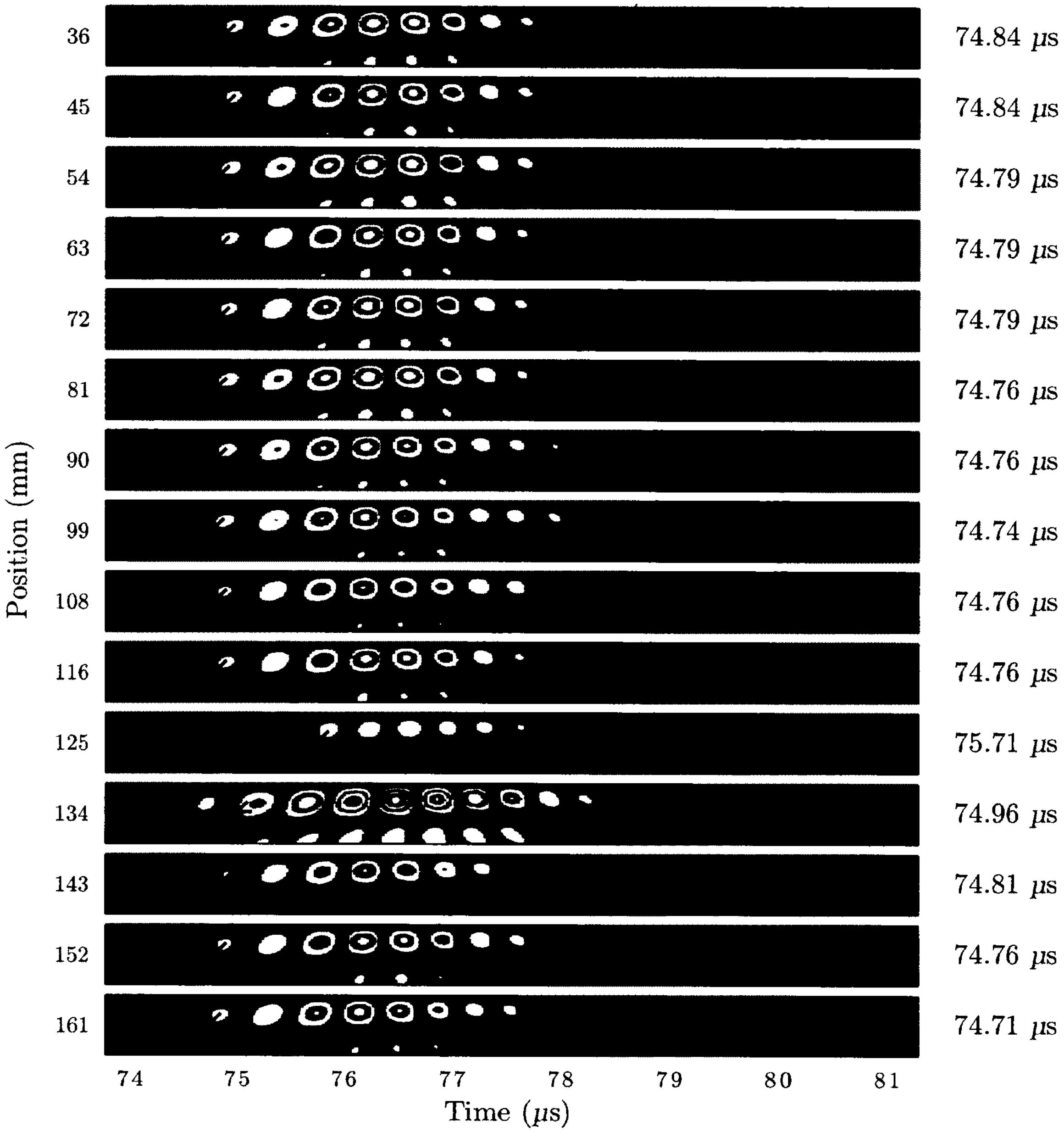


Figure 3.13: DWFP representations of the pitch/catch raw waveforms shown in Figure 3.12from a sample with a dent only. For each individual DWFP image, a specific feature is identified and automatically tracked throughout the scan progression. The feature in this case is the first arriving mode, indicated by the first "peak" (in white) in time. For each following DWFP image, this feature is identified and highlighted by a red star (∗), with the corresponding position in time given to the right of each fingerprint image.

Crack Detection - Pulse/Echo Scanning

We now use the same transmitting transducer in pulse/echo mode to determine if any energy is being reflected from either type of flaw to aid in detection. We again present results here for the two samples previously considered: one with a crack, and one with a dent. We first present the raw waveforms collected from interaction with the crack only, shown in Figure 3.14. The raw waveforms contain an initial high-amplitude portion that is a result of energy reflecting within the transducer itself. We window our search in the time-frame where a reflection would be expected to exist if present. It can be seen that these signals are all very similar to each other, and all have a low signal-to-noise ratio making it difficult to easily detect any features identifying a flaw.

Figure 3.15 provides the DWFP representations of these same signals. For each individual DWFP image, a specific feature is identified and automatically tracked throughout the scan progression. The feature in this case is the first peak "doublet" feature after the 65 μs mark, where a doublet is indicated by two fingerprint features existing for different scales at the same point in time. For each following DWFP image, this feature is identified and highlighted by a red star ($*$), with the corresponding position in time given to the right of each fingerprint image. If no such feature is found, the end point of the window is used. We can see that the DWFP representations of the pulse/echo signals are able to identify waveforms that correlate with the position of the crack if present.

The second plate under consideration contains a dent only. Raw waveforms collected as the transducer progressed along an edge of the sample can be seen in Figure 3.16. It is again clear that these signals are all very similar to each other, with the low signal-to-noise ratios making it difficult to easily extract any information from them. Figure 3.17 provides the DWFP representations of these same signals. The same feature described above, the first peak "doublet" feature after the 65 μs mark, is

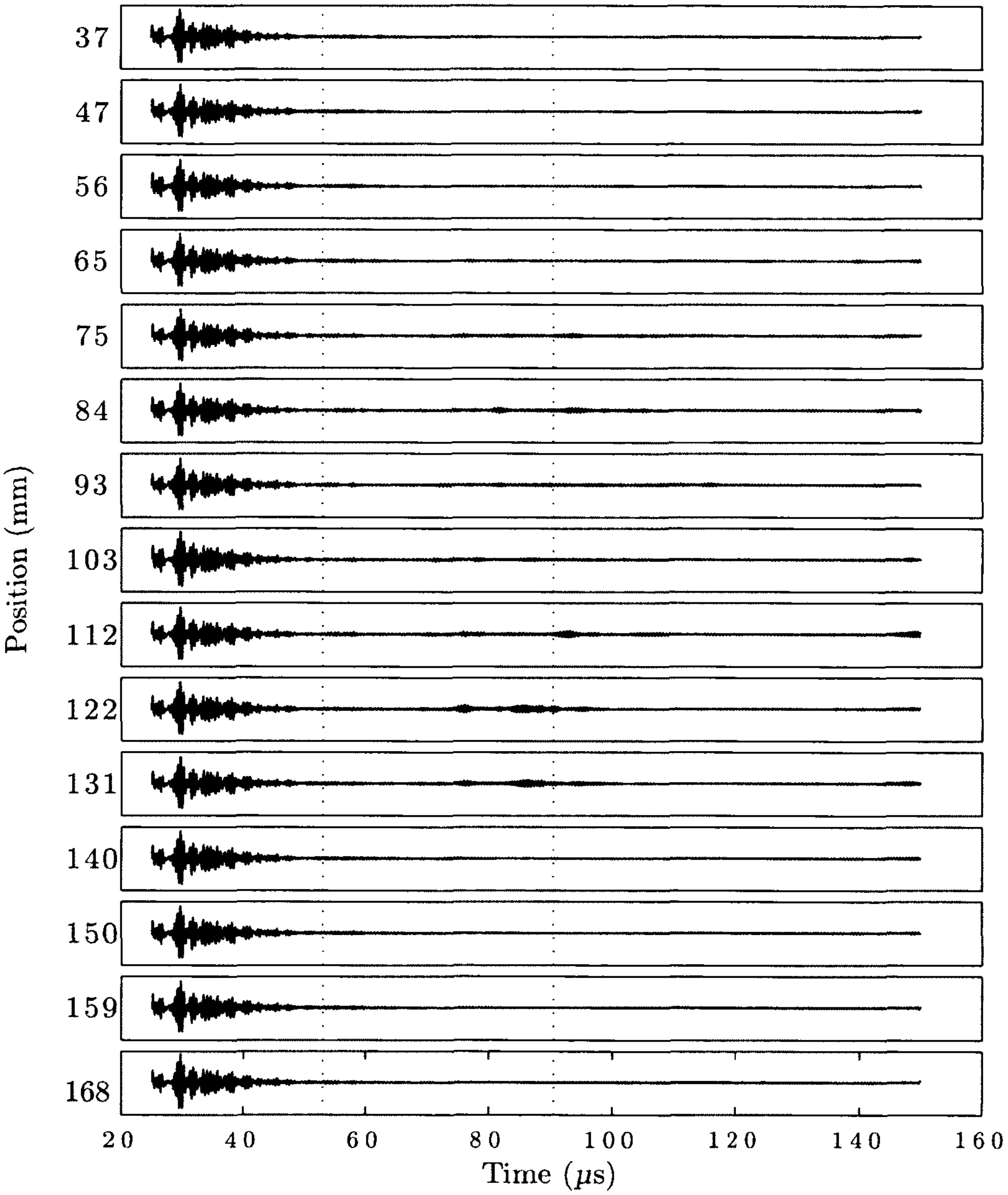


Figure 3.14: Raw waveforms collected from a plate sample with a crack only, using a pulse/echo scanning configuration. The low signal-to-noise ratio makes it difficult to analyze these raw time-domain signals. The dotted lines indicate the windowed region used for the DWFP image generation.

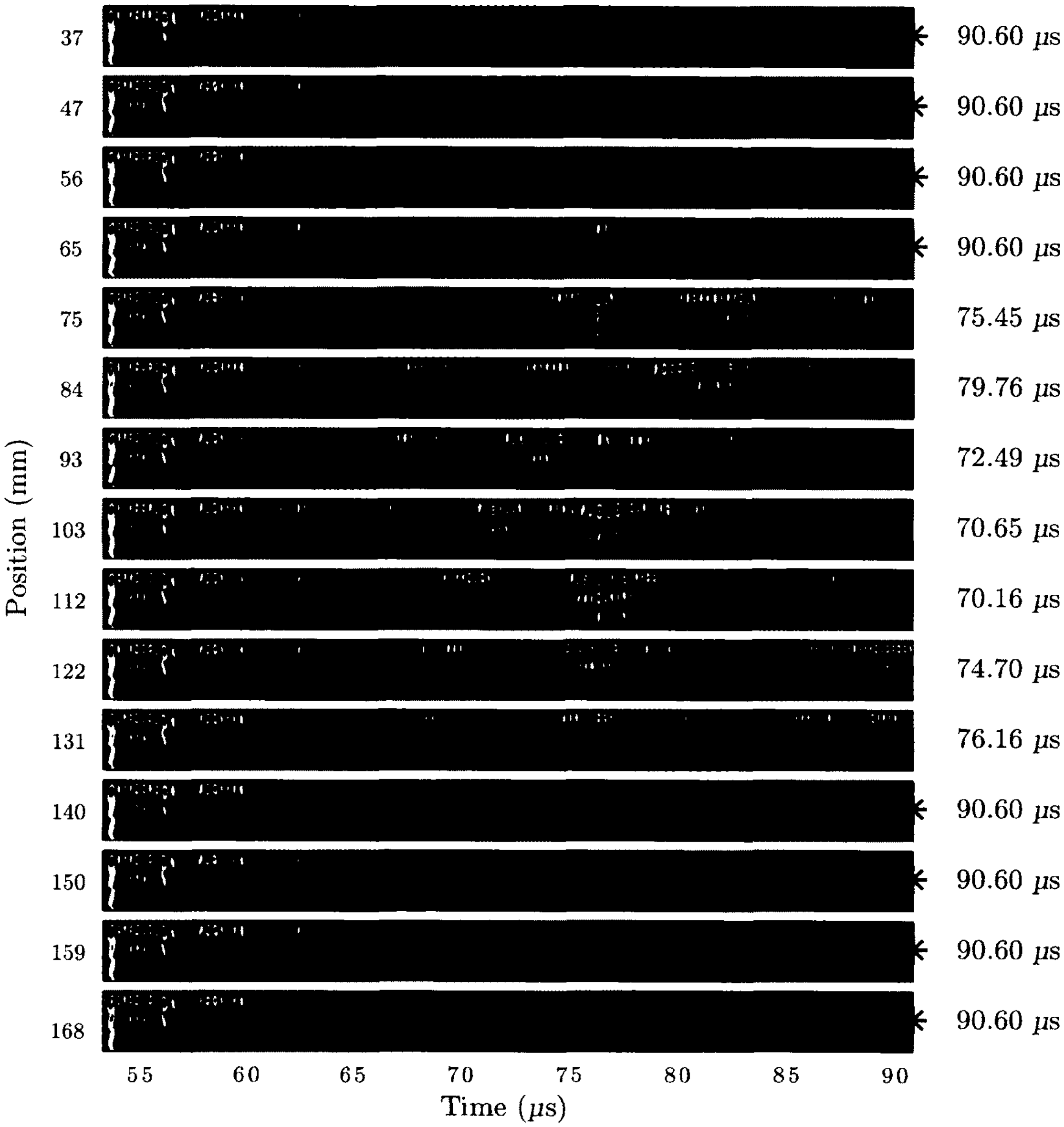


Figure 3.15: DWFP representations of the pulse/echo raw waveforms shown in Figure 3.14 from a sample with a crack only. For each individual DWFP image, a specific feature is identified and automatically tracked throughout the scan progression. The feature in this case is the first peak "doublet" feature after the 65 μs mark, where a doublet is indicated by two fingerprint features existing for different scales at the same point in time. For each following DWFP image, this feature is identified and highlighted by a red star (*), with the corresponding position in time given to the right of each fingerprint image.

again identified and tracked. It can be seen that there is no region in this scan where the feature is identified, indicating that this pulse/echo approach is not sufficient to identify dents.

For each of the DWFP images, we have identified and automatically tracked in time a specific feature within the image. We summarize the extracted time locations of these features in Figure 3.18, where the extracted times are plotted against their position relative to the plate. Vertical dotted lines indicate the actual (known) locations of each flaw present. We can clearly see that the pitch/catch waveforms are able to identify the dent, while the pulse/echo waveforms are able to identify the crack. This follows from the concept that the surface crack is not severe enough to distort much of the propagating waveform, but still reflects enough energy to be identified back at the initial transducer location.

Crack Angle Dependence

In order to determine how the angle of the crack affects the reflected signal's energy with respect to the incident wave angle, an angle-dependence study was performed. A sample containing a large crack only, a dent without a crack, and a dent with a large crack were included here. A point in the center of the flaw (either the center of the dent, or the center of the crack) was chosen as the center of rotation, and the Rayleigh-Lamb transducers were placed 10 cm from this center point in 10° increments around the point of rotation up to a minimum of 120° away from the starting location. A pulse/echo measurement was taken at each location to measure any energy reflected from the flaw.

Two features we extracted from the recorded pulse/echo signals are the arrival time of the reflected signal and the peak instantaneous amplitude of that reflection. The instantaneous amplitude is calculated by first calculating the discrete Hilbert

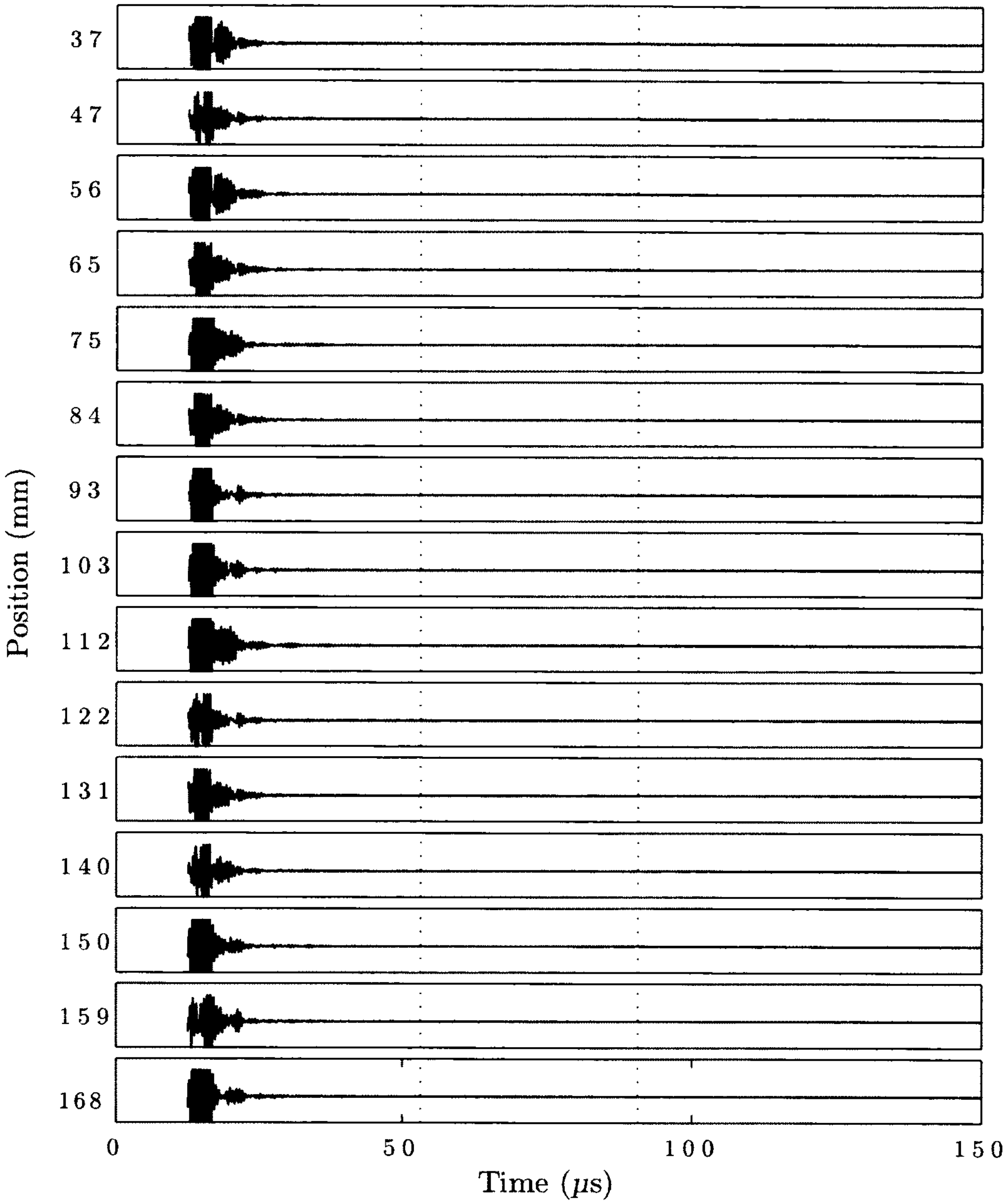


Figure 3.16: Raw waveforms collected from a plate sample with a dent only, using a pulse/echo scanning configuration. The low signal-to-noise ratio makes it difficult to analyze these raw time-domain signals. The dotted lines indicate the windowed region used for the DWFP image generation.

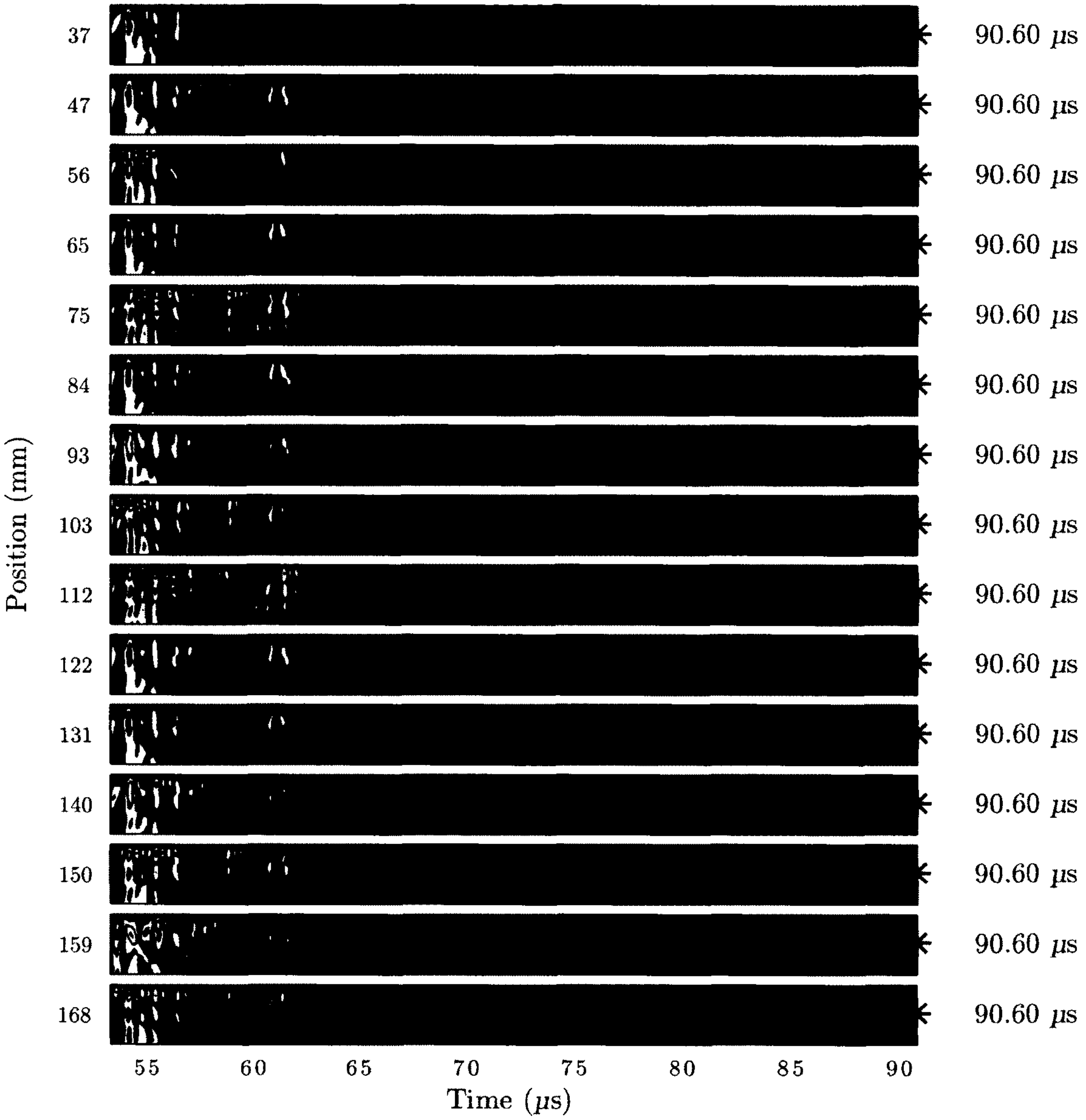
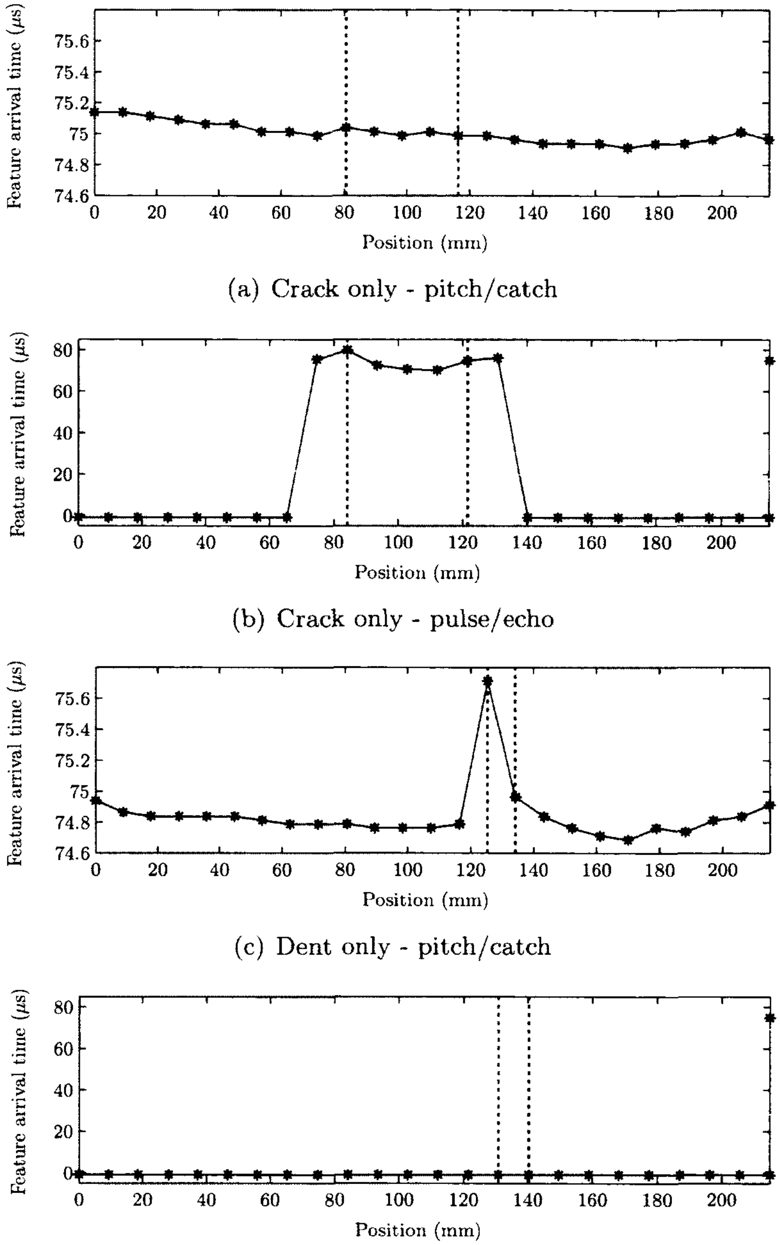


Figure 3.17: DWFP representations of the pulse/echo raw waveforms shown in Figure 3.16 from a sample with a dent only. For each individual DWFP image, a specific feature is identified and automatically tracked throughout the scan progression. The feature in this case is the first peak "doublet" feature after the 65 μs mark, where a doublet is indicated by two fingerprint features existing for different scales at the same point in time. For each following DWFP image, this feature is identified and highlighted by a red star ($*$), with the corresponding position in time given to the right of each fingerprint image.



(a) Crack only - pitch/catch

(b) Crack only - pulse/echo

(c) Dent only - pitch/catch

(d) Dent only - pulse/echo

Figure 3.18: Extracted time locations of the features identified in each of the DWFP representations for the sample with a crack only using pitch/catch (a) and pulse/echo waveforms (b), as well as those from the sample with a dent only using pitch/catch (c) and pulse/echo waveforms (d). We see that the pitch/catch scanning configuration is able to identify any dents present, while the pulse/echo configuration is better at identifying any cracks present.

transform of the signal $s(t)$, which returns a version of the original signal with a 90^{circ} phase shift (preserving the amplitude and frequency content of the signal) [91]. The magnitude of this phase-shifted signal and the original signal is the instantaneous amplitude of the signal, another name for the signal's envelope. The maximum of this instantaneous amplitude is what will now be referred to as the "peak energy" of the signal.

The first sample (Figure 3.19(a)), which did not have a crack present, did not return any measurable reflected energy at any angle (Figure 3.19(b)). The two samples with cracks (Figure 3.19(c) and Figure 3.19(e)) both show a measurable reflection from the crack when the incident angle is normal to the crack, regardless of whether or not a dent is present (Figure 3.19(d) and Figure 3.19(f)). Incident angles that are 0-20° from normal still had measurable reflection energy, however incident angles beyond that did not have a significant measurable reflection. These results agree with expectations that the cracks would be highly directional in their detection.

3.2.2 Corrosion Detection in Marine Structures

Structural health monitoring is an equally important area of research for the world's navies. From corrosion due to constant exposure to harsh salt-water environments, to the more recent issue of sensitization due to the cyclic day/night heat profile exposure of the open water, maritime vessels are in constant need of repair. The biggest cost in maintenance of these ships is often having to pull them out of services in order to characterize and repair any and all damage present. Navies and shipyards are actively researching intelligent monitoring systems that will provide constant feedback on the structural integrity of areas prone to these damages.

Lamb waves provide a natural approach to identifying corrosion in metals. Each mode's group velocity is dependent on both the inspection frequency used as well

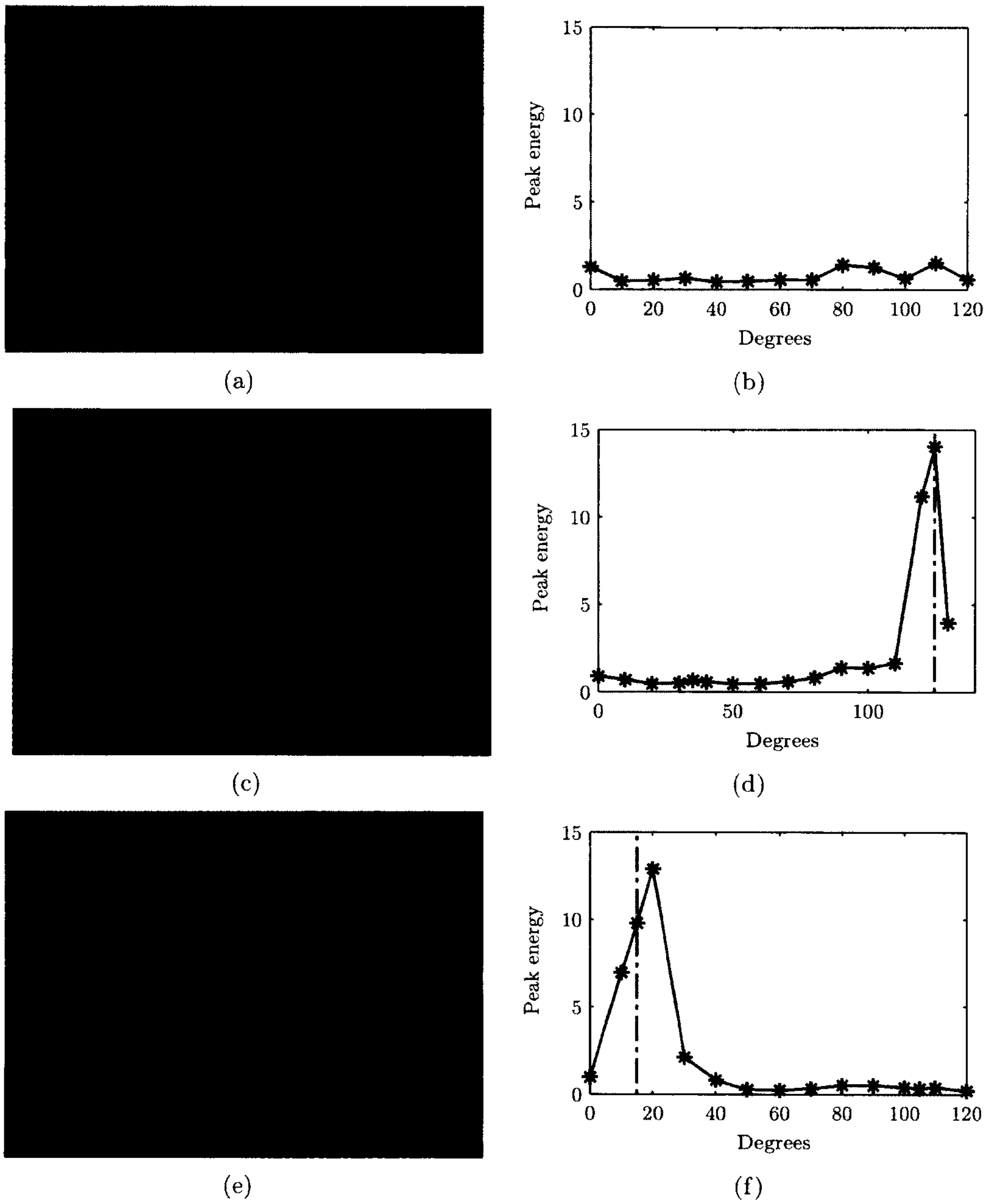


Figure 3.19: A plate sample with a dent only (a). Results show no reflection above the noise level at any angle due to no crack being present (b). A plate sample with a dent and a large crack (c). Results show significantly higher reflection energy when the Rayleigh-Lamb waves were incident at an angle close to broadside (d). A plate sample with a large crack only (e). Results show highest reflection energy at normal incidence to the crack (f).

as the material thickness (see Appendix A). Since corrosion can be thought of as a local material loss, it follows that mode velocities will change when passing through a corroded area. They will either speed up or slow down, depending on the modes and frequency-thickness product being used. Often several modes will propagate simultaneously within a structure and will overlap each other if they share a similar group velocity. As long as a frequency-thickness regime is chosen so that a single mode is substantially faster than the rest and therefore arrives earlier, most of the slower, jumbled modes can be windowed out of the signal. Key to utilizing Lamb waves for SHM is understanding this dispersion-curve behavior since this is what allows arrival time shifts to be correlated directly to material thickness changes.

In order to reliably identify these arrival time shifts in Lamb wave signals, we again employ the dynamic wavelet fingerprint (DWFP) technique. The patterns in DWFP images then allow us to identify particular Lamb wave modes and directly track subtle shifts in their arrival times.

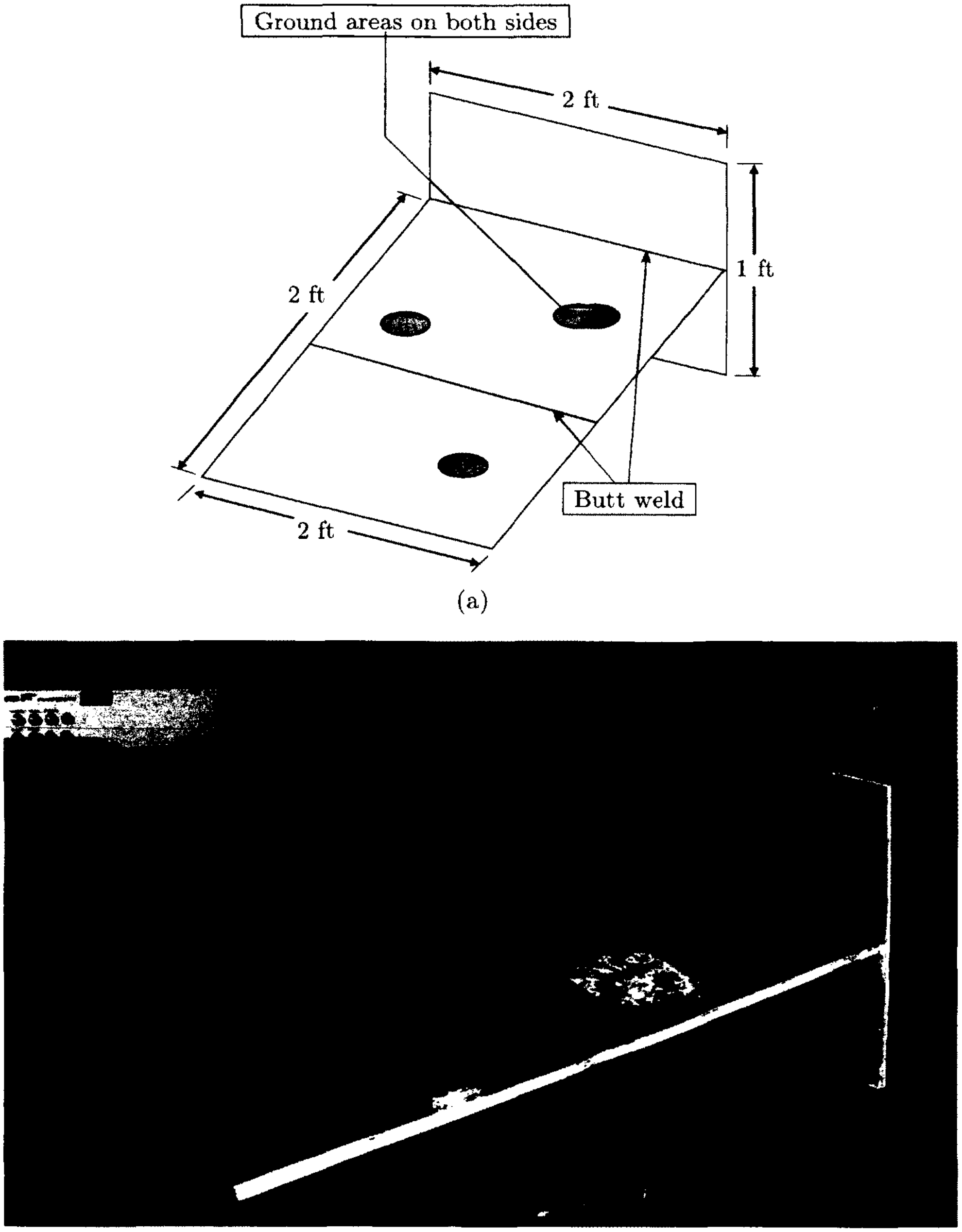
Experimental Setup

Apprentice shipbuilders fabricated a 'T'-shaped plate sample made of 9.5 mm (3/8-inch) thick mild steel for testing, shown in Figure 3.20. The sample was first ground down slightly in several different areas on both the top and bottom surfaces to simulate the effects of corrosion. The sample was then covered in yellow paint, and a 1-inch thick green foam layer was bonded to both the top and bottom surfaces. The sample was intended to be representative of bulkhead sections found in active Navy ships.

Shear wave contact transducers in a parallel pitch/catch scanning configuration were used to systematically scan the full length of the T-plate. A Matec[3] TB1000 pulser/receiver was paired with a Gage[4] CS8012a A/D digitizer to collect data. We

[3] Northborough, MA (`http://www.matec.com/`)

[4] Lockport, IL (`http://www.gage-applied.com/`)



(a)

(b)

Figure 3.20: A 9.5 mm (3/8-inch) thick steel bulkhead sample used in this analysis (a). Several areas on the sample were ground down to introduce simulate wastage or corrosion. The sample was then covered in yellow paint and then a 1 inch thick insulating layer of green foam was bonded to the surface (b). In this picture, a section of the green foam has been removed over one of the several underlying thinned regions.

selected a 0.78 MHz tone-burst excitation pulse for this inspection, giving a frequency-thickness product of 7.4 MHz-mm. This corresponds to an area of the dispersion curve where the S_2 mode is fastest and its group velocity is at a maximum, shown in Figure 3.21. As the thickness of the plate decreases, i.e. a corroded region, the frequency-thickness will decrease resulting in a shifted, slower S_2 group velocity and therefore a later arrival time of the S_2 mode. Because the entire T-plate sample was covered in the thick foam insulation, we removed a narrow strip from the edges of the plate in order to have direct contact between the transducers and the material. The remaining foam was left on the sample for scanning. This insulation removal could be avoided if transducers were bonded to the material during the construction process. The transducers were stepped in parallel along opposing exposed edges of the T-plate in 1 cm steps through 29 total locations, covering the full length of the sample (projection 1). The sample was rotated 90° and the process was repeated for the remaining two edges (projection 2). This is illustrated in Figure 3.22.

In order to extract mode arrivals from the raw waveforms, we used the DWFP technique. We first filtered the raw waveforms with a 3rd-order Coiflets mother wavelet. We then used the DWFP to generate 2D fingerprint images. These are used to identify the features of interest, here the S_2 mode arrival, in the signals. Proper mode selection is especially important for this plate sample because of the thick rubbery coating on each surface. Some modes are strongly attenuated by such coatings, although we found that the S_2 mode was able to both propagate well and detect the thinning flaws.

Results

The raw waveforms were converted to DWFP fingerprint images for analysis. The mode of interest is the first arriving S_2, so the signals were windowed around the

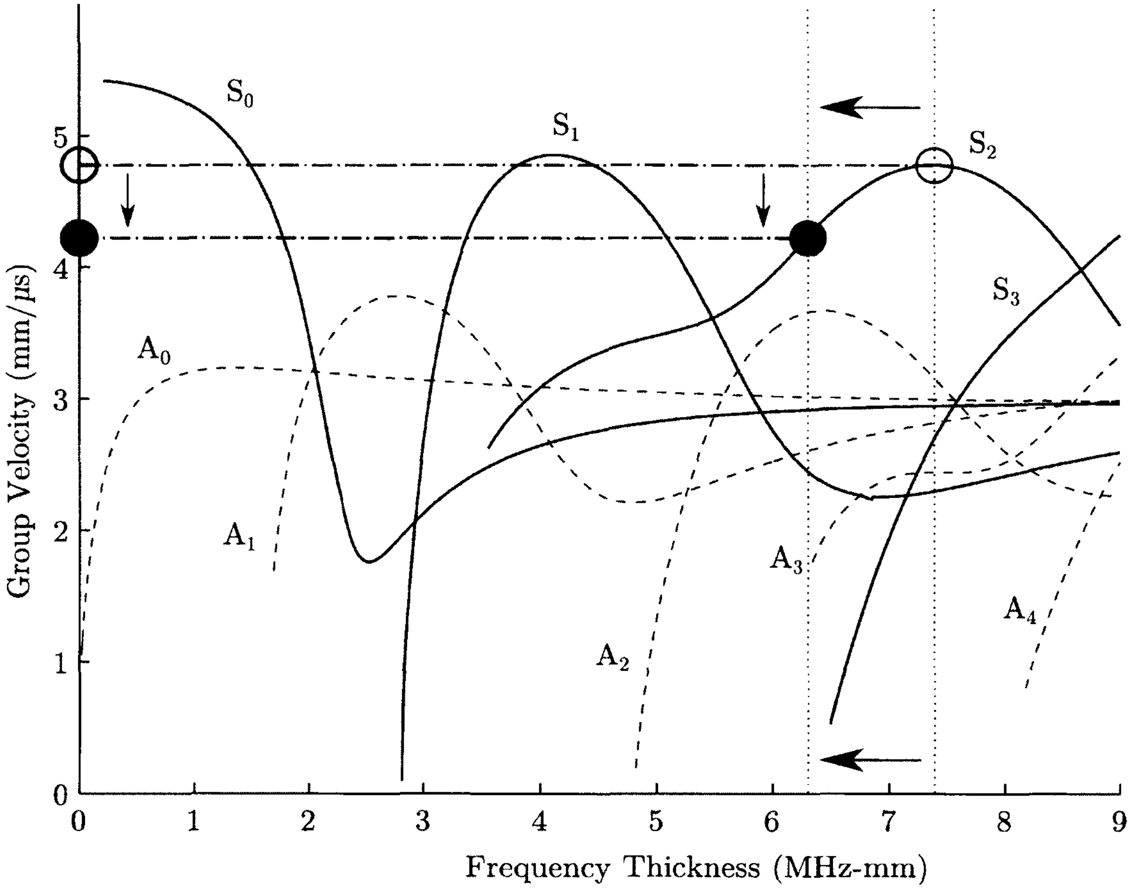


Figure 3.21: Group velocity dispersion curve for steel. At a frequency-thickness product of 7.4 MHz-mm, the S_2 mode is the fastest and therefore first arriving mode. If the wave propagates through an area of reduced thickness, the frequency-thickness product drops as well. This results in a slower S_2 mode velocity, and therefore a later arrival time.

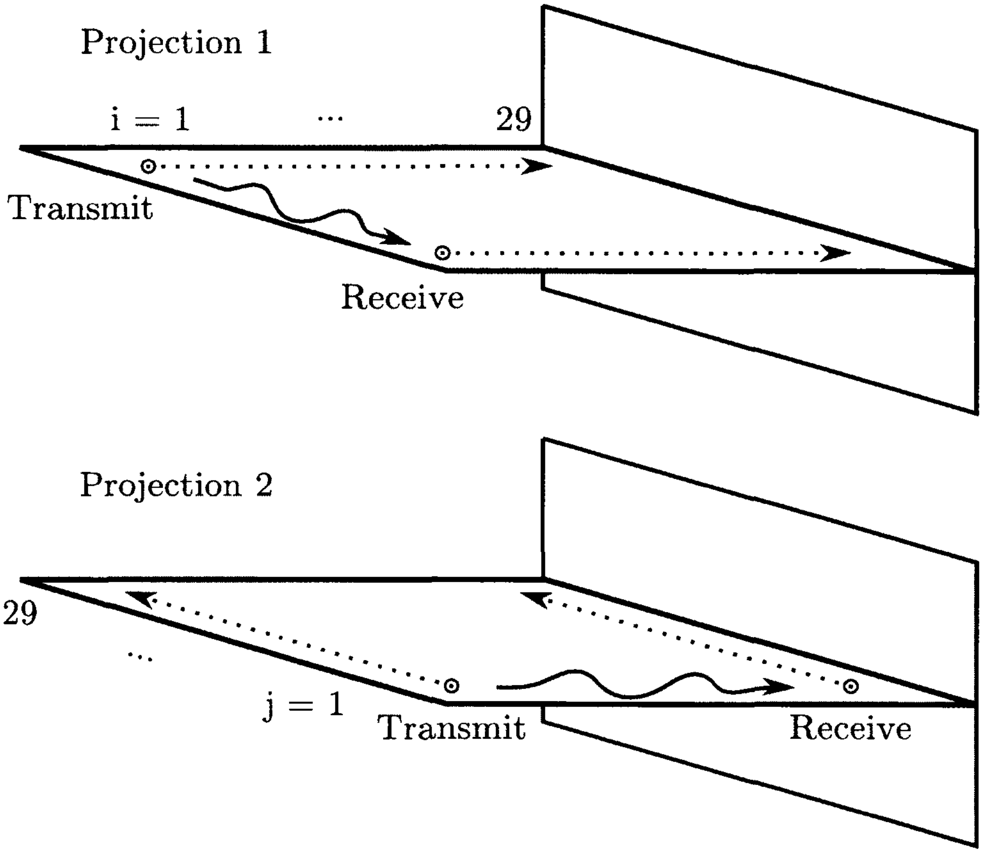


Figure 3.22: Scanning layout of the steel bulkhead section. Two transducers are stepped in parallel down opposing edges of the sample, collecting a pitch/catch signal at each location (top). Their orientation is then rotated 90° relative to the sample, and they are similarly stepped down the remaining two edges of the sample (bottom).

expected S_2 arrival time in order to observe any changes due to damage. A raw waveform is shown in Figure 3.23, with a windowed DWFP representation for both an unflawed raypath as well as one that passes through a flaw on the test sample. In each DWFP image, the red circle and corresponding red triangle indicate the automatically identified S_2 arrival time, provided to the right of each image. It was found that several areas along the length of the plate in each orientation had easily identifiable changes in S_2 mode arrival time from the 132.0 μs arrival time for the unflawed waveforms.

We can apply a simple threshold to these extracted mode arrival times, where any waveforms with arrival times later than 134.0 μs are labeled as "flawed", and any that arrive before this threshold are labeled as "unflawed". Since we collected data along two spatial directions in projections 1 and 2, we can map out the flawed vs. unflawed raypaths geometrically and identify any hot-spots where raypaths in both orientations identify a suspected flaw. This is illustrated in Figure 3.24, where "flawed" waveforms from each projection are indicated by gray, and any spatial areas that indicated "flaw" in *both* orientations highlighed in red. We also provide a photograph of the actual sample with the foam insulation completely removed for final identification of the flawed regions, indicated by the blue/white ovals in each subfigure. Excellent agreement was found between the suspected flaw locations and the sample's actual flaws.

It should be noted that these results were collected by with transducers stepping in parallel (keeping straight across from each other) in two orthogonal directions, allowing us to do a reasonably good job of localizing and to some extent sizing the flaws. Two of the expected flaw areas are over-sized and there is one 'ghost flaw' which isn't actually present, but is due to an artifact of the 'shadows' created by two alternate flaws. In order to more accurately localize and size flaws, it is necessary

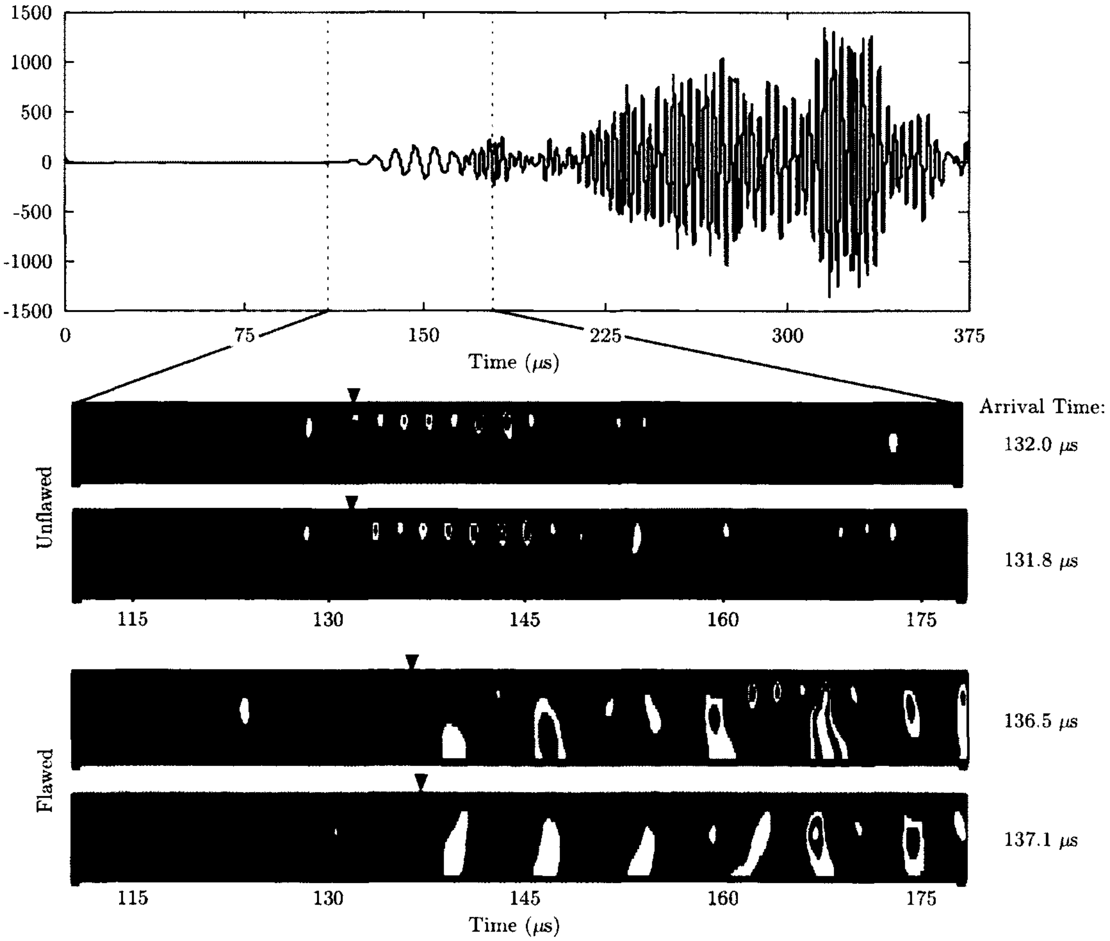
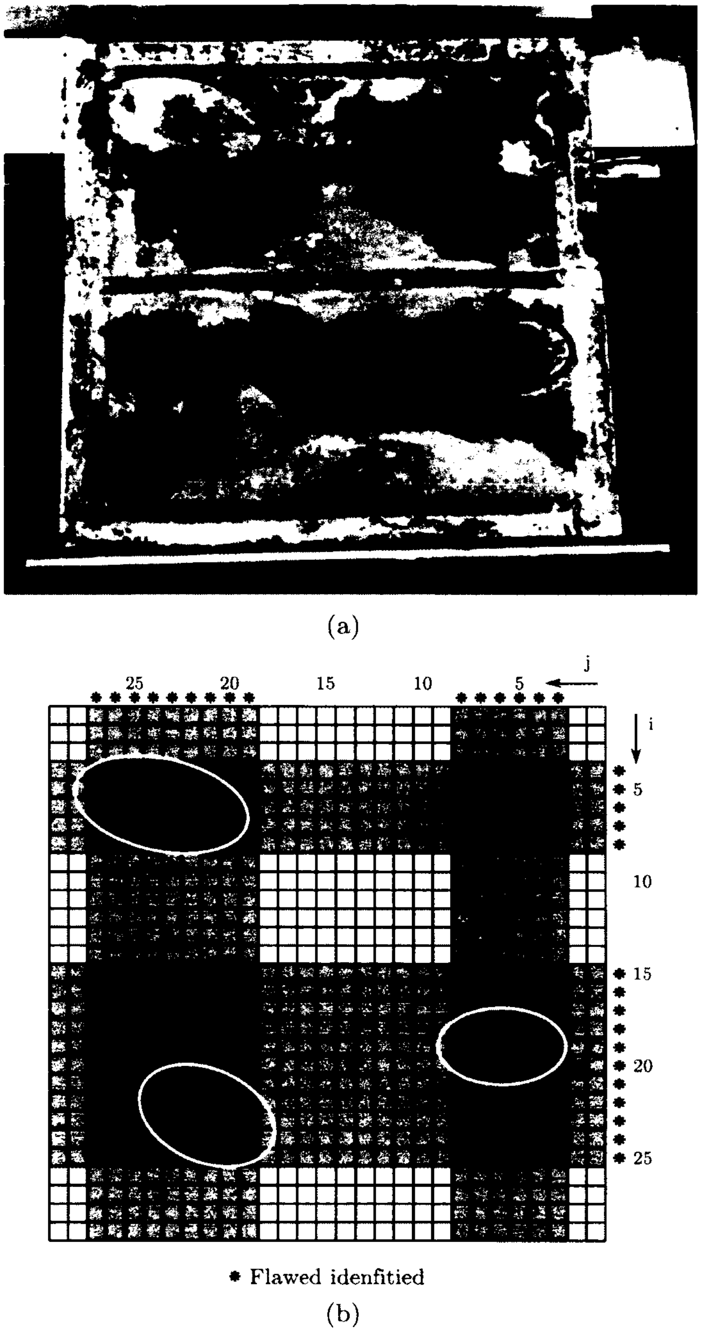


Figure 3.23: A raw Lamb wave signal (top) is first windowed around the region of interest, here the first mode arrival time. DWFP images of this region can then be compared directly between unflawed (middle) and flawed (bottom) signals. A simple tracking of the mode arrival time can be applied, shown here as the red dot and triangle with corresponding arrival time on the right of each image, allowing for identification of any flawed regions.



(a)

(b)

Figure 3.24: Spatially overlaying the individual raypaths onto a grid of the bulkhead sample and thresholding their first mode arrival time, we can see agreement between our experimentally identified "flawed" areas highlighted in red and the known thinned regions of the plate identified by the blue (top) and white (bottom) ellipses.

incorporate information from Lamb wave raypaths at other angles. The formal way to do this is called Lamb wave tomography, which will be discussed in later sections.

3.3 Summary

In this chapter, we have demonstrated the usefulness of the dynamic wavelet fingerprinting technique for analysis of time-domain signals, with two specific applications in the field of structural health monitoring provided. We have discussed the advantages of time-frequency signal representations, and the specific subset associated with time-scale wavelet transformations. We demonstrated how the DWFP technique can be used to automatically identify differences between dents and subtle surface cracks in aircraft-grade aluminum through a combination of pitch/catch and pulse/echo inspection configurations. Combined with straightforward image analysis techniques identifying and tracking specific features of interest within these DWFP images, we have shown how to implement a low-power, portable Rayleigh-Lamb wave inspection system for mapping flaws in airplane fuselages caused by incidental contact on runways. Additionally, we have presented a similar approach for the identification and localization of corrosion in marine structures. Even when the material is underneath a bonded layer of insulation, the guided wave modes were shown to reliably propagate the full length of the sample without significant distortion of the wavelet fingerprints. Multiple simulated corrosion regions were identified using DWFP image analysis, again by identifying a specific feature of interest and tracking it.

These examples both make use of a DWFP feature identification technique, where the feature identified is specific to the application at hand. This is an important point, in that there is no one 'magic' feature we can identify that works across all applications. Here we had insight into the underlying physics of Lamb wave propagation, and were able to predict what types of features in the signals would correlate with the changes we were interested in identifying, i.e. changes in mode arrival time or structure due to some form of material damage. For many time-domain signals we either do not have prior knowledge of expected signal changes, or they are too complicated for analytical solutions that might provide insight into the physical interaction at hand. The DWFP analysis technique provides enough freedom for generating and isolating useful features of interest, however only an exhaustive search would guarantee the inclusion of the best features for a given application. Additionally, some feature changes may not be related to their position in time. For example, the shape of the fingerprint itself may change as a result of some physical interaction. Advanced processing techniques are therefore required for use in conjunction with the power behind the DWFP image representation. In the next chapter, we apply this DWFP framework to RFID signal analysis and introduce pattern classification routines to distinguish between extracted DWFP features.

Chapter 4

Classification of Radio-Frequency Identification (RFID) Tags with Wavelet Fingerprinting

4.1 Introduction

Radio frequency identification (RFID) tags are widespread throughout the modern world, commonly used in retail, aviation, health care, and logistics [92]. As the price of RFID technology decreases with advancements in manufacturing techniques [93], new implementations of RFID technology will continue to arise. The embedding of RFID technology into currency, for example, is being developed overseas to potentially cut down on counterfeiting [94]. Naturally, the security of these RF devices has become a primary concern. Using techniques that range in complexity from simple eavesdropping to reverse engineering [95], researchers have shown authentication vulnerabilities in a wide range of current RFID applications for personal identification and security purposes, with successful cloning attacks made on proximity cards [96], credit cards [97], and even electronic passports [98].

Basic passive RFID tags, which lack the resources to perform most forms of cryptographic security measures, are especially susceptible to privacy and authentication attacks because they have no explicit counterfeiting protections built in. These low-cost tags are the type found in most retail applications, where the low price favors the large quantity of tags required. One implementation of passive RFID tags is to act as a replacement for barcodes. Instead of relaying a sequence of numbers identifying only the type of object a barcode is attached to, RFID tags use an Electronic Product Code (EPC) containing not only information about the type of object, but also a unique serial number used to individually distinguish the object. RFID tags also eliminate the need for line-of-sight scanning that barcodes have, avoiding scanning orientation requirements. In a retail setting, these RFID rags are being explored for point-of-sale terminals capable of scanning all items in a passing shopping cart simultaneously [99]. Without security measures, however, it is straightforward to surreptitiously obtain the memory content of these basic RFID tags and reproduce a cloned signal [100].

An emerging subset of RFID short-range wireless communication technology is near field communication (NFC), operating within the high-frequency RFID band at 13.56 MHz. Compatible with already existing RFID infrastructures, NFC involves an initiator that generates an RF field and a passive target, although interactions between two powered devices are possible. The smartphone industry is one of the leading areas for NFC research, as many manufacturers have begun putting NFC technology to their products. With applications enabling users to pay for items such as groceries and subway tickets by waving their phone in front a machine, NFC payment systems are an attractive alternative to the multitude of credit cards available today [101]. Similarly, NFC-equipped mobile phones are being explored for use as boarding passes, where the passenger can swipe their handset like a card, even when its batteries are dead [102].

A variety of approaches exist to solve this problem of RFID signal authentication, in which an RFID system identifies an RFID tag as being legitimate as opposed to a fraudulent copy. One such method involves the introduction of alternate tag-reader protocols, including the installation of a random number generator in the reader and tag [103], a physical proximity measure when scanning multiple tags simultaneously [104], or re-purposing the kill PIN in an RFID tag, which normally authorizes the deactivation of the tag [105].

Rather than changing the current tag-reader protocols, we approach this issue of RFID tag authentication by applying a wavelet-based RF fingerprinting technique, utilizing the physical layer of RF communication. The goal is to identify unique signatures in the RF signal that provide hardware specific information. First pioneered to identify cellular phones by their transmission characteristics [106], RF fingerprinting has been recently explored for wireless networking devices [107], wired Ethernet cards [108], universal software radio peripherals (USRP) [109] and RFID devices [64, 110–112].

Our work builds on that of Bertoncini et al. [64], in which a classification routine was developed using a novel wavelet-based feature set to identify 150 RFID tags collected with fixed tag orientation and distance relative to the reader with RF shielding. That data set, however, was collected in an artificially protected environment and did not include physical proximity variations relative to the reader, one of the most commonly exploited benefits of RFID technology over existing barcodes. The resulting classifier performance therefore can't be expected to translate to real-world situations. Our goal is to collect signals from a set of 40 RFID tags with identical Electronic Product Codes (EPC) at a variety of orientation and RFID reader frequencies as well as over several days to test the robustness of the classifier. The effects of tag damage in the form of water submersion and physical crumpling are also briefly explored. Unlike Bertoncini et al., we use a low-cost USRP to record the RF signals in an unshielded RF environment, resulting in more realistic conditions and SNR values than previously examined.

4.2 Classification Overview

The application of pattern classification for individual RFID tag identification begins with data collection, where each individual RFID tag is read and the EPC regions are windowed and extracted from all of the tag-reader events. A feature space is formed by collecting a variety of measurements from each of the EPC regions. Feature selection then reduces this feature space to a more optimal subset, removing irrelevant features. Once a data set has been finalized, it is then split into training and testing sets via a resampling algorithm, and the classifier is trained on the training set and tested on the testing set. The classifier output is used to predict a finalized class label for the testing set, and the classifier's performance can be evaluated.

Each tag is given an individual class label, however we are only interested in whether or not a new signal corresponds to an EPC from the specific tag of interest. The goal for this application is to identify false, cloned signals trying to emulate the original tag. We therefore implement a binary one-against-one classification routine, where we consider one individual tag at a time (the classifier tag), and all other tags (the testing tags) are tested against it one at a time. This assigns one of two labels to a testing signal, either $\omega = 1$ declaring that the signal corresponds to an EPC from the classifier tag, or $\omega = -1$ indicating that the signal does not correspond to an EPC from the classifier tag.

4.3 Materials and Methods

Avery-Dennison AD-612 RFID tags were used in this study, which follow the EPC-global UHF Class 1 Generation 2 (EPCGen2) standards [113]. There were 40 individual RFID tags available, labeled $AD01, AD02, \ldots, AD40$. The experimental procedure involves writing the same EPC code onto each tag with a Thing Magic Mercury 5e RFID Reader[1] paired with an omni-directional antenna (Laird Technologies[2]). Stand-alone RFID readers sold today perform all of the signal amplification, modulation/demodulation, mixing, etc. in special-purpose hardware. While this is beneficial for standard RFID use where only the demodulated EPC is of interest, it is inadequate for our research because we seek to extract the raw EPC RF signal.

Preliminary work [64] collected raw RF signals through a vector signal analyzer recording 327.50 ms of data at a 3.2 MHz sampling frequency, a laboratory-grade instrument often used in the design and testing of electronic devices. While the vector signal analyzer proved useful for data collection in the preliminary work, it is not a practical tool that could be implemented in real-world applications. We thus explore the use of an alternate RF signal recording device, a software-defined radio (SDR) system. Software-defined radios are beneficial over standard RFID units as they contain their own A/D converters and the majority of their signal processing is software controlled, allowing them to transmit and receive a wide variety of radio protocols based solely on the software used. The SDR system used here is from the Universal Software Radio Peripheral (USRP) family of products developed by Ettus Research LLC[3], specifically the USRP2, paired with a GnuRadio [114] interface. With board schemes and open source drivers widely available, the flexibility of the USRP

[1] Cambridge, MA (`http://www.thingmagic.com`)

[2] St Louis, MI (`http://www.lairdtech.com`)

[3] Mountain View, CA (`http://www.ettus.com`)

system provides a simple and effective solution for our RF interface.

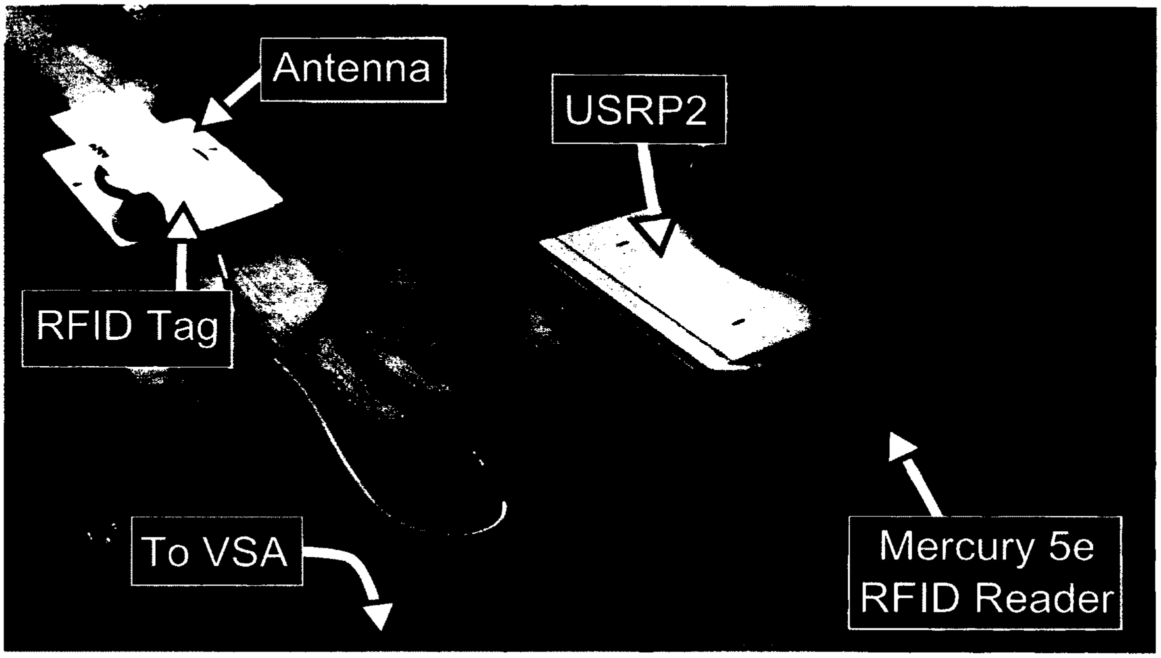
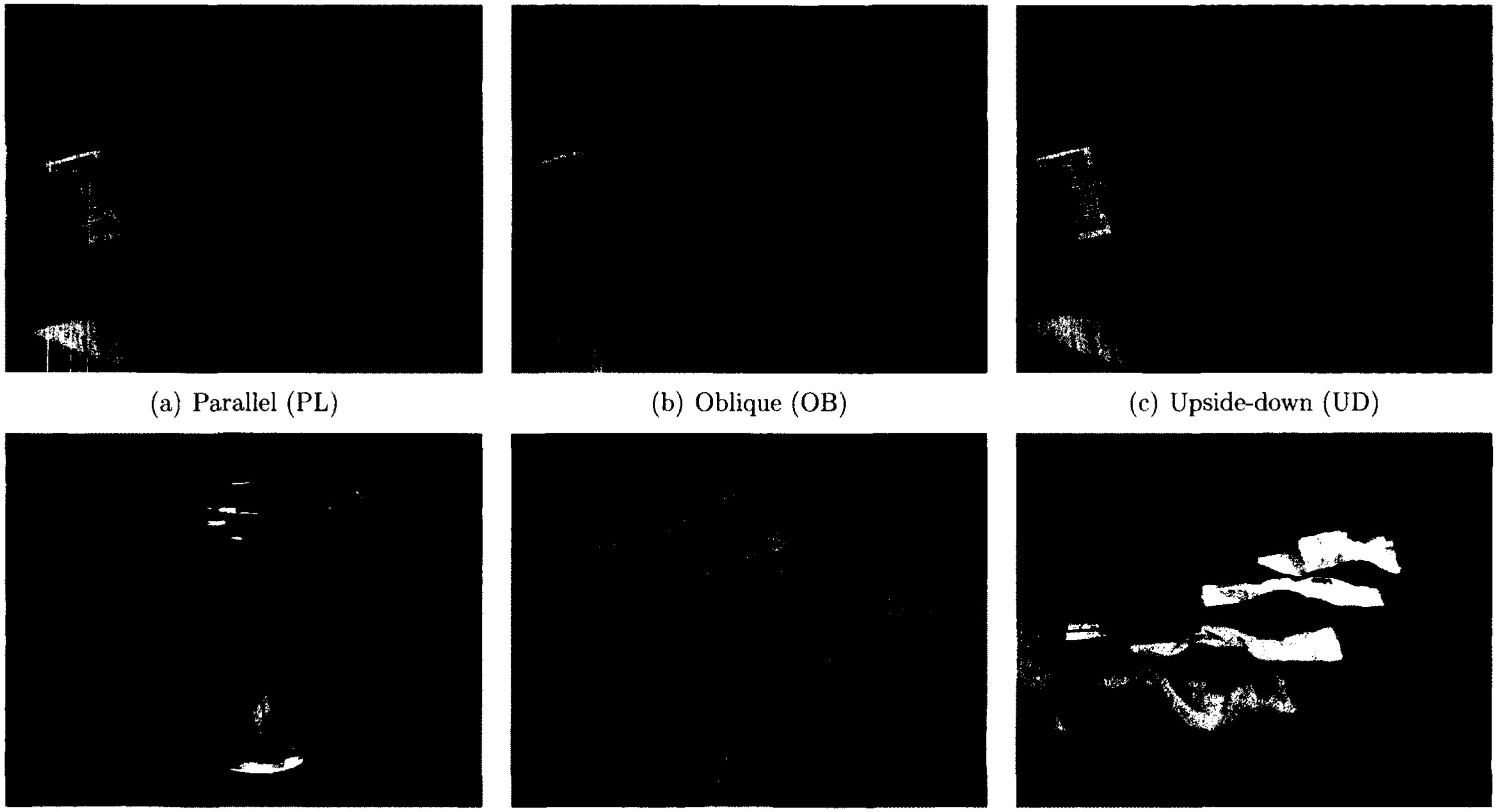


Figure 4.1: Experimental setup for RFID data collection is shown, with the RFID reader, tag, antenna, connection to the VSA, and USRP2 software defined radio.

Data was collected in two separate sessions: the first taking place in an environment that was electromagnetically-shielded over the span of a week, and the second without any shielding taking place at William and Mary (W&M) over the span of two weeks. The first session included twenty-five individual AD-612 RFID tags labeled $AD01 - AD25$. The same EPC code was first written onto each tag with the Thing Magic Mercury 5e RFID Reader, and no further modifications were performed to the tags. Data was collected by placing one tag at a time in a fixed position near the antenna. Tag transmission events were recorded for 3 seconds for each tag using the USRP2, saving all data as a MATLAB format. Each tag was recorded at three different RFID reader operating frequencies (902, 915, and 928 MHz), with three tag orientations relative to the antenna being used at each frequency (parallel (PL), upside-down (UD), and a 45° oblique angle (OB)). The second session of data collection at W&M included a second, independent set of fifteen AD-612 RFID tags labeled $AD26 - AD40$. Similar to before, the same EPC code was first written onto each tag with the Thing Magic Mercury 5e RFID Reader.

The second session was different from the first in that the tags were no longer in a fixed position relative to the antenna, but rather simply held by hand near the antenna. This introduces additional variability into the individual tag-reader events throughout each signal recording. Tag transmission events were again recorded for 3 seconds for each tag. Data was collected at a single operating frequency (902 MHz) with a constant orientation relative to the antenna (parallel (PL)), however data was collected on four separate days allowing for environmental variation (temperature, humidity, etc.). These tags were then split into two subsets, one of which was used for a water damage study while the other was used for a physical damage study. For the water damage, tags $AD26 - AD32$ were submerged in water for three hours, at which point they were patted dry and used to record data (labeled as Wet). They were then allowed to dry overnight, and again used to record data (Wet-to-Dry). For the physical damage, tags $AD33 - AD40$ were first lightly crumpled by hand (Light Damage) and subsequently heavily crumpled (Heavy Damage). Pictures of the tag orientation variations as well as the tag damage can be seen in Figure 4.2.

From these data sets, four separate studies were performed. First, a frequency comparison was run in which the three operating frequencies were used as training and testing data sets for the classifiers, collected while maintaining a constant PL orientation. Second, an orientation comparison was performed in which the three tag orientations were used as training and testing data sets, collected at a constant 902 MHz operating frequency. Third, the four days worth of constant PL and 902 MHz hand-held recordings were used as training and testing data sets. Finally, the classifiers were trained on the four days' worth of recordings, and the additional



(a) Parallel (PL)

(b) Oblique (OB)

(c) Upside-down (UD)

(d) Water Damage

(e) Light Damage

(f) Heavy Damage

Figure 4.2: Tag orientations used for data collection. Parallel (PL) (a), oblique (OB) (b), and upside-down (UD) (c) can be seen, named because of the tag position relative to the antenna. Real-world degradation was also applied to the tags in the form of water submersion (d), as well as both light (e) and heavy (f) physical deformation.

damage data sets were used as testing sets. The specific tags used for each comparison are summarized in Table 4.1.

Table 4.1: There were 40 individual Avery-Dennison AD 612 RFID tags used for this study, split into z subsets $\mathcal{D}_z$ for the various comparisons. Tag numbers τ_i are given for each comparison.

Comparison Type	$\mathcal{D}_z$	Tags Used (τ_i)
Frequency Variations	902, 915, 928 MHz	$i = 1, \ldots, 25$
Orientation Variations	PL, UD, OB	$i = 1, \ldots, 25$
Different Day Recordings	Day 1, 2, 3, 4	$i = 26, \ldots, 40$
Water Damage	Wet, Wet-to-dry	$i = 26, \ldots, 32$
Physical Damage	Light, Heavy damage	$i = 33, \ldots, 40$

4.4 EPC Extraction

In most RFID applications, the RFID reader only has a few seconds to identify a specific tag. For example, consumers would not want a car's key-less entry system that required the user to stand next to the car for half a minute while it interrogated the tag. Rather, the user expects access to their car within a second or two of being within the signal's range. These short transmission times result in only a handful of individual EPCs being transmitted, making it important that each one is extracted efficiently and accurately.

In our data set, each tag's raw recording is a roughly 3 second tag-to-reader communication. During this time there is continuous communication between the antenna and any RFID tags within range. This continuous communication is composed of repeated individual tag-reader (T⇔R) events. The structure and duration of each T⇔R event is pre-defined by the specific protocols used. The AD-612 RFID tags are built to use the EPCGen2 protocols [113], so we can use the inherent structure within these protocols to automatically extract the EPCs within each signal.

Previous attempts at identifying the individual EPC codes within the raw signals involved a fixed-window cross-correlation approach, where a manually extracted EPC region was required for comparison [64]. With smart window sizing, this approach can identify the majority of EPC regions within a signal. As the communication period is shortened and the number of EPCs contained in each recording decreases, however, this technique becomes insufficient.

We have developed an alternative technique that automatically identifies components of the EPCGen2 communication protocols. The new extraction algorithm is outlined in Program 1, with a detailed explanation to follow. It should be noted that the region identified as [EPC+] is a region of the signal that is composed of a preamble which initiates the transmission, a protocol-control element, the EPC itself, as well as a final 16-bit cyclic-redundancy check.

The first step in the EPC extraction routine is to window each raw signal by locating the portions that occur between reader transmission repetitions. These periods of no transmission are referred to here as "downtime" regions. These are the portions of the signal during which the RFID reader is not communicating with the tag at all. An amplitude threshold is sufficient to locate the downtime regions, which divide the raw signal into separate sections, each of which contains several individual T⇔R events. There is another short 'dead' zone between each individual T⇔R event where the RFID reader stops transmitting briefly. Because of this, the upper envelope of the active communication region is taken and another amplitude threshold is applied to identify these dead zones, further windowing the signal into its individual T⇔R events.

Each individual T⇔R event is then processed to extract the individual [EPC+] region within. First, the envelope of the T⇔R event is taken, which highlights the back-and-forth communicating between the tag and the RFID reader. The [EPC+]

Program 1 The EPC extraction routine used to find the EPC regions of interest for analysis.

```
For each tag $AD01$-$AD40$ {
    Raw recorded signal is sent to getEPC.m
         o Find and window "downtime" regions between tag/reader
           communication periods
         o Envelope windowed sections, identify individual T⇔R events
         For each T⇔R event
             • Envelope the signal, locate flat [EPC+]
               region
             • Set start/finish bounds on [EPC+]
             • Return extracted [EPC+] regions
    Each extracted [EPC+] is sent to windowEPC.m
         o Generate artificial Miller (M=4) modulated preamble
         o Locate preamble in recorded signal via cross-correlation
         o Identify all subsequent Miller (M=4) basis functions via cross-
           correlation
         o Extract corresponding bit values
         o Verify extracted bit sequence matches known EPC bit sequence
         o Return start/end locations of EPC region
    Save EPC regions
}
```

region, being the longest in time duration out of all the communication, is relatively consistent in amplitude compared to the up-and-down structure of the signal. Therefore, a region is located that meets a flatness as well as time duration requirements corresponding to this [EPC+]. Once this [EPC+] region is found, an error check is applied that envelopes the region and checks this envelope for outliers that would indicate an incorrectly-chosen area.

The next step in this process is to extract the actual EPC from the larger [EPC+] region. For all Class 1 Gen 2 EPCs, the tags encode the backscattered data using either FM0 baseband or Miller modulation of a subcarrier, the encoding choice made by the reader. The Thing Magic Mercury 5e RFID Reader uses Miller (M=4) encoding, the basis functions of which can be seen in Figure 4.3. The Miller (M=4) preamble is then simulated and cross correlated with the [EPC+] region to determine its location within. From the end of the preamble, the signal is broken up into individual bits, and cross correlation is used to determine which bits are present for the remainder of the signal (positive or negative, 0 or 1). Upon completion, the bit sequence is compared to a second known bit sequence generated from the output of the RFID reader's serial log for verification, shown in Table 4.2. The bounds of this verified bit sequence are then used to window the [EPC+] region down to the EPC only. A single T⇔R event as well as a close-up of a [EPC+] region can be seen in Figure 4.4.

The goal of the classifier is to identify individual RFID tags despite the fact that all the tags are of the same type, from the same manufacturer, and written with the same EPC. The raw RFID signal $s(t)$ is complex-valued, so an amplitude representation $\alpha(t)$ is used for the raw signal [115]. An 'optimal' version of our signal was also reverse-engineered using the known Miller (M=4) encoding methods, labeled $s_0(t)$. We then subtract the raw signal from the optimal representation, producing an EPC

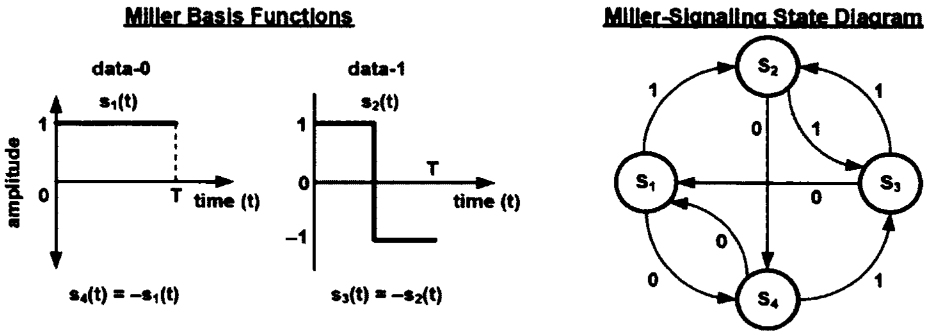


Figure 4.3: Excerpt from the EPC Class 1 Gen 2 Protocols showing the Miller basis functions and a generator state diagram [113].

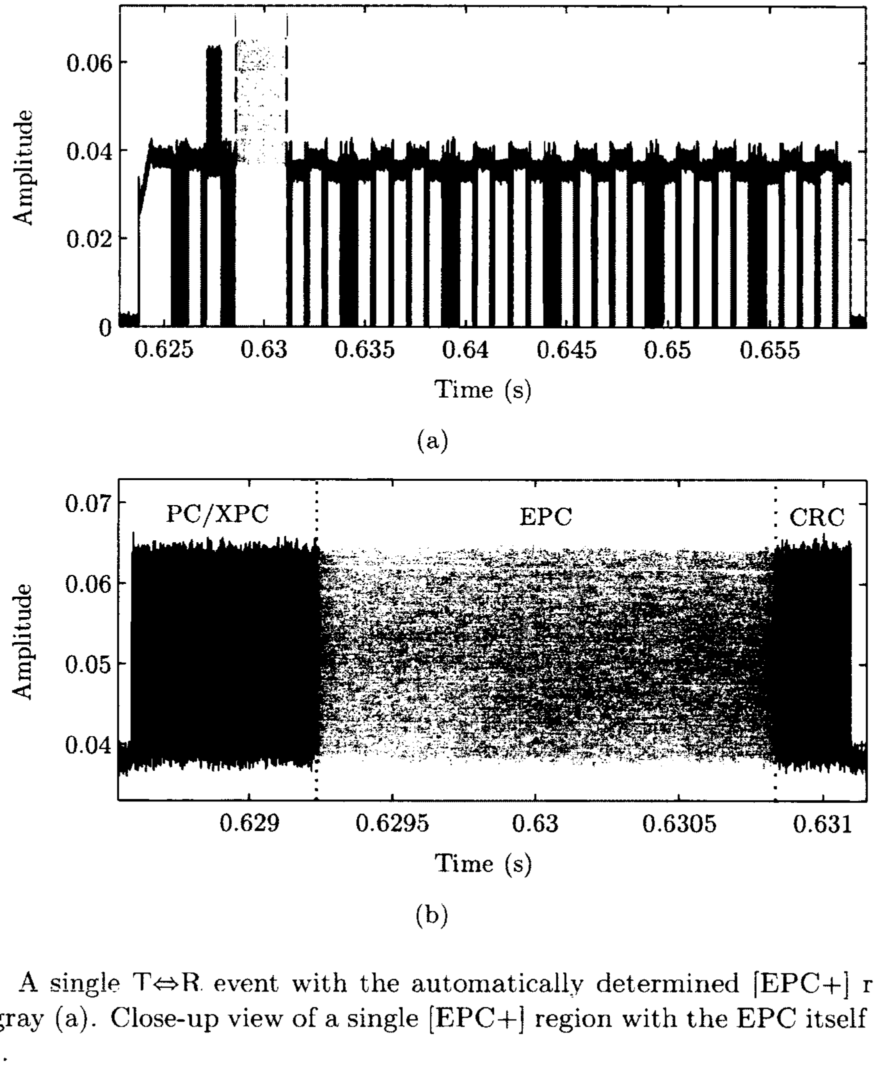


Figure 4.4: A single T⇔R event with the automatically determined [EPC+] region highlighted in gray (a). Close-up view of a single [EPC+] region with the EPC itself highlighted in gray (b).

Table 4.2: Thing Magic Mercury 5e RFID Reader Serial Log

```
(17:05:31.625 - TX(63)): 00 29 CRC:1D26
(17:05:31.671 - RX(63)): 04 29 00 00 00 00 00 01 CRC:9756
(17:05:31.671 - TX(64)): 03 29 00 07 00 CRC:F322
(17:05:31.718 - RX(64)): 19 29 00 00 00 07 00 01 07 72 22 00 80 30 00
   30 08 33 B2 DD D9 01 40 35 05 00 00 42 E7 CRC:4F31
(17:05:31.734 - TX(65)): 00 2A CRC:1D25
(17:05:31.765 - RX(65)): 00 2A 00 00 CRC:01E8
(17:05:31.765 - TX(66)): 00 2A CRC:1D25
(17:05:31.796 - RX(66)): 00 2A 00 00 CRC:01E8
(17:05:31.796 - TX(67)): 05 22 00 00 00 00 FA CRC:0845
(17:05:32.093 - RX(67)): 04 22 00 00 00 00 00 01 CRC:7BA9
(17:05:32.093 - TX(68)): 00 29 CRC:1D26
(17:05:32.140 - RX(68)): 04 29 00 00 00 00 00 01 CRC:9756
(17:05:32.140 - TX(69)): 03 29 00 07 00 CRC:F322
(17:05:32.187 - RX(69)): 19 29 00 00 00 07 00 01 07 72 22 00 80 30 00
   30 08 33 B2 DD D9 01 40 35 05 00 00 42 E7 CRC:4F31
(17:05:32.203 - TX(70)): 00 2A CRC:1D25
(17:05:32.234 - RX(70)): 00 2A 00 00 CRC:01E8
(17:05:32.234 - TX(71)): 00 2A CRC:1D25
(17:05:32.265 - RX(71)): 00 2A 00 00 CRC:01E8
```

EPC code (in hex) is in **bold**.

error signal as well, labeled e_{EPC}. These are summarized by:

$$\begin{aligned} s(t) &= r(t) + ic(t) \\ \alpha(t) &= \sqrt{r^2(t) + c^2(t)} \\ e_{EPC}(t) &= s_0(t) - s(t) \end{aligned} \tag{4.1}$$

This signal processing step of reducing the complex-valued $s(t)$ to either $\alpha(t)$ or $e_{EPC}(t)$ will be referred to as EPC compression. A signal that has been compressed using either one of these methods will be denoted $\hat{s}(t)$ for generality. Figure 4.5 compares the different EPC compression results on a typical complex RFID signal.

4.5 Feature Generation

For each RFID tag τ_i, where i is indexed according to the range given in Table 4.1, the EPC extraction routine produces N-many different EPCs $\hat{s}_{i,j}(t)$, $j = 1, \ldots, N$. Four different methods are then used to extract features from these signals: Dynamic Wavelet Fingerprinting (DWFP), Wavelet Packet Decomposition (WPD), higher-order statistics, and Mellin transform statistics. Using these methods, M feature values are extracted which make up the feature vector $X = x_{i,j,k}$, $k = 1, \ldots, M$. It should be noted here that due to the computation time required to perform this analysis, the computer algorithms were adapted to run on William and Mary's Scientific Computer Cluster[4].

DWFP

The DWFP technique described in Section 3.1.1 are used to generate a subset of the features used for classification. Wavelet-based measurements provide the ability to

[4] `http://www.compsci.wm.edu/SciClone/`

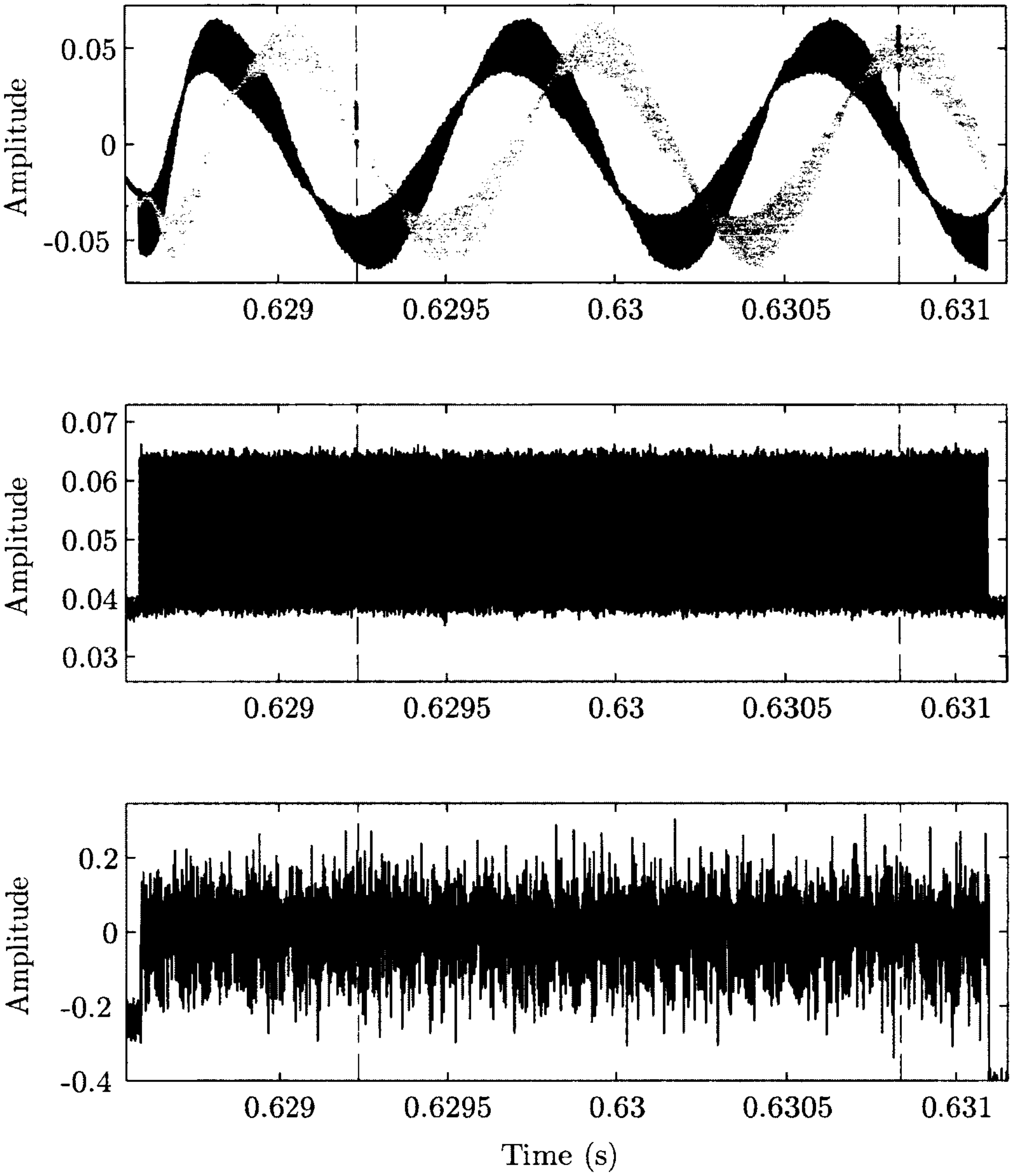


Figure 4.5: The different EPC compression techniques are shown here, displaying the real $r(t)$ and imaginary $c(t)$ components of a raw [EPC+] region (black and gray, respectively, top), the amplitude $\alpha(t)$ (middle), and the EPC error $e_{EPC}(t)$ (bottom). The EPC portion of the [EPC+] signal is bound by vertical red dotted lines.

decompose noisy and complex information and patterns into elementary components. To summarize this process, the DWFP technique first applies a continuous wavelet transform on each original time domain signal $\hat{s}_{i,j}(t)$ [37]. The resulting coefficients are then used to generate "fingerprint"-type images $I_{i,j}(a, b)$ that are coincident in time with the raw signal. Mother wavelets used in this study include the Daubechies-3 (db3), Symelet-5 (sym5), and Meyer (meyr) wavelets, chosen based on preliminary results.

Since pattern classification uses one-dimensional feature vectors to develop decision boundaries for each group of observations, the dimension of the binary fingerprint images $I_{i,j}(a, b)$ that are generated for each EPC signal needs to be reduced. A subset of ν individual values that best represent the signals for classification will be selected. The number ν ($\nu < M$) of DWFP features to select is arbitrary, and can be adjusted based on memory requirements and computation time restraints. For this RFID application, we consider all cases of $\nu \in [1, 5, 10, 15, 20, 50, 75, 100]$.

Using standard MATLAB routines[5], the feature extraction process consists of several steps:

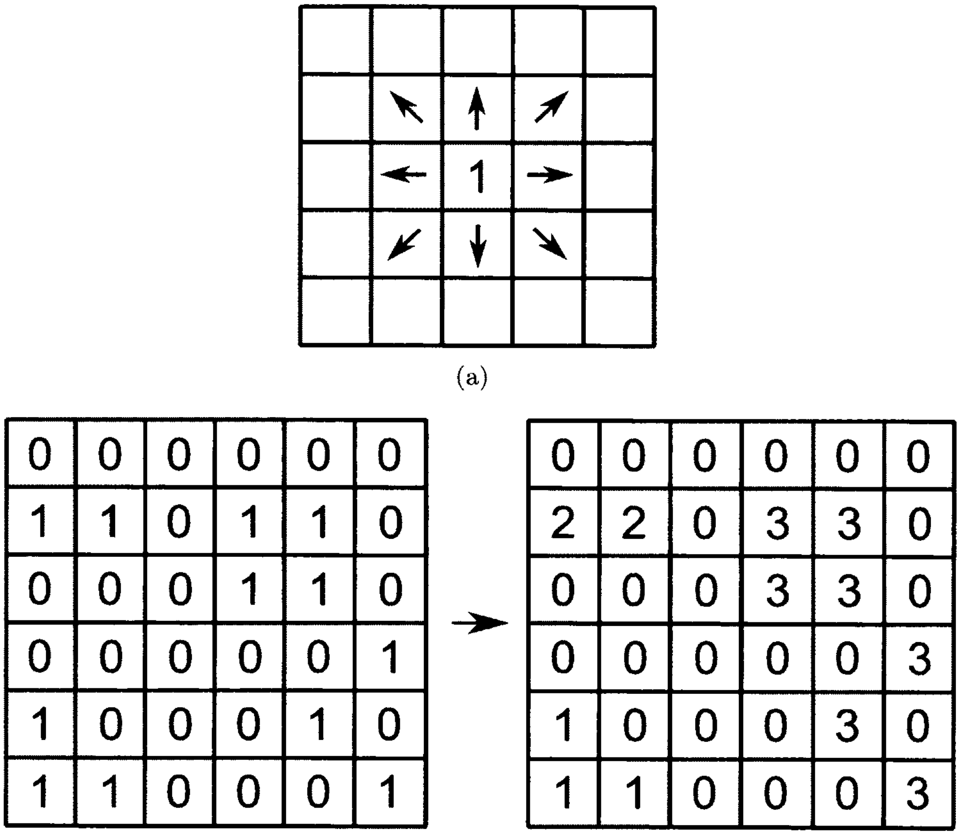
1. Label each binary image with individual values for all sets of connected pixels.
2. Re-label concentric objects centered around a common area (useful for the ring-like features found in the fingerprints).
3. Apply thresholds to remove any insignificant objects in the images.
4. Extract features from each labeled object.
5. Linearly interpolate in time between individual fingerprint locations to generate a smoothed array of feature values.

[5] MATLAB's Image Processing Toolbox (MATLAB, 2008, The Mathworks, Natick, MA.).

6. Identify points in time where the feature values are consistent among individual RFID tags yet separable between different tags.

The binary nature of the images allows us to consider each pixel of the image as having a value of either 1 or 0. The pixels with a value of 0 can be thought of as the background, while pixels with an value of 1 can be though of as the significant pixels. The first step in feature extraction is to assign individual labels to each set of 8-connected components in the image [116], demonstrated in Figure 4.6. Since the fingerprints are often concentric shapes, different concentric "rings" are often not connected to each other, but still are components of the same fingerprint object. Therefore, the second step in the process is to re-label groups of concentric objects using their center of mass, which is the average time-coordinate of each pixel, demonstrated in Figure 4.7. The third step in the feature extraction process is to remove any fingerprint objects from the image whose area (sum of the pixels) is below a particular threshold. Objects that are too small for the computations in later steps are removed, however this threshold is subjective and depends on the mother wavelet used.

At this point in the processing, the image is ready for features to be generated. Twenty-two measurements are made on each remaining fingerprint object, including the area, centroid, diameter of a circle with the same area, Euler number, convex image, solidity, coefficients of 2nd and 4th degree polynomials fit to the fingerprint boundary, as well as major/minor axis length, eccentricity, and orientation of an ellipse that has the same normalized second central moment as the fingerprint. For more details on these features, see Reference [63]. The property measurements result in a sparse property array $\mathcal{P}_{i,j,n}[t]$, where n represents the property index $n = 1, \ldots, 22$, since each extracted value is matched to the time value of the corresponding fingerprint's center of mass. Therefore, these sparse property vectors are linearly interpolated to



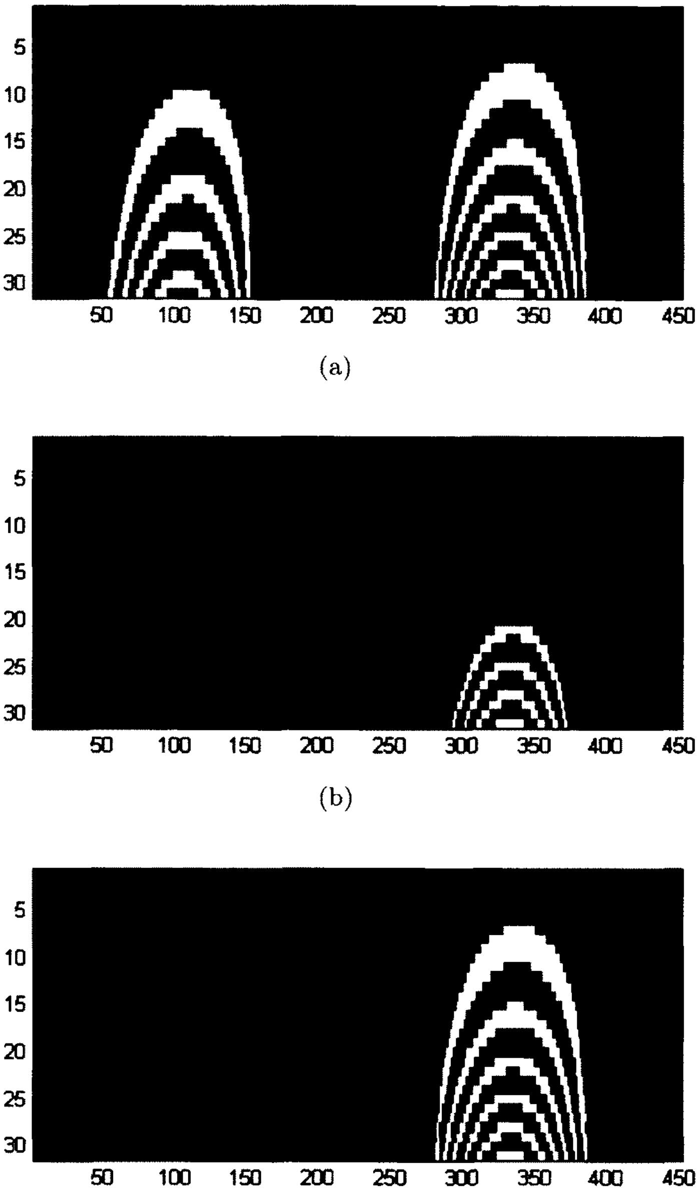
(a)

0	0	0	0	0	0
1	1	0	1	1	0
0	0	0	1	1	0
0	0	0	0	0	1
1	0	0	0	1	0
1	1	0	0	0	1

0	0	0	0	0	0
2	2	0	3	3	0
0	0	0	3	3	0
0	0	0	0	0	3
1	0	0	0	3	0
1	1	0	0	0	3

(b)

Figure 4.6: An example of 8-connectivity (a) and its application on a binary image (b).



(a)

(b)

(c)

Figure 4.7: An example of the fingerprint labeling process. The components of the binary image (a) and the resulting 8-connected components (b), where each label index corresponds to a different index on the 'hot' colormap in this image. Concentric objects are then relabeled (c), resulting in unique labels for each individual fingerprint object, shown here as orange and white fingerprint objects for clarity.

produce a smoothed vector of property values, $\mathcal{P}_{i,j,n}(t)$. This process is shown for a typical time-domain EPC signal in Figure 4.8.

Once an array of fingerprint features for each EPC has been generated, it still needs to be reduced into a single vector of ν-many values to be used for classification. Without this reduction, not only is the feature set too large to process even on a computing cluster, but most of the information contained within it is redundant. Since we are implementing a one-against-one classification scheme, where one testing tag (τ_t) will be compared against features designed to identify one classifier tag (τ_c), we are looking for features values that are consistent among each individual RFID tag, yet separable between different tags.

First, the dimensionality of the property array is reduced by calculating the inter-tag mean property value for each tag τ_i,

$$\mu_{i,n}(t) = \frac{1}{|j|} \sum_j \mathcal{P}_{i,j,n}(t). \tag{4.2}$$

Each inter-tag mean vector is then normalized to the range $[0, 1]$. Next, the difference in inter-tag mean vectors for property n are considered for all binary combinations of tags τ_{i_1}, τ_{i_2},

$$d_n(t) = |\mu_{i_1,n}(t) - \mu_{i_2,n}(t)| \quad \text{for} \quad i_1, i_2 \in i \tag{4.3}$$

for values of i shown in Table 4.1. We are left with a single vector representing the average intra-class difference in property n values as a function of time.

Similarly, we compute the standard deviation within each class,

$$\sigma_{i,n}(t) = \sqrt{\frac{1}{|j|} \sum_j (P_{i,j,n}(t) - \mu_{i,n}(t))^2}. \tag{4.4}$$

We next identify the maximum value of standard deviation among all tags τ_i at each

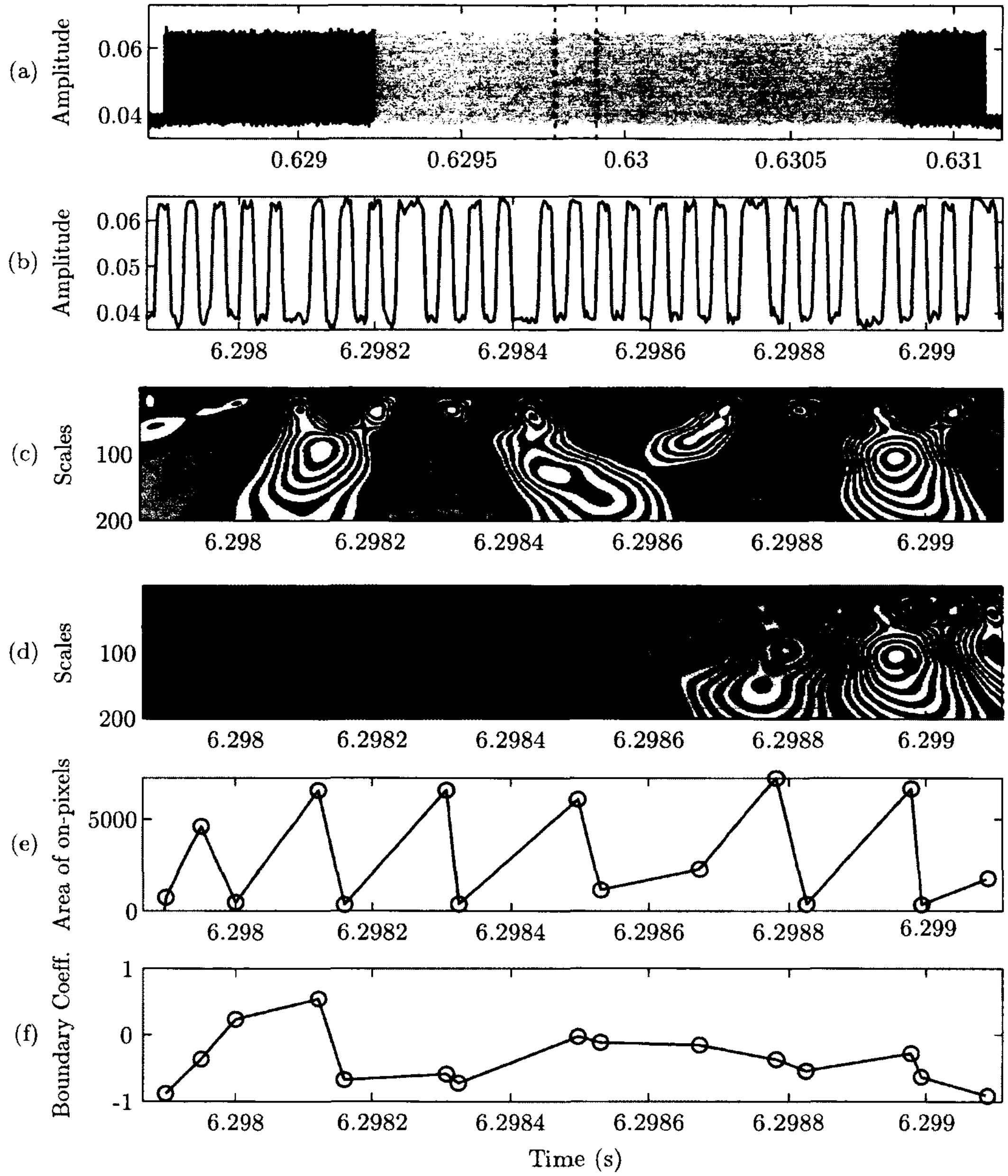


Figure 4.8: The DWFP is applied to an EPC signal $\hat{s}_{i,j}(t)$, shown in gray (a). A close-up of the signal is shown for clarity (b), from which the fingerprint image (c) is generated, shown here with white peaks and gray valleys for distinction. Each fingerprint object is individually labeled and localized in both time and scale (d). A variety of measures are extracted from each fingerprint and interpolated in time, including the area of the on-pixels for each object (e), as well as the coefficients of a fourth-order polynomial $(p_1x^4 + p_2x^3 + p_3x^2 + p_4x + p_5)$ fit to the boundary of each fingerprint object, with coefficients p_3 shown here (f).

point in time t, essentially taking the upper envelope of all values of $\sigma_{i,n}(t)$,

$$\sigma_n(t) = \max_i (\sigma_{i,n}(t)). \tag{4.5}$$

Times t_m, where $m = 1, \ldots, \nu$, are then identified for each property n at which the the average intra-class difference $d_n(t_m)$ is high while the inter-class standard deviation $\sigma_n(t_m)$ remains low. The resulting DWFP feature vector for EPC signal $\hat{s}_{i,j}(t)$ is

$$x_{i,j,k_m} = \mathcal{P}_{i,j,n_m}(t_m).$$

Wavelet Packet Decomposition

Another wavelet-based feature used in classification is generated by Wavelet Packet Decomposition [25]. First, each EPC signal is filtered using a stationary wavelet transform and removing the first 3 levels of detail as well as the highest approximation level. A Wavelet Packet Transform (WPT) is applied to the filtered waveform with a specified mother wavelet and the number of levels to decompose the waveform, generating a tree of coefficients similar in nature to the continuous wavelet transform. From the WPT tree, a vector containing the percentages of energy corresponding to the T terminal nodes of the tree is computed, known as the wavelet packet energy. Because the WPT is an orthonormal transform, the entire energy of the signal is preserved in these terminal nodes [117]. The energy matrix E_j for each RFID tag τ_i for can then be represented as

$$E_i = [\mathbf{e}_{1,i}, \mathbf{e}_{2,i}, \ldots, \mathbf{e}_{N,i}] \tag{4.6}$$

where N is the number of EPCs extracted from tag τ_i, and $e_{i,j}[b]$ is the energy from bin number $b = 1, \ldots, T$ of the energy map for signal $j = 1, \ldots, N$. Singular value decomposition is then applied to each energy matrix E_i:

$$E_i = U_i \Sigma_i V_i^* \tag{4.7}$$

where U_i is composed of T-element left singular column vectors $\mathbf{u}_{b,i}$

$$U_i = [\mathbf{u}_{1,i}, \mathbf{u}_{2,i}, \ldots, \mathbf{u}_{T,i},] \,. \tag{4.8}$$

The Σ_i matrix is a $T \times N$ singular value matrix. The row space and nullspace of E_i are defined in the $N \times N$ matrix V_i^*, and are not used in the analysis of the energy maps. For the energy matrices E_i, we found that there was a dominant singular value relative to the second highest singular value, implying that there was a dominant representative energy vector corresponding to the first singular vector $\mathbf{u}_{1,i}$. From the set of all singular vectors $\mathbf{u}_{b,i}$, the significant bins that have energies above a given threshold are identified. The threshold is lowered until all the vectors return a common significant bin. Finally, the WPT elements corresponding to the extracted bin are used as features. In the case of multiple bins being selected, all corresponding WPT elements are included in the feature set. Wavelet packet decomposition uses redundant basis functions and can therefore provide arbitrary time-frequency resolution details, improving upon the wavelet transform when analyzing signals containing close, high frequency components.

Statistical Features

Several statistical features were generated from the raw EPC signals $\hat{s}_{i,j}(t)$ [20]:

1. The mean of the raw signal

$$\mu_{i,j} = \frac{1}{|\hat{s}|}\sum_t \hat{s}_{i,j}(t)$$

   where $|\hat{s}|$ is the length of $\hat{s}_{i,j}(t)$

2. The maximum cross-correlation of $\hat{s}_{i,j}(t)$ with another EPC from the same tag, $\hat{s}_{i,k}(t)$, where $\tau_j = \tau_k$

$$\max\left(\sum_t \hat{s}^*_{i,j}(t)\hat{s}_{i,k}(t+\tau)\right)$$

3. The Shannon entropy

$$\sum_t \hat{s}^2_{i,j}(t)\ln(\hat{s}^2_{i,j}(t))$$

4. The unbiased sample variance

$$\frac{1}{|\hat{s}|-1}\sum_t (\hat{s}_{i,j}(t) - \mu_{i,j})^2$$

5. The skewness (third central moment)

$$\frac{1}{\sigma^3_{i,j}|\hat{s}|}\sum_t (\hat{s}_{i,j}(t) - \mu_{i,j})^3$$

6. The kurtosis (fourth central moment)

$$\kappa_{i,j} = \frac{1}{\sigma^4_{i,j}|\hat{s}|}\sum_t (\hat{s}_j(t) - \mu_{i,j})^4$$

Statistical moments provide insight by highlighting outliers due to any specific flaw-type signatures found in the data.

Mellin Features

The Mellin transform is an integral transform, closely related to the Fourier transform and the Laplace transform, that can represent a signal in terms of a physical attribute similar to frequency known as scale. The β-Mellin transform is defined as [119]

$$M_f(p) = \int_0^\infty \hat{s}(t) t^{p-1} \, dt, \tag{4.9}$$

for the complex variable $p = -jc + \beta$, with fixed parameter $\beta \in \mathbb{R}$ and independent variable $c \in \mathbb{R}$. This variation of the Mellin transform is used because the β parameter allows for the selection of a variety of more specific transforms. In the case of $\beta = 1/2$, this becomes a scale-invariant transform, meaning invariant to compression or expansion of the time axis while preserving signal energy, defined on the vertical line $p = -jc + 1/2$. This scale transform is defined as

$$D_f(c) = \frac{1}{\sqrt{2\pi}} \int_0^\infty \hat{s}(t) \mathrm{e}^{(-jc-1/2)\ln t} \, dt. \tag{4.10}$$

This transform has the key property of scale invariance, which means that $\hat{s}'$ is a scaled version of a function $\hat{s}$, they will have the same transform magnitude. Variations in each RFID tag's local oscillator can lead to slight but measurable differences in the frequency of the returned RF signal, effectively scaling the signal. Zanetti et al. call this the time interval error (TIE), and extract the TIE directly to use as a feature for individual tag classification [111]. We observed this slight scaling effect in our data and therefore explore the use of a scale-invariant feature extraction technique.

The Mellin transform's relationship with the Fourier transform can be highlighted by setting $\beta = 0$, which results in a logarithmic-time Fourier transform:

$$M_f(c) = \int_{-\infty}^{\infty} \hat{s}(t) e^{-jc(\ln t)}\, d(\ln t). \tag{4.11}$$

Similarly, the scale transform of a function $\hat{s}(t)$ can be defined using the Fourier transform of $g(t) = \hat{s}(\mathrm{e}^t)$:

$$M_f(c) = \int_{-\infty}^{\infty} g(t) e^{-jct}\, d(t) = \mathcal{F}(g(t)). \tag{4.12}$$

References [119, 120] discuss the complexities associated with discretizing the fast Mellin transform (FMT) algorithm, as well as provide a MATLAB-based implementation[6]. The first step in implementing this is to define both an exponential sampling step along with the number of samples needed for a given signal in order to exponentially resample it, an example of which can be seen in Figure 4.9. Once the exponential axis has been defined, an exponential point-by-point multiplication with the original signal is performed. A fast Fourier transform (FFT) is then computed, followed by an energy normalization step. This process is summarized in Figure 4.10.

Once the Mellin transform is computed, features are extracted from the resulting Mellin domain including the mean of the Mellin transform, as well as the standard deviation, the variance, the second central moment, the Shannon entropy, the kurtosis, and the skewness of the mean-removed Mellin transform [22, 23].

4.6 Classifier Design

Since the goal of our classification routine is to distinguish individual RFID tags from nominally identical copies, each individual RFID tag is assigned a unique class label.

[6] http://profs.sci.univr.it/~desena/FMT

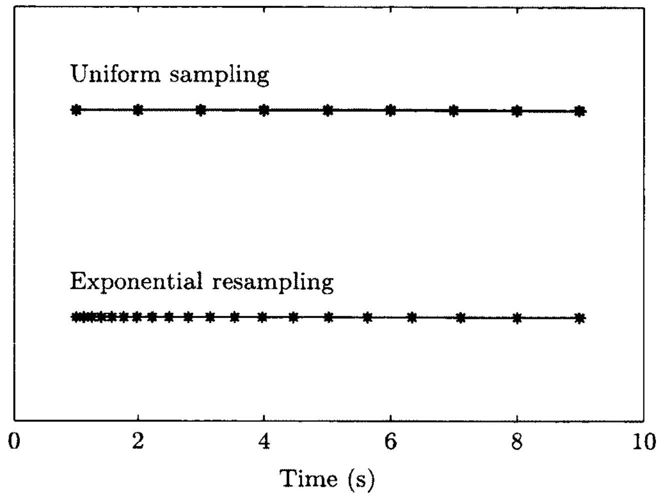


Figure 4.9: An example of uniform sampling (in blue, top) and exponential sampling (in red, bottom).

This results in a multiclass problem, with the number of classes being equivalent to the number of tags being compared. There are two main methods which can be used to address multiclass problems. The first uses classifiers that have multidimensional discriminant functions, which often output classification probabilities for each test object that then need to be reduced for a final classification decision. The second method uses a binary comparison between all possible pairs of classes utilizing a two-class discriminant function, with a voting procedure used to determine final classification. We have discussed in Section 4.2 our choice of the binary classification approach, allowing us to include intrinsically two-class discriminants in our analysis. Therefore, only two tags will be considered against each other at a time, a classifier tag $\tau_c \in \mathcal{D}_R$ and a testing tag $\tau_t \in \mathcal{D}_T$, where $\mathcal{D}_R$ represents the training data set used and $\mathcal{D}_T$ the testing data set, outlined in Table 4.1.

For each binary combination of tags (τ_c, τ_t), a training set (R) is generated composed of feature vectors from k-many EPCs associated with tags τ_c and τ_t from data set $\mathcal{D}_R$. Corresponding known labels (ω_k) are $\omega_k = 1$ when $k \in c$, and $\omega_k = -1$ when

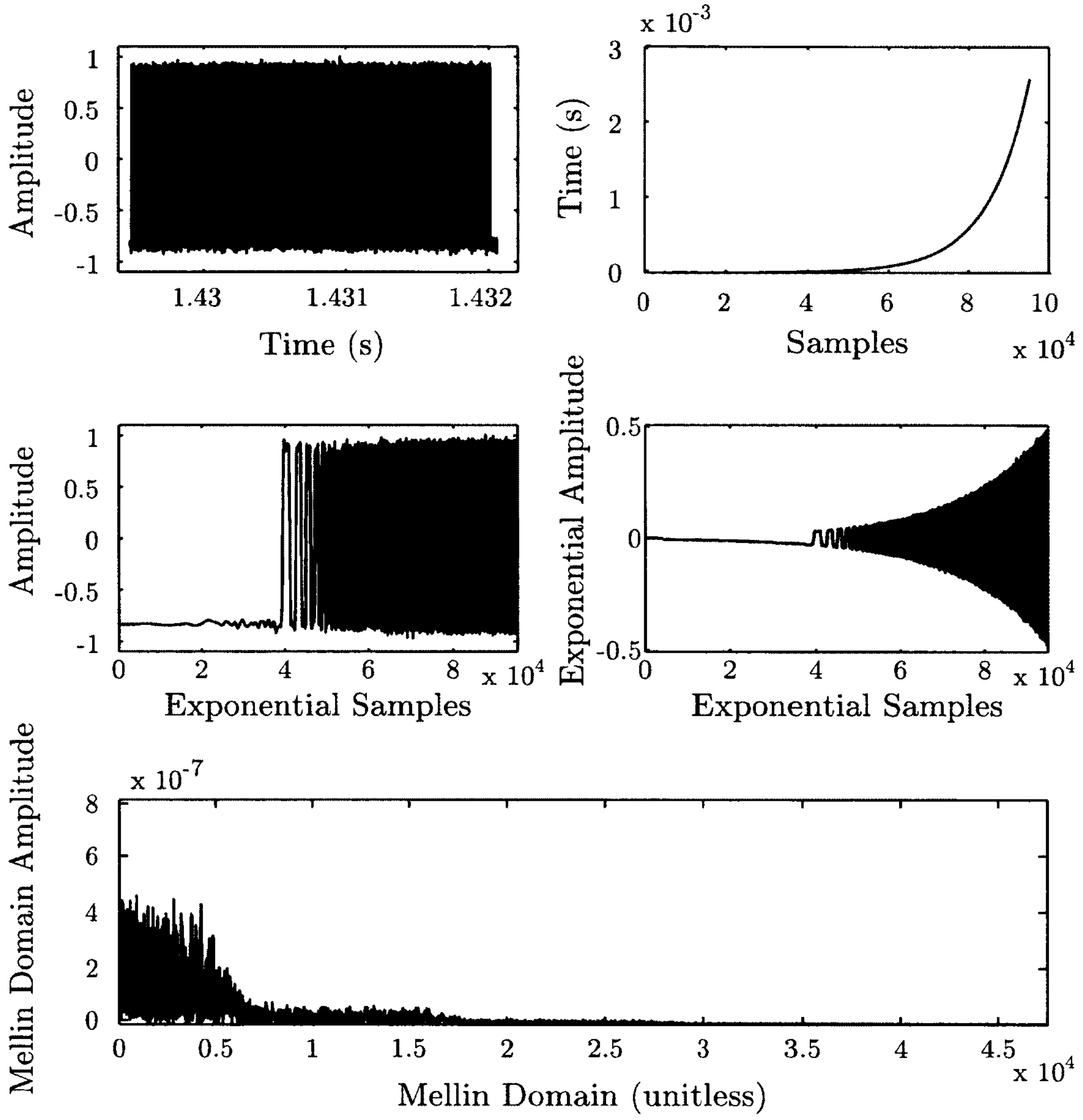


Figure 4.10: A raw EPC signal $s_{i,j}(t)$ (top, left), the exponentially resampled axis (top, right), the signal resampled according to the exponential axis (middle, left), this same signal after point-by-point multiplication of the exponential axis (middle, right), and the resulting Mellin domain representation (bottom).

$k \in t$. The testing set (T) is composed of feature vectors of tag τ_t only from data set $\mathcal{D}_T$, where predicted labels are denoted $y_k = \pm 1$. In other words, the classifier is trained on data from both tags in the training data set and tested on the testing tag only from the testing data set. When $\mathcal{D}_R = \mathcal{D}_T$, which means the classifier is trained and tested on the same data set, a holdout algorithm is used to split the data into R and T [63].

The problem of class imbalance, where the class labels ω_k are unequally distributed (i.e. $|\omega_k = 1| \gg |\omega_k = -1|$, or vice versa), can affect classifier performance and has been a topic of further study by several researchers [121 123]. While the number of EPCs extracted from each tag here does not present a significant natural imbalance as all recordings are approximately the same length in time, it is not necessarily true that the natural distribution between classes, or even a perfect 50:50 distribution, are ideal. To explore the effect of class imbalance on the classifier performance, a variable ρ is introduced here, defined as

$$\rho = \frac{|\omega_k = -1|}{|\omega_k = 1|}, \quad k \in R \tag{4.13}$$

This variable defines the ratio of negative versus positive EPC labels in R, with $\rho \in \mathbb{Z}^+$. When $\rho = 1$, the training set T contains an equal number of EPCs from tags τ_c as it does τ_t where under-sampling is used as necessary for equality. As ρ increases, additional tags are included at random from $\tau_m, m \neq c, t$ with $\omega_m = -1$ until ρ is satisfied. When all of the tags in $\mathcal{D}_R$ are included in the training set, ρ is denoted as 'all'.

The process of selecting which classifiers to use is a difficult problem. The No Free Lunch Theorem states that there is no inherently best classifier for a particular application, and often times in practice several classifiers are compared and contrasted.

As previously discussed, there exists a hierarchy of possible choices that are application dependent. We have previously determined that supervised, statistical pattern classification techniques using both parametric and nonparametric probability-based classifiers are appropriate for consideration.

For parametric classifiers, we include a linear classifier using normal densities (LDC) and a quadratic classifier using normal densities (QDC). For nonparametric classifiers, we include a k-nearest-neighbor classifier (KNNC) for $k = 1, 2, 3$, and a linear support vector machine (SVM) classifier. The mathematical explanations for these classifiers can be found in [1, 2, 4–6, 9, 124–127]. For implementation of these classifier functions, we use routines from the MATLAB toolbox PRTools [126].

For the above classifiers that output densities, a function is applied that converts the output to a proper confidence interval, where the sum of the outcomes is one for every test object. This allows for comparison between classifier outputs. Since each EPC's feature vector is assigned a confidence value for each class, the final label is decided by the highest confidence of all the classes.

4.7 Classifier Evaluation

Since we have implemented a binary classification algorithm, a confusion matrix $\mathcal{L}(c, t)$, where τ_c is the classifier tag and τ_t is the testing tag, can be used to view the results of a given classifier. Each entry in a confusion matrix represents the number of EPCs from the testing tag that are labeled as the classifier tag, denoted by a label of $y_t = 1$, and is given by

$$\mathcal{L}(c, t) = \frac{|y_t = 1|}{|y_t|} \quad \text{when} \quad \tau_c \in R, \tau_t \in T \tag{4.14}$$

A perfect classifier would therefore have values of $\mathcal{L} = 1$ whenever $\tau_c = \tau_t$ (on the diagonal) and values of $\mathcal{L} = 0$ when $\tau_c \neq \tau_t$ (off-diagonal). Given the number of classifier configuration parameters used in this study, it does not make sense to compare individual confusion matrices to each other to determine classifier performance. Each entry of the confusion matrix is a measure of the number of EPCs from each testing set that are determined to belong to each possible training class. We can therefore apply a threshold h to each confusion matrix, where the value of h lies within the range $[0, 1]$. All confusion matrix entries that are above this threshold are *positive* matches for class membership, and all entries below the threshold are identified as *negative* matches for class membership. It follows that we can determine the number of false positive (f_+), false negative, (f_-), true positive(t_+), and true negative (t_-) rates for each confusion matrix, given by

$$\begin{aligned} f_+(h) &= |\mathcal{L}(c,t) > h|, c \neq t \\ t_+(h) &= |\mathcal{L}(c,t) > h|, c = t \\ f_-(h) &= |\mathcal{L}(c,t) \leq h|, c = t \\ t_-(h) &= |\mathcal{L}(c,t) \leq h|, c \neq t \end{aligned} \tag{4.15}$$

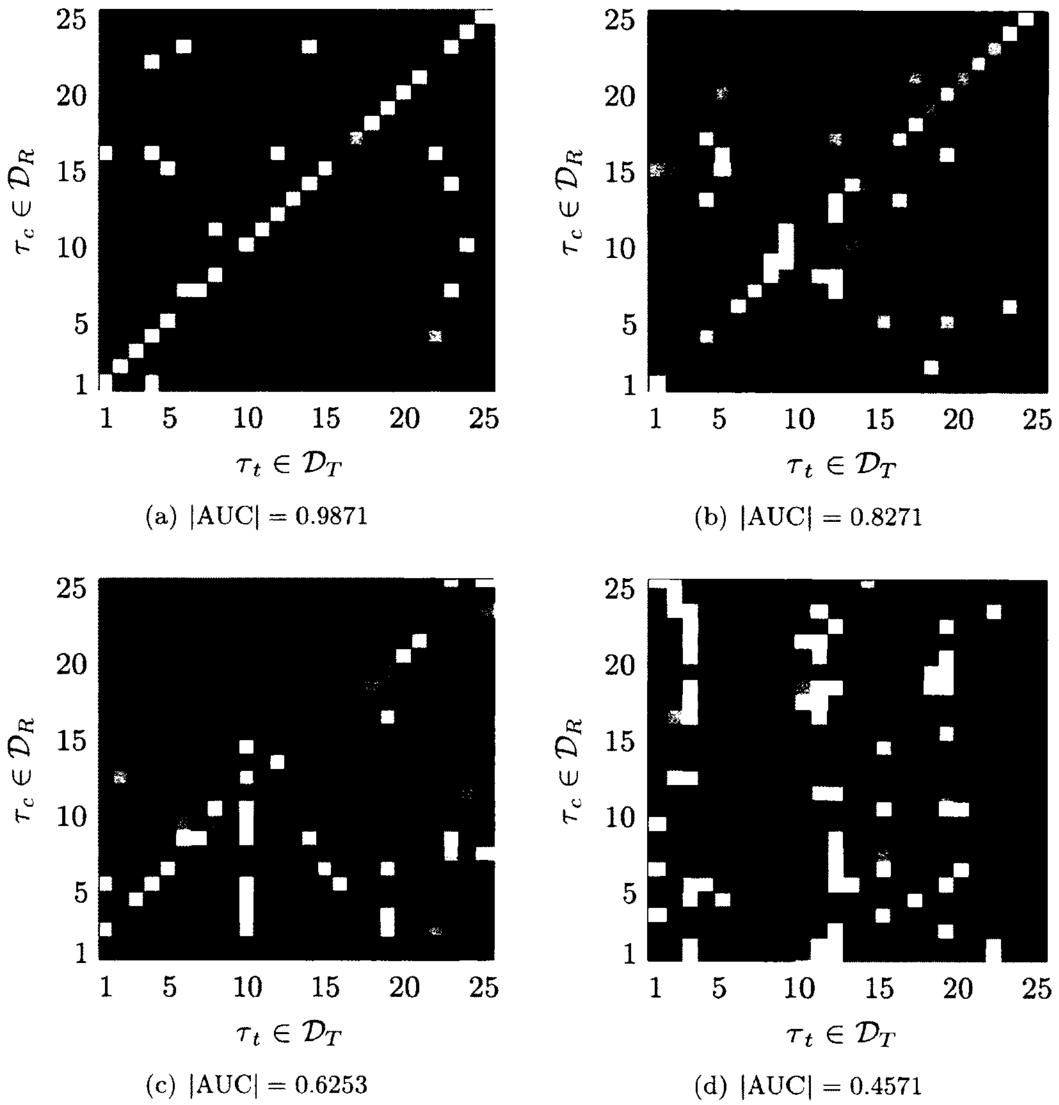
From these values, we can calculate the sensitivity (χ) and specificity (ψ),

$$\begin{aligned} \chi(h) &= \frac{t_+(h)}{t_+(h)+f_-(h)} \\ \psi(h) &= \frac{t_-(h)}{t_-(h)+f_+(h)} \end{aligned} \tag{4.16}$$

The concept of sensitivity and specificity values is inherent in binary classification, where testing data is identified as either a positive or negative match for each possible class. High values of sensitivity indicate that the classifier successfully classified most of the testing tags whenever the testing tag and classifier tag were the same, while high values of specificity indicate that the classifier successfully classified most of the testing tags being different than the classifier tag whenever the testing tag and the classifier tag were not the same. Since sensitivity and specificity are functions of the threshold h, they can be plotted with sensitivity ($\chi(h)$) on the y-axis and $1-$specificity ($1 - \psi(h)$) on the x-axis for $0 < h \leq 1$ in what is known as a receiver operatic characteristic (ROC) [128]. The resulting curve on the ROC plane is essentially a summary of the sensitivity and specificity of a binary classifier as the threshold for discrimination changes. Points on the diagonal line $y = x$ represent a result as good as random guessing, where classifiers performing better than chance have curves above the diagonal in the upper left-hand corner of the plane. The point $(0, 1)$ corresponding to $\chi = 1$ and $\psi = 1$ represents perfect classification.

The area under each classifier's ROC curve ($|\mathrm{AUC}|$) is a common measure of a classifier's performance, and is calculated in practice using simple trapezoidal integration. Higher $|\mathrm{AUC}|$ values generally correspond to classifiers with better performance [129]. This is not a strict rule, however, as a classifier with a higher $|\mathrm{AUC}|$ may perform worse in specific areas of the ROC plane than another classifier with a lower $|\mathrm{AUC}|$ [130]. Several example confusion matrices can be seen in Figure 4.11, where each corresponding $|\mathrm{AUC}|$ value is provided to highlight their relationship to performance. It can be seen that the confusion matrix with the highest $|\mathrm{AUC}|$ has a clear, distinct diagonal of positive classifications while the lowest $|\mathrm{AUC}|$ has positive classifications scattered throughout the matrix.

The use of $|\mathrm{AUC}|$ values for directly comparing classifier performance has recently been questioned [131, 132], identifying the information loss associated with summarizing the ROC curve distribution as a main concern. We therefore do not use $|\mathrm{AUC}|$ as a final classifier ranking measure. Rather, they are only used here to narrow the results down from all the possible classifier configurations to a smaller subset of the



(a) |AUC| = 0.9871

(b) |AUC| = 0.8271

(c) |AUC| = 0.6253

(d) |AUC| = 0.4571

Figure 4.11: A comparison of confusion matrices $\mathcal{L}$ for classifiers of varying performance, with 0 → black and 1 → white. A perfect confusion matrix has values of $\mathcal{L} = 1$ whenever $\tau_c = \tau_t$, seen as a white diagonal here, and values of $\mathcal{L} = 0$ whenever $\tau_c \neq \tau_t$, seen as a black off-diagonal here. In general, |AUC| = 1 corresponds to a perfect classifier, while |AUC| = 0.5 performs as well as random guessing. This trend can be seen in the matrices.

'best' ones. The values of $\chi(h)$ and $\psi(h)$ are still useful measures of the remaining top classifiers. At this point, however, they have been calculated for a range of threshold values that extend over $0 < h \leq 1$. A variety of methods can be used to determine a final decision threshold h for a given classifier configuration, the choice of which depends heavily on the classifier's final application. A popular approach involves sorting by the minimum number of misclassifications, $\min(f_+ + f_-)$, however this does not account for differences in severity between the differnt types of misclassifications [133]. Instead, the overall classifier results were sorted here using their position in the ROC space corresponding to the Euclidean distance from the point $(0, 1)$ as a metric. Formally, this is

$$d_{\mathrm{ROC}}(h) = \sqrt{(\chi - 1)^2 + (1 - \psi)^2}. \tag{4.17}$$

For each classifier configuration, the threshold value $\hat{h}$ corresponding to the minimum distance was determined,

$$\hat{h} = \operatorname*{argmin}_h d_{\mathrm{ROC}}(h) = \{h|\forall h' : d_{\mathrm{ROC}}(h') \geq d_{\mathrm{ROC}}(h)\}. \tag{4.18}$$

In other words, $\hat{h}$ is the threshold value corresponding to the point in the ROC space that is closest to the $(0, 1)$ 'perfect classifier' result. The classifier configurations are then ranked by the lowest distance $d_{\mathrm{ROC}}(\hat{h})$. Figure 4.12 shows an example of the ROC curves for the classifiers that are generated from the confusion matrices found in Figure 4.11. In it, the point corresponding to $\hat{h}$ is indicated by a circle, with the $(0, 1)$ point indicated by a star.

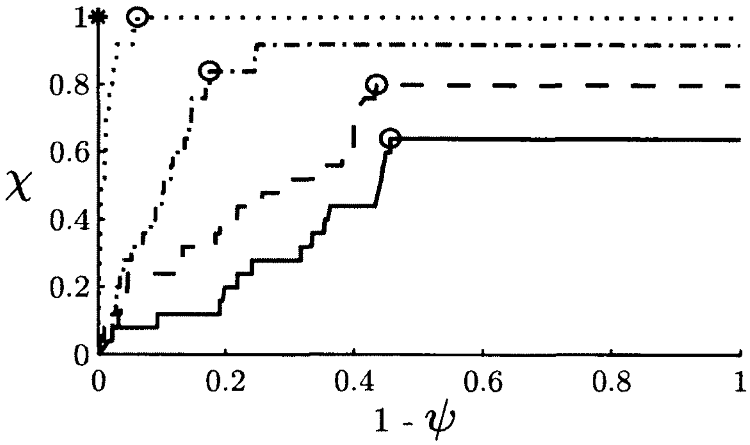


Figure 4.12: ROC curves for the classifiers corresponding to the confusion matrices in Figure 4.11, with |AUC| = 0.9871 (dotted line), |AUC| = 0.8271 (dash-dot line), |AUC| = 0.6253 (dashed line), and |AUC| = 0.4571 (solid line). The 'perfect classifier' result at $(0,1)$ in this ROC space is represented by the black star $(*)$, and each curve's closest point to this optimal result at threshold $\hat{h}$ is indicated by a circle $(\circ)$.

4.8 Results

Frequency Comparison

The ultra-high-frequency (UHF) range of RFID frequencies spans from 868-928 MHz, however in North America UHF can be used unlicensed from 902-928 MHz (± 13 MHz from a 915 MHz center frequency). We test the potential for pattern classification routines to uniquely identify RFID tags at several of operating frequencies within this range. Data collected at three frequencies (902, 915, and 928 MHz) while being held at a single orientation (PL) were used as training and testing frequencies for the classifier. Only amplitude $(\alpha(t))$ signal compression was used in this frequency comparison.

Table 4.3 shows the top individual classifier configuration for the RFID reader operating frequency comparison. Results are presented as sensitivity and specificity values for the threshold value $\hat{h}$ that corresponds to the minimum distance d_{ROC} in the ROC space. Similarly, confusion matrices are presented in Figure 4.13 for each classifier configuration listed in Table 4.3.

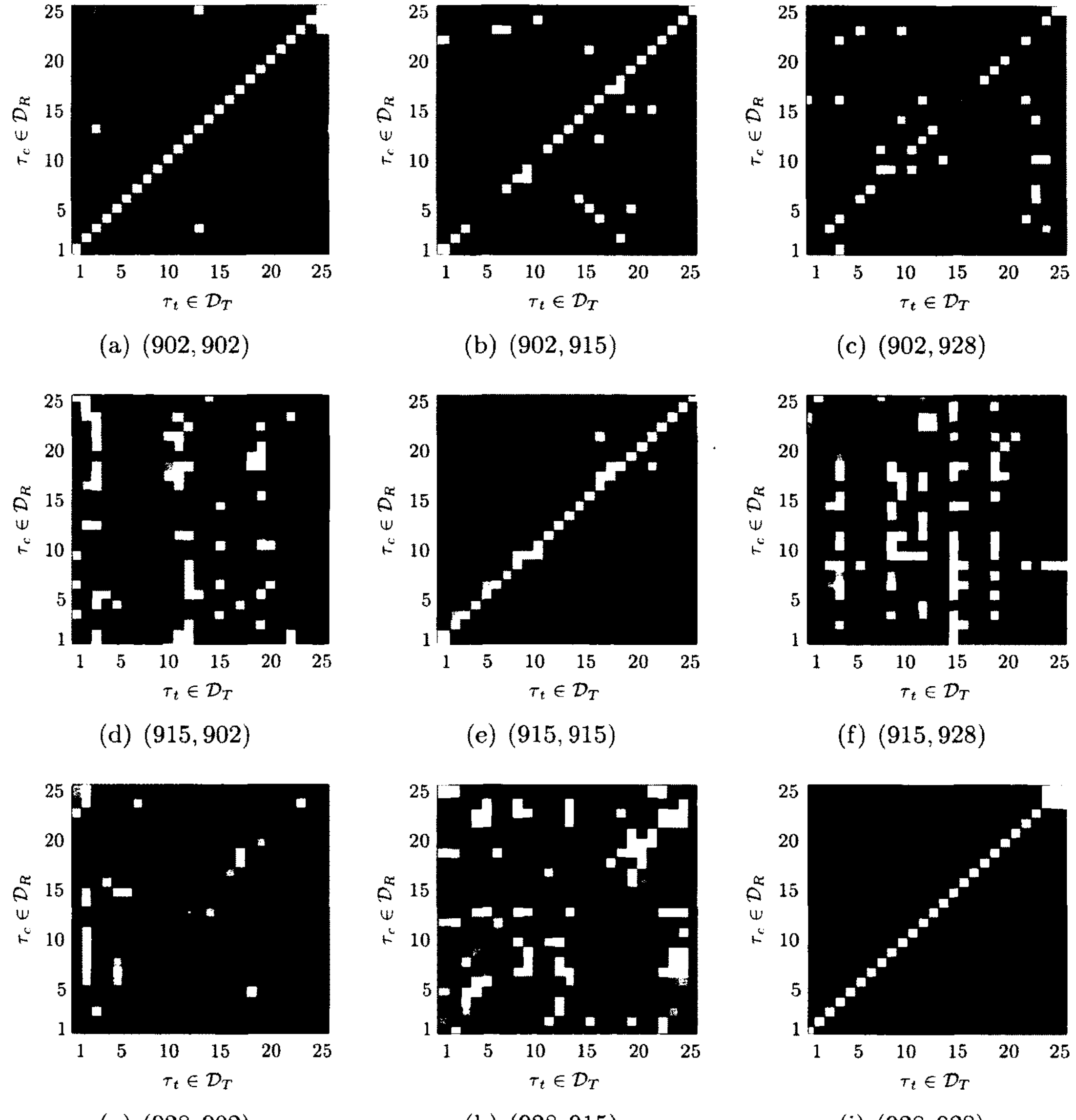
The classifier performed well when trained on the data set collected at 902 MHz, regardless of what frequency the testing data was collected. Accuracies were above 94.6% for all three testing frequencies, and sensitivity ($\chi(\hat{h})$) and specificity ($\psi(\hat{h})$) values were all above 0.943, very close to the ideal value of 1.000. These confusion matrices shown in Figure 4.13(a) - (c) all display the distinct diagonal line which indicates accurate classification. When the classifier was trained on either the 915 MHz or 928 MHz data sets, however, the classification accuracy was low. Neither case was able to identify tags from other frequencies very well, even though they did well classifying tags from their own frequency. When D_R = 915 MHz and D_T = 928 MHz, for example, the |AUC| value was only 0.5195, not much higher than the 0.5000 value associated with random guessing. The corresponding confusion matrix (Figure 4.13(f)) shown no diagonal but instead vertical lines at several predicted tag labels, indicating that the classifier simply labeled all of the tags as one of these values.

Orientation Comparison

A second variable that is inherent in real-world RFID application is the orientation relative to the antenna at which the RFID tags are read. This is one of the main reasons why RFID technology is considered advantageous compared to traditional barcodes, however antenna design and transmission power variability results in changes in the size and shape of the transmission field produced by the antenna [134]. It follows that changes in the tag orientation relative to this field will result in changes in the pre-demodulated RF signals. To test how the pattern classification routines will behave with a changing variable like orientation, data was collected at three different orientations (PL, OB, and UD) while being held at a common operating frequency (902 MHz). This data was used as training and testing sets for the classifiers. Again,

Table 4.3: The classifier configurations ranked by $d_{\mathrm{ROC}}(\hat{h})$ over all values of the classifier configuration variables when trained and tested on the frequency parameters 902, 915, and 928 MHz. $\mathcal{D}_R$ and $\mathcal{D}_T$ correspond to the training and testing data sets, respectively. ρ represents the ratio of negative versus positive EPCs in the training set. The threshold $\hat{h}$ corresponding to the minimum distance d_{ROC} is presented, along with the values of $\chi(\hat{h})$ and $\psi(\hat{h})$.

$\mathcal{D}_R$	$\mathcal{D}_T$	Classifier Configuration			|AUC|	Results			Accuracy (%)
		#DWFP Features (ν)	Classifier	ρ		$\chi(\hat{h})$	$\psi(\hat{h})$	$\hat{h}$ (%)	
902	902	100	QDC (MATLAB)	3	0.9983	1.000	0.997	96.6	99.7
902	915	1	LDC (PRTools)	5	0.9898	1.000	0.943	8.5	94.6
902	928	50	3NN	1	0.9334	0.960	0.950	10.9	95.0
915	902	1	LDC (PRTools)	12	0.4571	0.640	0.543	2.3	54.7
915	915	1	QDC (MATLAB)	9	0.9977	1.000	0.995	82.2	99.5
915	928	10	LDC (MATLAB)	all	0.5195	0.720	0.538	1.9	54.6
928	902	10	1NN	3	0.4737	0.520	0.757	9.1	74.7
928	915	1	LDC (MATLAB)	7	0.6587	0.880	0.498	2.0	51.4
928	928	75	QDC (MATLAB)	2	1.0000	1.000	1.000	86.9	100.0



(a) (902, 902)

(b) (902, 915)

(c) (902, 928)

(d) (915, 902)

(e) (915, 915)

(f) (915, 928)

(g) (928, 902)

(h) (928, 915)

(i) (928, 928)

Figure 4.13: Confusion matrices ($\mathcal{L}$) for the classifier configurations corresponding to the minimum distance $d_{\text{ROC}}(\hat{h})$ over all combinations of ($\mathcal{D}_R$, $\mathcal{D}_T$) where $\mathcal{D}_R, \mathcal{D}_T \in$ 902, 915, 928 MHz (a) - (i). Values of $\mathcal{L}$ range from [0, 1] with 0 → black and 1 → white here.

only amplitude ($\alpha(t)$) signal compression was used.

Table 4.4 shows the top individual classifier configuration for the RFID tag orientation comparison. Results are presented as sensitivity and specificity values for a threshold value $\hat{h}$ that corresponds to the minimum distance d_{ROC} in the ROC space. Similarly, confusion matrices are presented in Figure 4.14 for each classifier configuration listed in Table 4.4.

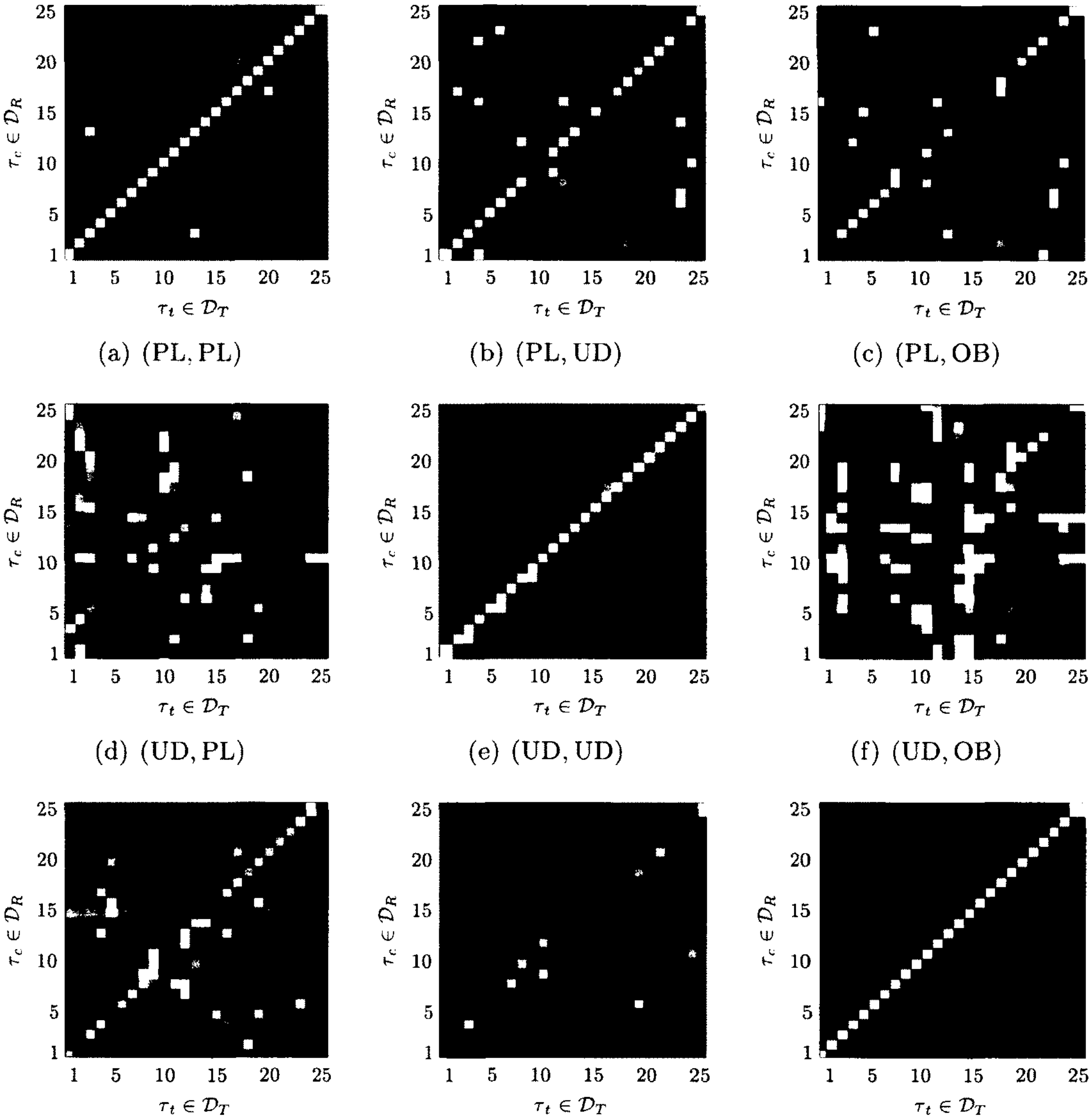
Similar to the frequency results, the classification results again show a single orientation that performs well as a training set regardless of the subsequent tag orientation of the testing set. When trained on data collected at the parallel (PL) orientation, the classification accuracies range from 94.9% to 99.7% across the three testing tag orientations. Values of $\chi(\hat{h})$ range from 0.880 - 1.000, meaning that over 88% of the true positives are correctly identified, and $\psi(\hat{h})$ values range from 0.952 - 0.997, indicating that over 95% of the true negatives are accurately identified as well. These accuracies are verified in the confusion matrix representations found in Figure 4.14(a) - (c). When the classifiers are trained on either the oblique (OB) or upside-down (UD) orientations, we again see that the classifiers struggles to identify testing data from alternate tag orientations. The best performing of these results is for D_R = OB and D_T = PL, where $\chi(\hat{h}) = 0.920$ and $\psi(\hat{h}) = 0.770$ suggesting accurate true positive classification with slightly more false positives as well, resulting in an overall accuracy of 77.6%. When D_R = UD, the testing results are again only slightly better than random guessing, with |AUC| values of 0.5398 for D_T = PL and 0.5652 for D_T = OB.

Different Day Comparison

We next present classification results when data recorded on multiple days were used as training and testing data sets. The following analysis provide a better understanding of how signals taken from the same tag, same frequency, same orientation, but in

Table 4.4: The classifier configurations ranked by $d_{\mathrm{ROC}}(\hat{h})$ over all values of the classifier configuration variables when trained and tested on the orientation parameters PL, UD, and OB. $\mathcal{D}_R$ and $\mathcal{D}_T$ correspond to the training and testing data sets, respectively. ρ represents the ratio of negative versus positive EPCs in the training set. The threshold $\hat{h}$ corresponding to the minimum distance d_{ROC} is presented, along with the values of $\chi(\hat{h})$ and $\psi(\hat{h})$.

		Classifier Configuration				Results			
$\mathcal{D}_R$	$\mathcal{D}_T$	#DWFP Features (ν)	Classifier	ρ	$\lvert\mathrm{AUC}\rvert$	χ	ψ	h (%)	Accuracy (%)
PL	PL	1	QDC (MATLAB)	13	0.9979	1.000	0.997	88.9	99.7
PL	UD	75	3NN	1	0.9489	0.960	0.953	12.1	95.4
PL	OB	20	1NN	1	0.8627	0.880	0.952	2.8	94.9
UD	PL	10	LDC (MATLAB)	19	0.5398	0.680	0.658	2.9	65.9
UD	UD	1	QDC (MATLAB)	5	0.9994	1.000	0.995	73.6	99.5
UD	OB	5	LDC (MATLAB)	15	0.5652	0.680	0.622	1.9	62.4
OB	PL	10	LDC (MATLAB)	13	0.8250	0.920	0.770	5.8	77.6
OB	UD	5	1NN	4	0.6042	0.760	0.622	1.9	62.7
OB	OB	75	QDC (MATLAB)	2	1.0000	1.000	1.000	47.7	100.0



(a) (PL, PL)

(b) (PL, UD)

(c) (PL, OB)

(d) (UD, PL)

(e) (UD, UD)

(f) (UD, OB)

(g) (OB, PL)

(h) (OB, UD)

(i) (OB, OB)

Figure 4.14: Confusion matrices ($\mathcal{L}$) for the classifier configurations corresponding to the minimum distance $d_{\text{ROC}}(\hat{h})$ over all combinations of ($\mathcal{D}_R$, $\mathcal{D}_T$) where $\mathcal{D}_R, \mathcal{D}_T \in$ PL, UD, OB (a) - (h). Values of $\mathcal{L}$ range from [0, 1] with 0 → black and 1 → white here.

subsequent recordings on multiple days are comparable to each other. It is important to note that the data used here was collected with the RFID tag being held by hand above the antenna. While it was held as consistently as possible, it was not fixed in position. Additionally, each subsequent recording was done when environmental conditions were intentionally different from the previous recordings (humidity, temperature, etc.). Data was collected on four different days (Day 1, 2, 3, and 4). This data was used as training and testing sets for the classifiers. Amplitude ($\alpha(t)$) as well as EPC error ($e_{EPC}(t)$) were both used as signal compression methods.

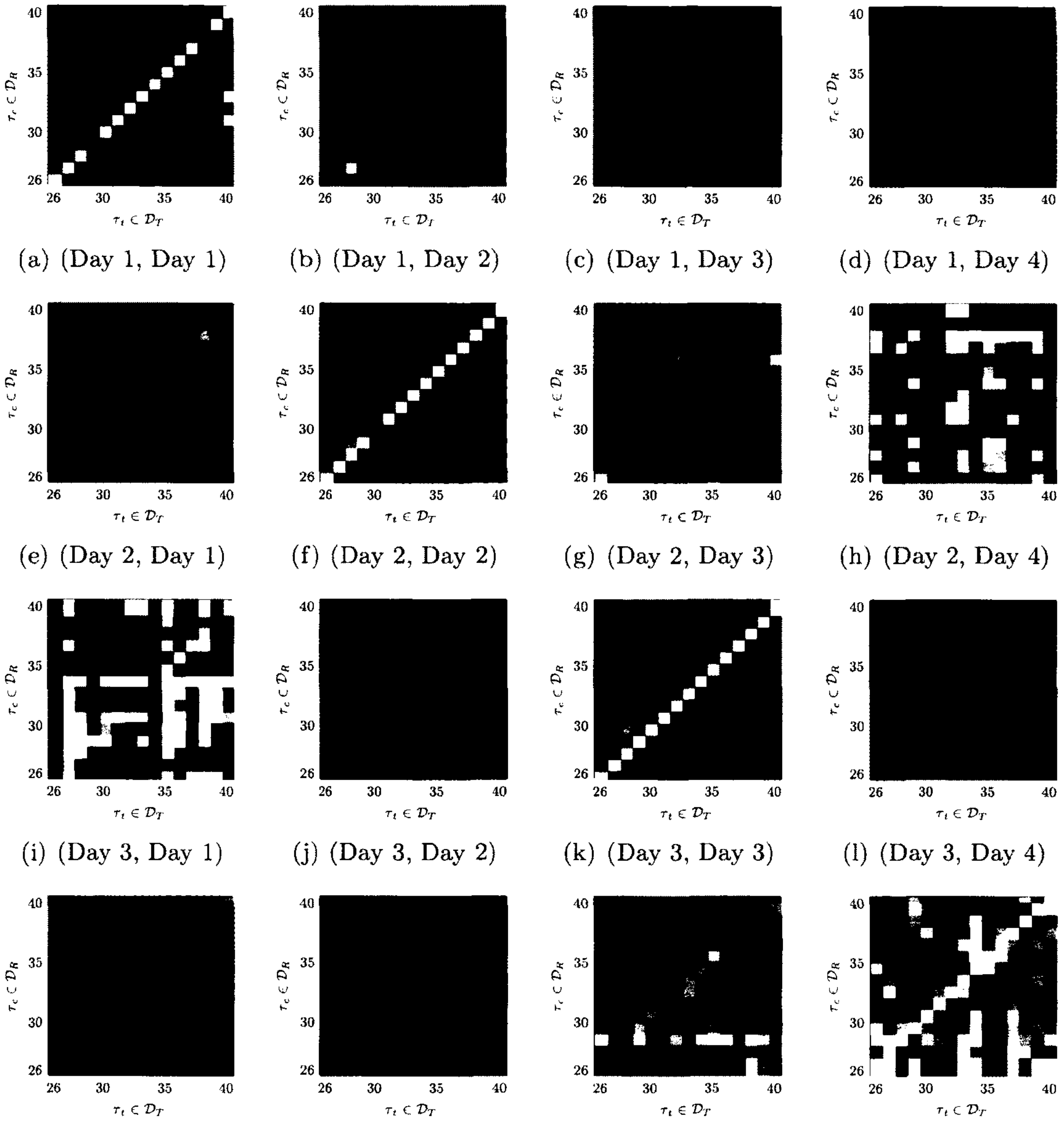
Table 4.5 shows the top individual classifier configuration for the different day tag recording comparison. Results are presented as sensitivity and specificity values for a threshold value $\hat{h}$ that corresponds to the minimum distance d_{ROC} in the ROC space. Similarly, confusion matrices are presented in Figure 4.15 for each classifier configuration listed in Table 4.5.

The first thing to note in these results is the prevalence of the EPC error ($e_{EPC}(t)$) signal compression compared to the amplitude ($\alpha(t)$) signal compression. This suggests that $e_{EPC}(t)$ is more able to correctly classify the RFID tags than the raw signal amplitude is. Unlike the two previous sets of results, where one frequency and one orientation classified well compared to the others, there is no dominant subset here. All the different days classified similarly when tested against each other. This is expected, since there should be no reason data trained on a specific day should perform better than any other. |AUC| values were mainly above 0.6700 yet below 0.7500, with accuracies ranging from 63.6% to 80.9% when $\mathcal{D}_R \neq \mathcal{D}_T$.

The confusion matrix representations of these classification results (Figure 4.15) again indicate there is no single dominant training subset. We see that $\mathcal{D}_R = \mathcal{D}_T$ results all show distinct diagaonl lines, even with $\mathcal{D}_R, \mathcal{D}_T =$ Day 4 where there are additional high off-diagonal entries in the matrix. This is indicated in Table 4.5 by

Table 4.5: The classifier configurations ranked by $d_{\mathrm{ROC}}(\hat{h})$ over all values of the classifier configuration variables when trained and tested on the different day parameters Day 1, 2, 3, and 4. $\mathcal{D}_R$ and $\mathcal{D}_T$ correspond to the training and testing data sets, respectively. ρ represents the ratio of negative versus positive EPCs in the training set. The threshold $\hat{h}$ corresponding to the minimum distance d_{ROC} is presented, along with the values of $\chi(\hat{h})$ and $\psi(\hat{h})$.

		Classifier Configuration					Results			
$\mathcal{D}_R$	$\mathcal{D}_T$	EPC Comp.	#DWFP Features (ν)	Classifier	ρ	|AUC|	χ	ψ	h (%)	Accuracy (%)
Day 1	Day 1	α, e_{EPC}	15	QDC (MATLAB)	2	0.9949	1.000	0.986	23.9	98.7
Day 1	Day 2	e_{EPC}	20	LDC (MATLAB)	4	0.7432	0.867	0.657	39.5	67.1
Day 1	Day 3	e_{EPC}	10	1NN	12	0.6735	0.800	0.662	9.1	67.1
Day 1	Day 4	e_{EPC}	10	QDC (MATLAB)	5	0.7287	0.667	0.724	37.6	72.0
Day 2	Day 1	e_{EPC}	20	LDC (MATLAB)	10	0.7443	0.800	0.748	42.4	75.1
Day 2	Day 2	α, e_{EPC}	20	QDC (MATLAB)	2	0.9990	1.000	0.986	42.4	98.7
Day 2	Day 3	e_{EPC}	20	3NN	1	0.7990	0.800	0.790	52.6	79.1
Day 2	Day 4	e_{EPC}	1	SVM	1	0.7083	0.800	0.733	20.1	73.8
Day 3	Day 1	e_{EPC}	1	SVM	1	0.7014	0.867	0.619	21.9	63.6
Day 3	Day 2	e_{EPC}	15	3NN	8	0.6919	0.800	0.719	4.6	72.4
Day 3	Day 3	α, e_{EPC}	50	QDC (MATLAB)	5	1.0000	1.000	1.000	72.8	100.0
Day 3	Day 4	e_{EPC}	50	3NN	7	0.6390	0.800	0.648	4.6	65.8
Day 4	Day 1	α, e_{EPC}	1	3NN	3	0.7705	0.800	0.710	17.9	71.6
Day 4	Day 2	e_{EPC}	5	1NN	3	0.7395	0.733	0.719	29.8	72.0
Day 4	Day 3	α	1	3NN	1	0.7422	0.667	0.819	57.6	80.9
Day 4	Day 4	α, e_{EPC}	50	LDC (PRTools)	1	1.0000	1.000	1.000	95.7	100.0



(a) (Day 1, Day 1) (b) (Day 1, Day 2) (c) (Day 1, Day 3) (d) (Day 1, Day 4)

(e) (Day 2, Day 1) (f) (Day 2, Day 2) (g) (Day 2, Day 3) (h) (Day 2, Day 4)

(i) (Day 3, Day 1) (j) (Day 3, Day 2) (k) (Day 3, Day 3) (l) (Day 3, Day 4)

(m) (Day 4, Day 1) (n) (Day 4, Day 2) (o) (Day 4, Day 3) (p) (Day 4, Day 4)

Figure 4.15: Confusion matrices ($\mathcal{L}$) for the classifier configurations corresponding to the minimum distance $d_{\text{ROC}}(\hat{h})$ over all combinations of ($\mathcal{D}_R$, $\mathcal{D}_T$) where $\mathcal{D}_R, \mathcal{D}_T \in$ Day 1, 2, 3, and 4 (a) - (p). Values of $\mathcal{L}$ range from [0, 1] with 0 → black and 1 → white here.

a relatively high threshold ($\hat{h}$) value of 95.7. When $\mathcal{D}_R \neq \mathcal{D}_T$, there are still faint diagonal lines present in some of the confusion matrices. For example, when $\mathcal{D}_R =$ Day 2 and $\mathcal{D}_T =$ Day 3 (Figure 4.15(g)), diagonal entries coming out of the lower left-hand corner are somewhat higher in accuracy (closer to white in the confusion matrix) than their surrounding off-diagonal entries. We see in Table 4.5 that this classifier has an |AUC| equal to 0.7990 and a 79.1% overall accuracy.

Damage Comparison

We next present the results of the RFID tag damage analysis to explore how physical degradation affects the RFID signals and the resulting classification accuracy. The data sets from Day 1, 2, 3, and 4 are combined and used here as a single training set. The tags which make up this data set, $AD26 - AD40$, are split into two subsets: tags $AD26 - AD32$ were subjected to water damage study, while tags $AD33 - AD40$ were subjected to a physical damage study. The AD-612 tags are not waterproof nor are they embedded in a rigid shell of any kind, although many RFID tags exist that are sealed to the elements and/or encased in a shell for protection. For the water damage, each tag was submerged in water for three hours, at which point they were patted dry to remove any excess water and used to collect data (labeled as Wet). They were then allowed to air-dry overnight, and were again used to collect data (Wet-to-dry). For the physical damage, each tag was first gently crumpled by hand (Light damage) and subsequently balled up and then somewhat flattened (Heavy damage), with data being collected after each stage.

Table 4.6 shows the top individual classifier configuration for the two RFID tag damage comparisons. Results are presented as sensitivity and specificity values for a threshold value $\hat{h}$ that corresponds to the minimum distance d_{ROC} in the ROC space. Similarly, confusion matrices are presented in Figure 4.16 for each classifier configuration listed in Table 4.6.

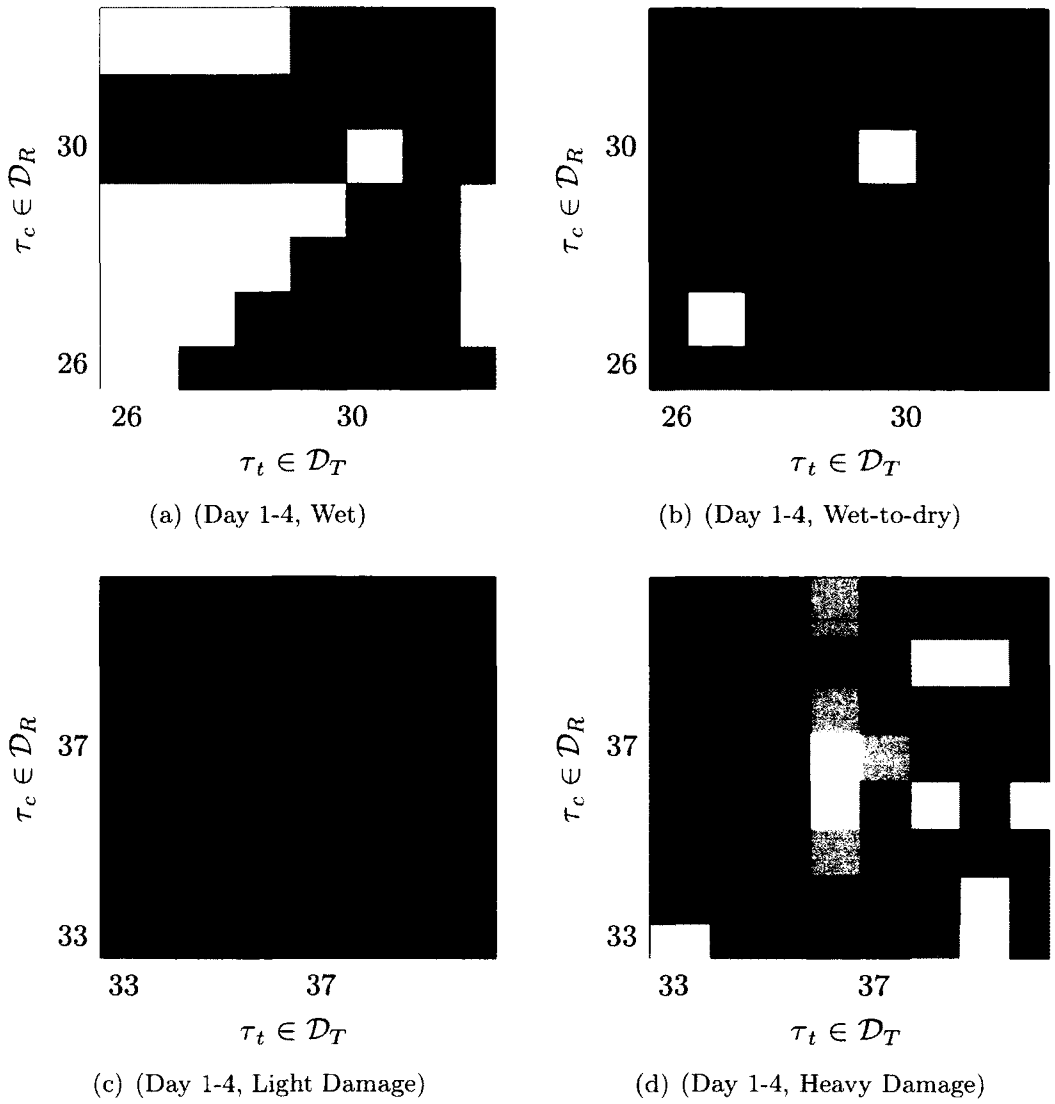
The RFID tag damage classification results provide similar to the previous different day comparison. The water damage did not seem to have a severe effect on the classification accuracy, while the more severe physical damage showed lower classifier accuracy. However, rather than relatively equal $\chi(\hat{h})$ and $\psi(\hat{h})$ values, the heavy damage resulted in values of $\chi(\hat{h}) = 1.000$ and $\psi(\hat{h}) = 0.589$, which means that the classifier was optimistically biased and over-classified positive matches. This lower accuracy was not unexpected, as deformation of the tag's antenna should distort the RF signal and therefore the classifier's ability to identify a positive match for the tag.

4.9 Discussion

The results presented above suggest that a dominant reader frequency, 902 MHz in this case, may exist at which data can be initially collected for the classifier to be trained on and then used to correctly identify tags read at alternate frequencies. In our analysis, we have explored reader frequencies that span the North American UHF range, yet were only part of the full 865-928 MHz UHF range for which the AD-612 tags used here were optimized. Therefore, the dominant 902 MHz read frequency we observed lies at the center of the actual tags operating frequency range. It is of no surprise that the tags perform best at the center of their optimized frequency range rather than at the upper limit. Similarly, a classifier can be trained on a tag orientation (relative to the reader antenna) that may result in accurate classification of RFID tags regardless of their subsequent orientation to the reader antenna. Antenna design for both the readers and the tags is an active field of research [93], and it is expected that the RF field will be non-uniform around the antennas. This could explain why only one of the experimental orientations used here performs better than

Table 4.6: The classifier configurations ranked by $d_{\mathrm{ROC}}(\hat{h})$ over all values of the classifier configuration variables when trained and tested on the tag damage comparisons for both water and physical damage. $\mathcal{D}_R$ and $\mathcal{D}_T$ correspond to the training and testing data sets, respectively. ρ represents the ratio of negative versus positive EPCs in the training set. The threshold $\hat{h}$ corresponding to the minimum distance d_{ROC} is presented, along with the values of $\chi(\hat{h})$ and $\psi(\hat{h})$.

		Classifier Configuration					Results			
$\mathcal{D}_R$	$\mathcal{D}_T$	EPC Comp.	#DWFP Features (ν)	Classifier	ρ	|AUC|	χ	ψ	h (%)	Accuracy (%)
Day 1, 2, 3, 4	Wet	α	1	SVM	1	0.6361	0.714	0.786	80.1	73.8
Day 1, 2, 3, 4	Wet-to-dry	α	1	3NN	17	0.7789	0.857	0.738	4.8	75.5
Day 1, 2, 3, 4	Light Damage	α	5	1NN	16	0.7589	0.750	0.839	17.9	82.8
Day 1, 2, 3, 4	Heavy Damage	α	20	LDC (PRTools)	7	0.7980	1.000	0.589	44.5	64.1



(a) (Day 1-4, Wet)

(b) (Day 1-4, Wet-to-dry)

(c) (Day 1-4, Light Damage)

(d) (Day 1-4, Heavy Damage)

Figure 4.16: Confusion matrices ($\mathcal{L}$) for the classifier configurations corresponding to the minimum distance $d_{\mathrm{ROC}}(\hat{h})$ over all combinations of ($\mathcal{D}_R$, $\mathcal{D}_T$) where $\mathcal{D}_R$ = Day 1, 2, 3, and 4, and $\mathcal{D}_T \in$ wet, wet-to-dry, light damage, and heavy damage (a) - (d). Values of $\mathcal{L}$ range from [0, 1] with 0 → black and 1 → white here.

the others. Regardless of the field strength, however, the unique variations in the RF signature of an RFID tag should be present. It is promising that the classifier still had an accuracy of over 60% with these variations, and up to 94.9% accuracy if trained on the parallel (PL) orientation.

Changes in the environmental conditions, like ambient temperature and relative humidity, were also allowed in the different day study where the RFID tags were suspended by hand near the antenna (in generally the same spot) for data collection on successive afternoons. It is important to note that the tags were not in a fixed location for this study, and that slight variations in both distance to the reader as well as orientation were inherent due to the human element. Even so, the classifier was generally able to correctly identify the majority of the RFID tags as being either a correct match or a correct mismatch when presented with a data set it had never seen before, with accuracies ranging from 63.6% to 80.9%. This study represents a typical real-world application of RFID tags due to these environmental and human variations.

As previously mentioned, the EPC compression method tended to favor the EPC error signal $e_{EPC}(t)$, although there was not a large difference in classifier performance between the different day comparison that used both $\alpha(t)$ and $e_{EPC}(t)$ compression, and the frequency/orientation comparisons that used only $\alpha(t)$ compression. The parameter ρ had a large spread of values across the classifiers, indicating that the classification results may not be very sensitive to class imbalance within the training set. The number of DWFP features also shows no consistent trend in our results, other than being often larger than 1, indicating that there may be room for feature reduction. With any application of pattern classification, a reduction in the feature space through feature selection can lead to improved classification results [9]. Individual feature ranking is one method that can be used to identify features on a one-by-one basis, however it can overlook the usefulness of combining feature variables. In combination with a nested selection method like sequential backward floating search (SBFS), the relative usefulness of the DWFP features, as well as the remaining features, can be evaluated [16].

The results in Tables 4.3 - 4.5 where $\mathcal{D}_R = \mathcal{D}_T$ are comparable to those observed by Bertoncini et al. [64], with some classifiers having 100% accuracy and the rest near 99%. In these instances, the classifier was trained on subsets of the testing data, so it is expected that the classifier performs better in these cases.

It is also important to note that the final decision threshold $\hat{h}$ used can still vary greatly depending on the classifier's application. It is important to note that adjusting the classifier's final threshold value does not alter the classification results of said classifier. The |AUC| takes into account all possible threshold values, and is therefore fixed for each classifier configuration. The threshold values only determine the distribution of error types, χ vs. ψ, within the results. Aside from the minimum d_{ROC} metric, weights can be applied to determine an alternate threshold if the application calls for a trade-off between false negatives and false positive results. For example, if a user is willing to allow up to five false positives before allowing a false negative, a minimizing function can be used to identify this weighted optimal threshold.

A comparison of different methods to determine a final threshold can be seen in Table 4.7, where the classifier configuration trained on Day 1 data and tested on Day 4 data from Table 4.5 is presented for several alternate threshold values. First, the threshold is shown for the minimum distance $d_{\mathrm{ROC}}(\hat{h})$, as was previously presented in Table 4.5. The threshold is then shown for the minimum number of total misclassifications $(f_+ + f_-)$, followed by minimum number of false positives (f_+), and then by that of the lowest number of false negatives (f_-). Several weighting ratios are then shown, where the cost of returning a false-positive (f_+) is increased compared to the cost of returning a false negative (f_-). For example, a weighting ratio of 1 : 5 [f_- : f_+] means that 5 f_+ cost as much as a single f_-, putting more emphasis on reducing the number of f_- present. It can be seen that the values of $\chi(h)$ and $\psi(h)$ change as the threshold h changes. An overly optimistic classifier is the result of threshold values that are too low, when all classifications are identified as positive matches ($\chi \approx 1$ and $\psi \approx 0$). Alternatively, an overly pessimistic classifier is the result of threshold values that are too high, resulting in all classifications identified as negative matches ($\chi \approx 0$ and $\psi \approx 1$). The weighting ratio 1 : 10 [f_- : f_+] returns the most even values of χ and ψ, which matches the $\hat{h}$ threshold. Figure 4.17 shows an example of the trade-off between the values of $\chi(h)$ and $\psi(h)$ as the threshold h is increased, where two examples from Table 4.7 are highlighted.

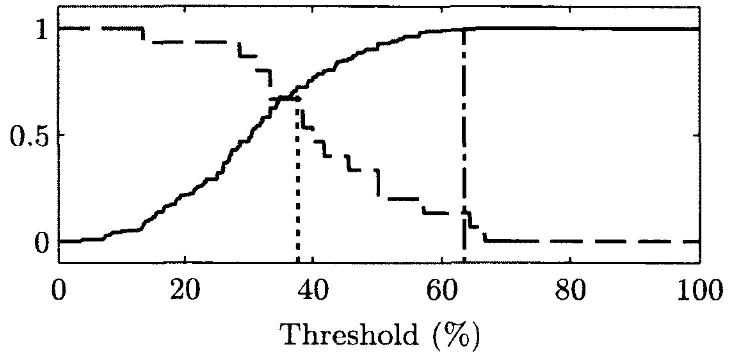


Figure 4.17: A plot of sensitivity $\chi(h)$ (solid line) and specificity $\psi(h)$ (dashed line) versus threshold for the classifier trained on Day 1 and tested on Day 4 from Table 4.5. Threshold values are shown corresponding to the $\min(f_+ + f_-)$ ($h = 63.4$, dash-dot line) as well as $\hat{h}$ ($h = 37.6$, dotted line). The threshold value determines the values of χ and ψ, and is chosen based on the classifier's final application.

It is useful to discuss a few examples here to better understand the different threshold results. In a security application, for example, where RFID badges are used to control entry into a secure area, it is most important to minimize the number of f_+ results because allowing a cloned RFID badge access to a secure area could be devastating. In this situation, we would want the value of ψ to be as close to 1.0

Table 4.7: The classifier configuration trained on Day 1 and tested on Day 4 data from Table 4.5 using several different metrics to determine the final threshold h. Metrics include $\min(f_+ + f_-)$, $\min(f_+)$, $\min(f_-)$, a weight of 1:5, 1:10, 1:15, and 1:20 for $[f_+ : f_-]$, meaning that up to 20 f_+ are allowed for each f_-. The choice of metric used to determine the final threshold value depends on the classifier's final application.

		Classifier Configuration						Results			
$\mathcal{D}_R$	$\mathcal{D}_T$	EPC Comp.	#DWFP Features	Classifier	ρ	|AUC|	Sorted by	χ	ψ	h (%)	Accuracy (%)
Day 1	Day 4	e_{EPC}	10	QDC (MATLAB)	5	0.7287	$\min d_{\mathrm{ROC}}(\hat{h})$	0.667	0.724	37.6	72.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	$\min(f_+ + f_-)$	0.133	0.995	63.4	93.8
							$\min(f_+)$	0.067	1.000	65.3	93.8
							$\min(f_-)$	1.000	0.000	0.1	6.7
							1:5 $[f_+ : f_-]$	0.200	0.981	56.7	92.9
							1:10 $[f_+ : f_-]$	0.667	0.724	37.6	72.0
							1:15 $[f_+ : f_-]$	0.867	0.529	30.8	55.1
							1:20 $[f_+ : f_-]$	0.933	0.443	28.2	47.6

as possible. In Table 4.7, this result corresponds to a threshold value of $h = 65.3$. Unfortunately, the value of χ at this threshold is 0.067, which means that almost all of the true ID badges would also be identified as negative matches. Therefore, that specific classifier is not appropriate for a security application.

An alternate example is the use of RFID-embedded credit cards in retail point of sale. To a store, keeping the business of a repeat customer may be much more valuable than losing some merchandise to a cloned RFID credit card. In this sense, it is useful to determine an appropriate weight of $[f_- : f_+]$ that evens the gains and losses of both cases. If it were determined that a repeat customer would bring in 20 times as much revenue as it would cost to refund a fraudulent charge due to a cloned RFID account, then a weight of $1 : 20$ $[f_- : f_+]$ could be used to determine the optimal classifier threshold. From Table 4.7, it can be seen that a corresponding threshold is $h = 28.2$, resulting in values of $\chi = 0.933$ and $\psi = 0.443$. This classifier would incorrectly identify 7% of the repeat customers as being fraudulent while correctly identifying 44% of the cloned signals as being fraudulent. This specific classifier could be useful in this retail example.

4.10 Conclusion

The USRP software defined radio system has been shown to capture signals at a usable level of detail for RFID tag classification applications. Since the signal manipulations are performed in software, this allows us to extract not only the raw RF signal, but it also allows us to generate our own, ideal signal to compare against. A new signal representation has been created this way that is the difference between the recorded and ideal signal representations, $e_{EPC}(t)$, and has proven to be very useful in the classification routines.

The binary classification routine has been explored on more real-world grounds, including exposure to a variety of environmental conditions without the use of RF shielding to boost the SNR level. To explore classifier robustness without fixed proximity and orientation relative to the RFID reader, several validation classifications were performed, including an RFID reader frequency comparison, a tag orientation comparison, a multi-day data collection comparison, as well as physical damage and water exposure comparisons. The frequency comparison was performed to determine the effect that the variability of potential RFID readers inspection frequencies would have on the classification routines. The results were promising, although not perfect, and suggest that while it is best to train a classifier on all possible scenarios, a main frequency (i.e. center frequency) could potentially be used for a master classifier training set. A similar orientation comparison was done, altering the RFID tag's orientation relative to the antenna. Again, the results showed it was best to train the classifiers on the complete set of data, however there was again promise for a potential single main orientation that could be used to train a classifier.

In the multi-day collection comparison, data was collected by hand in an identical fashion but on separate days. The results showed that the inconsistency associated with holding an RFID tag near the antenna cause the classifiers to have trouble correctly identifying EPCs as coming from their correct tag. Two further comparisons were performed to assess the degree that physical degradation had on the RFID tags. When subjected to water, promising classifier configurations were found that were on the same level of accuracy as results seen for undamaged tags, suggesting that the water may not have a significant effect on the RFID classification routines. A separate subset of the RFID tags were subjected to a similar degradation analysis, this time with physical bending as a result of being crumpled by hand. The results show that, as expected, bending of the RFID tag's antenna caused degradation in the raw signal that caused the classifier to misclassify many tags.

Applications of RFID technology that implement a fixed tag position are a potential market for the classification routine we present. One example of this is with ePassports, which are embedded with an RFID chip containing digitally signed biometric information [98]. These passports are placed into a reader that controls the position and distance of the RFID chip relative to the antennas. Additionally, passports are generally protected from the elements and can be replaced if they undergo physical wear and tear. We have demonstrated a specific emitter identification technique that performs well given these restrictions.

We have applied the DWFP framework to generate a potentially innumerable amount of features for input into a pattern classification routine. We then used these features to study the effect of variations in the physical act of recording RFID tag signals. We have shown through reduced classification accuracies that these changes manifest themselves in the data in an unforeseen way. The framework to systematically study these changes has been established here. In the next chapter, we continue building our analysis technique through a new set of time-domain signals for Lamb wave damage characterization.

Chapter 5

Characterization of Flaw Severity using Pattern Recognition for Guided Wave-Based Structural Health Monitoring

We have discussed the application of pattern classification routines for time-domain signal analysis, and have shown that physical changes in the collection of data can result in changes in the data itself. These changes were shown to have a negative effect on the classification accuracy, as the manifestation of these physical changes in the data was not fully understood. To explore this concept in a systematic manner, we turn to the field of structural health monitoring. Corrosion in materials can be thought of as an effective reduction in material thickness resulting in reduced structural integrity. Since corrosion is a function of time, it follows that corrosion provides a representative process for studying a physical change over time. We simulate corrosion to an aircraft-grade aluminum plate by introducing a thinning at a variety of intermediate severities, ranging from unflawed to a through-hole. By measuring guided waves which interact with the flaw at each level of severity, we construct a data set which we then use to explore how this physical process manifests itself within a given feature space.

5.1 Introduction

Flaw detection in metals remains an important area of research as the world's aviation and naval fleets continue to age [135]. It is critical that flaws be discovered before structural failure, and accurate maintenance planning requires knowledge of a structure's state of health. Lamb waves [73,81] have proven to be a popular technique in structural health monitoring (SHM) [82, 83] due to their multi-mode propagation and dispersive properties. Of interest in SHM is that Lamb waves are confined by a structure's boundaries and so follow its shape and curvature, giving sensitivity to material discontinuities at either surface as well as in the interior of the plate, pipe or shell. Lamb waves have the ability to propagate relatively long distances [136] and beneath layers of insulation or other coverings [43]. Together these properties allow Lamb waves to rapidly cover large areas of interest, providing a useful tool for identifying defects in a variety of structures. However, their multi-mode nature is often too complex for direct interpretation. A typical multi-mode waveform is shown in 5.1, with predicted mode arrival times highlighted.

The majority of the Lamb wave literature side-steps the complexities associated with multi-mode Lamb wave signals. Some inspection techniques include specific assumptions or restrictions that can render them impractical for field use. Many assume a uniform thickness of the plate [84] or rely on comparing signals from damaged and undamaged areas in an instantaneous cross-correlation analysis [87]. A

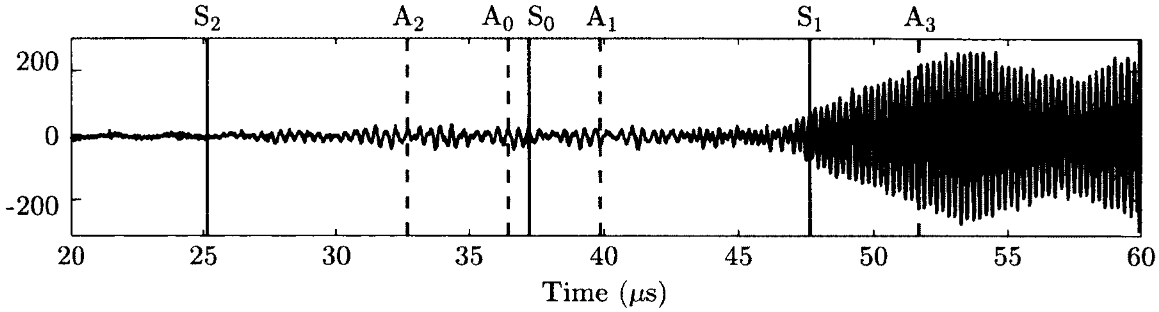


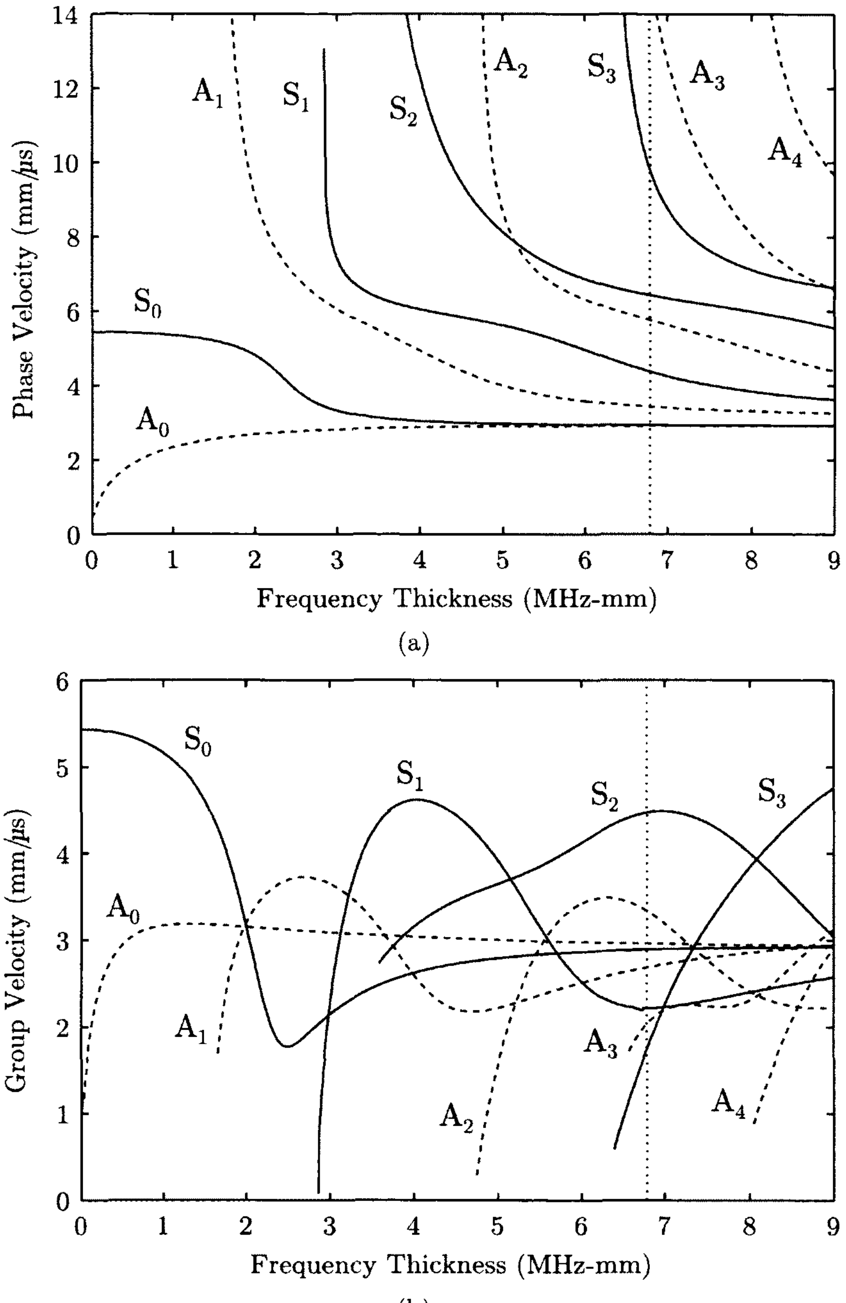
Figure 5.1: A raw waveform from the unflawed plate is shown here with predicted mode arrival times, indicated for both symmetric (solid vertical lines) and antisymmetric modes (dashed vertical lines). The portion of the signal that is of interest often lies buried in noise (from 25-45 μs here), while the higher-amplitude portion of the signal (45-60 μs) contains an overlap of both slower, low velocity modes as well as faster edge-reflected modes due to the finite size of the plate sample. For the deeper flaw, mode conversion and scattering effects prevent an accurate prediction of arrival times in addition to reducing the signal-to-noise ratio of the waveform.

common implementation of Lamb waves for damage detection involves restricting the frequency-thickness product to a regime where only the fundamental modes exist [83]. Doing so promises easier analysis of mode arrival-time shifts, since mode overlap is less likely when the lower-order modes are spread far apart in group velocity. Larger frequency-thickness values gives rise to higher-order modes, but result in signals that often prove quite difficult to analyze. Nevertheless, each mode propagates with a unique through-thickness displacement and stress profile [72]. It can therefore be beneficial to use multi-mode Lamb wave signals since each mode has different dispersion characteristics and sensitivity to defects.

Tomographic reconstructions have been shown to produce quantitative maps of damage in regions of interest in both single-mode and multi-mode Lamb wave signals [45, 46, 61, 137 150]. By considering the dispersion curves for a material, shown in Figure 5.2 for aluminum, it can be seen that decreasing frequency-thickness values gives fewer Lamb wave modes, eventually reducing down to two fundamental zeroth-order modes. Changes in material thickness can be directly linked to changes in wave speed for these two modes, making them attractive to use for NDE applications. As frequency-thickness values are increased, higher order modes overlap and mix together, quickly masking the direct association with material thickness. Even though their behavior is not always predictable, these modes are still sensitive to material thickness changes. By extracting Lamb wave mode arrival times directly from raw signals, tomographic reconstructions generate a slowness map that highlights *any* changes in mode speed. These can then be geometrically related to material changes in the plate.

It is important to note here that for shallow flaws (relative to the plate thickness), the main effect of mode interaction is a change in group velocity and hence arrival time for the various modes. For deeper flaws, the mode interaction becomes much more complex with three-dimensional scattering and mode mixing occurring. This behavior is not accounted for in Lamb wave tomography reconstructions. Malyarenko and Hinders [144] present an improvement on straight ray tomography by incorporating ray bending though diffraction tomography, however that is presented for relatively low frequency-thickness values and does not account for the mode mixing and scatter that dominates the behavior for deeper, strongly scattering flaws. This higher-order complexity is explored further by Leckey et al. [151], who present multi-mode Lamb wave scattering through 3D elastodynamic finite integration (EFIT) simulations. They show that for even for the relatively simple geometries, supercomputer simulations are required to explain the 3D behavior of the wave fields.

Because Lamb wave tomography reconstructions are able to size and localize flaws, including strongly scattering flaws, they present a mechanism to automatically identify which waveforms have interacted with any material defects present. Pattern classification is the study of how machines can be taught to observe and learn from an environment in an attempt to recognize and identify patterns in sensor data. Pat-



(a)

(b)

Figure 5.2: Dispersion curves for an aluminum plate. Solutions to the Rayleigh-Lamb wave equations are plotted here for both symmetric (solid lines) and antisymmetric (dashed lines) modes for both phase (a) and group (b) velocity. The value of frequency-thickness used in the data collection for this analysis is indicated by the vertical dotted line, where several modes can be seen to exist.

tern classification techniques [1,3,4,9] can then be used to provide a formulism which exploits the scattering and/or mode mixing information in the waveforms in order to determine the flaw depth.

In most applications of pattern classification, the distribution of classes within a feature space is not important as long as the classes are optimally separated. For a process like incremental damage detection, each class can be thought of as a step in a physical process which changes over time. For example, corrosion is a continuous process where severity of damage can only increase with time. If the damage caused by corrosion is discretized into individual thickness loss steps, each can be given a different class label. In a feature space, these classes should not only be optimally separated but should remain sequentially ordered. That is, each class should ideally border the classes before and after it when sequentially ordered with respect to severity. By introducing this restriction, new data from intermediate severities will fall between their bounding classes in the feature space. This concept is referred to here as the 'sequential' ordering of classes, and will be important in down-selecting the feature space.

5.2 Discussion on Feature Selection

From the Rayleigh-Lamb wave equations (Appendix A), we know that the Lamb wave mode structures and velocities are functions of a frequency-thickness parameter. Since the inspection frequency can be considered a constant, and the thickness change due to corrosion can be considered a continuous function, it follows that any changes to the waveform will also be continuous. Therefore, properties extracted from these waveforms will change continuously as well. This means that a point in the feature space will "move" continuously as a function of material loss over time. Each class, therefore, represents a region of this curve in the feature space, shown conceptually in Figure 5.3.

If the feature space is chosen so that this curve is nearly linear, as is seen in Figure 5.3(a), then it can be discretized neatly into successive classes. When new data is introduced, seen as the red dot between classes 6 and 7 in Figure 5.3(b), the data will lie between these successive classes only and the classification is straightforward. If this curve is bunched up together or even overlapping itself, as seen in Figure 5.3(c), several problems can arise when classifying new data. We present one possible issue in Figure 5.3(d), where the same new data is introduced between classes 6 and 7, however this time it is located in an area where several alternate classes exist. Rather than correctly being classified as between classes 6 and 7 only, it could also be classified as classes 2 and 3 here. To the user, this jump in classification results would appear to be a significant error which could be dangerous in SHM.

It should be clear by now that there are many complications associated with the concept of classification for a set of classes that are essentially a discrete representation of a continuous function. This problem presents itself primarily in the feature selection step of classification, where a subset of "ideal" features is chosen that result in the optimal value of a specific metric, a common choice being the minimal classification error of a given classifier. It follows that a classifier would likely be overtrained using this approach, identifying a feature subset that best classifies the training data alone without taking into account the introduction of new data existing between established classes. For this reason, we explore a variety of novel feature selection techniques based on measures of the various class data within the feature space itself in an attempt to select a more appropriate feature set for general application.

There exists a theorem in pattern classification known as the *Ugly Duckling Theorem* which states that in the absence of assumptions there is no 'best' feature rep-

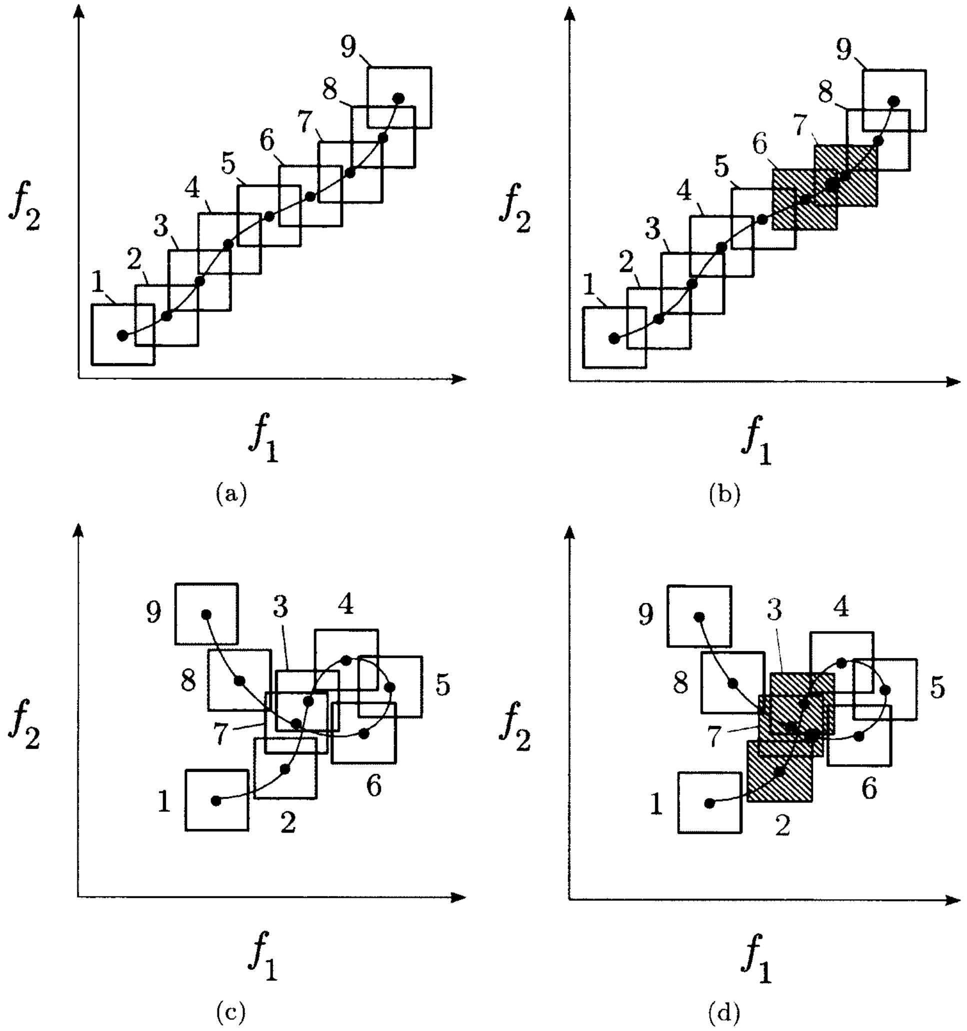


Figure 5.3: An example of possible class distributions in a two-dimensional feature space (f_1 vs. f_2). Here, flaw thickness is a function of time, represented by $g(t)$, where class 1 corresponding to $g(t_1)$, class 2 corresponding to $g(t_2)$, etc. with $t_1 < t_2 < \ldots < t_9$. Two possible class distributions are presented here, the first being a more linear distribution (a) and the other a more complicated distribution (c). For both cases, a new data point is considered that lies between existing classes 6 and 7, seen here as a red dot. For the linear case, this new data lies within the class boundaries of classes 6 and 7 only (b), making the classification of this new data straightforward. In the more complicated distribution, this new data lies not only within class 7, but also within classes 2 and 3 (d), resulting in an apparent error in the classification.

resentation for a data set, since assumptions about what 'best' means are necessary for the choice of features [58]. Appropriate features for a given problem are usually unknown *a priori*, and as a result many features are often generated without any knowledge of their relevancy [11]. This is the motivation behind generating a large number of features in the first place. It is also important to take into account the relative size of the number of objects to be classified and the size of the feature space to be used for classification. Including too many features may degrade the performance of a classifier if the number of samples per class is comparatively small, resulting in a sparse distribution of data within the feature space. This concept is better known as the *curse of dimensionality* [12]. While an exact relationship is difficult to determine, a generally accepted limit is to use at least ten times as many training samples per class (N) as there are dimensions in the feature space (d), seen as $N/d \geq 10$.

It follows that an intelligent reduction in the size of the feature space is needed. As discussed in Chapter 2, there are two general approaches to reducing the size of the feature set: feature extraction and feature selection. Feature extraction reduces the feature set by creating new features through transformations and combinations of the original features. Principal component analysis is a commonly used feature extraction technique. Since we are interested in retaining the original physical interpretation of the feature set, we opt not to use any feature extraction techniques in our analysis.

There are three general techniques within feature selection: wrapper methods, embedded methods, and filter methods [13]. Wrapper methods use formal classification to rank individual feature space subsets, applying an iterative search procedure which trains and tests a classifier using different feature subsets for accuracy comparison. This search continues until a given stopping criterion is met [14]. This approach is computationally intensive, and there is often a trade-off among algorithms between computation speed and the quality of results that are produced [15 17]. Additionally, these methods have a tendency to over-train themselves, where data in the training set is perfectly fitted and results in poor generalization performance [18]. Similar to wrapper methods, embedded methods perform feature selection while constructing the classification algorithm itself. The difference is that the feature search is intelligently guided by the learning process, reducing the computation time but relying heavily on the choice of classification algorithm. Filter methods perform their feature ranking by looking at intrinsic properties of the data, without the input of a formal classification algorithm. Traditionally, these methods are univariate and therefore don't account for multi-feature dependencies.

Our feature selection approach explored here can be most appropriately considered a multivariate filter method, as it considers features in combination with each other and does not directly depend on the classification model used. A main difference here is that we redefine what it means for a feature to be 'relevant'. Typically, relevant features are identified by their contribution toward class separability. While still important, we add a requirement that the features in combination must result in a sequential class distribution.

5.3 Data Collection

The data used in this study was collected from a 305 *mm* × 305 *mm* × 3.15 *mm* sample of aluminum, shown in Figure 5.4(a). The scanner consisted of two 2.25 MHz, 6.4 *mm* diameter longitudinal transducers arranged in a pitch/catch configuration to transmit and receive the signals. Each transducer was fitted with an 11.5 *mm* cylindrical acrylic delay line with glycerin couplant used at both the transducer surface and the plate surface. The signals were generated and received using a Matec[1]

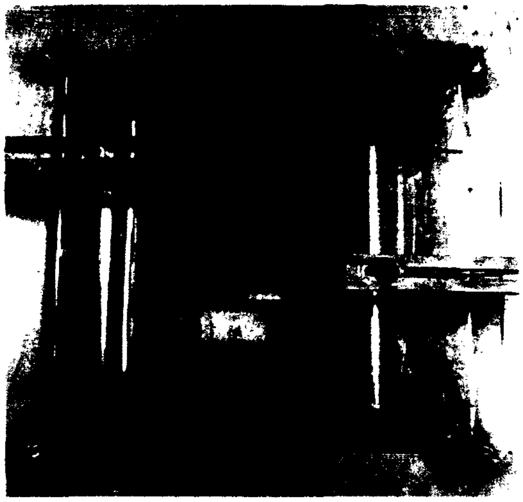
[1] Northborough, MA (`http://www.matec.com/`)

TB1000 pulser/receiver and then digitized with a Gage[2] CS8012a A/D card sampling at 100 MHz. The transducers were found to have a peak frequency response at 2.15 MHz, which was used as a center frequency to drive the toneburst. A 76 *mm* × 30 *mm* rounded rectangular flat-bottom flaw was incrementally introduced into the plate, which can be seen in Figure 5.4(b).

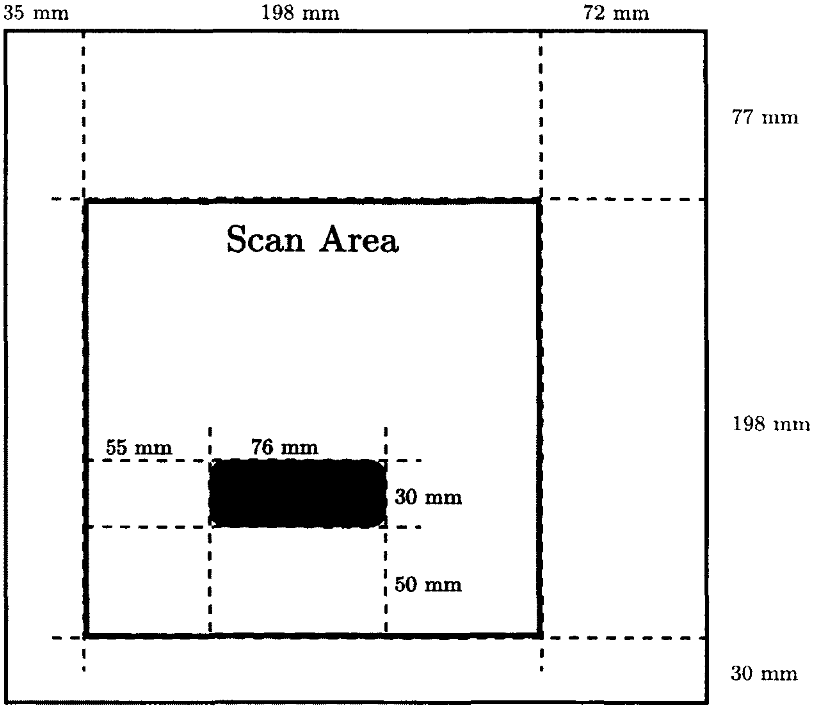
The transducers were stepped through 100 locations per axis in 2 *mm* increments in a double-crosshole geometry [45]. This technique uses linear slides to mimic a four-legged perimeter array of transducers surrounding an area of interest. The motion of both the transmitting and receiving transducers was controlled by a Velmex[3] VP9000 motion controller. Each individual scan was performed in two projections. For each projection, the two linear slides are placed parallel to each other on opposing edges of the scan area. The transmitting transducer is advanced one position along a particular edge while the receiving transducer is stepped through all positions on the opposing edge (Fig. 5.5(a)). The transmitting transducer is then advanced to the next position, and the receiving transducer again steps through all positions on the opposing edge (Fig. 5.5(b)). This continues until all transmitting transducer positions have been reached (Fig. 5.5(c)), completing the projection 1 scan. For the projection 2 scan, the slides on which the transducers are relocated so they are in a second orientation that is 90° relative to the first (Fig. 5.5(d)), and the plate is scanned again. The resulting two datasets are then combined for the full double-crosshole scan (Fig. 5.5(e)). The flaw depth was increased after each full scan, with data sets from fourteen total flaw depths recorded (including 0% and 100% material loss), labeled D_j where $j = 1, \ldots, 14$, summarized in Table 5.1. The experimental apparatus used to collect data is shown in Figure. 5.6.

[2] Lockport, IL (`http://www.gage-applied.com/`)

[3] Bloomfield, NY (`www.velmex.com/`)



(a)



(b)

Figure 5.4: An image of the plate sample in the scanner (a). The flaw severity shown here corresponds to class D_{14}, the through-thickness hole. The dimensions of the scan area as well as the flaw, relative to the plate (b).

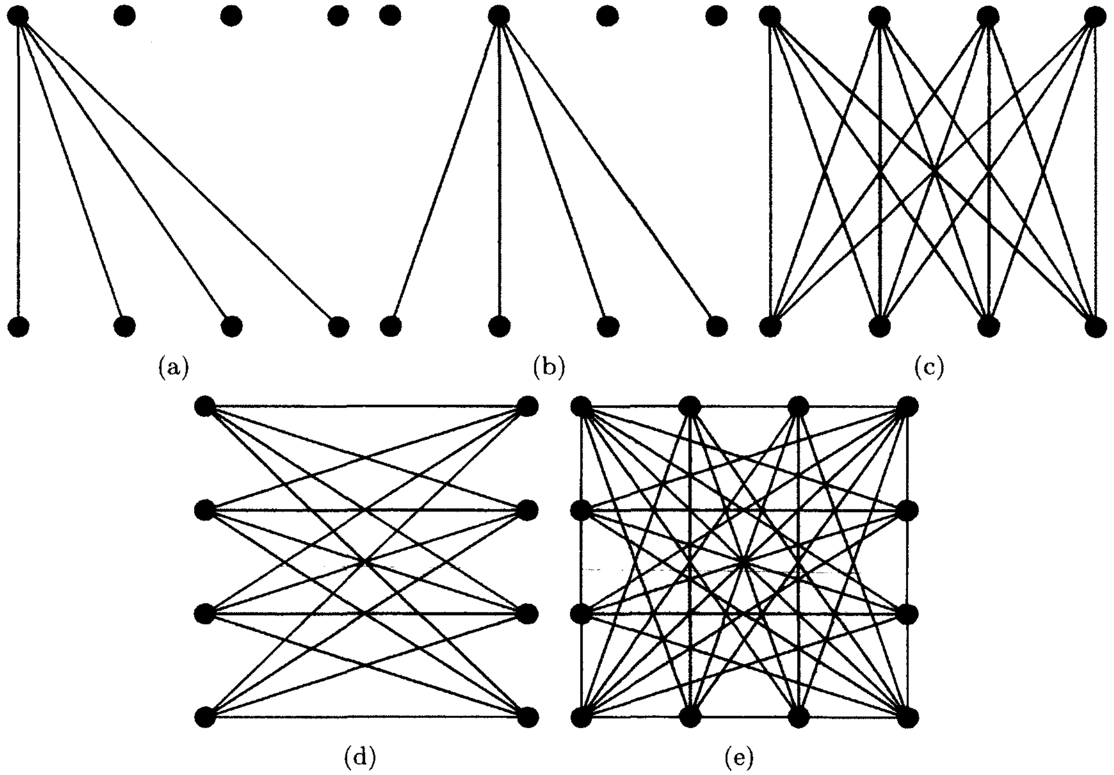


Figure 5.5: The double-crosshole scanning procedure for lamb wave tomography in the laboratory. The transmitting transducer remains fixed while the receiving transducer steps along the length of the plate (a). The transmitting transducer is then advanced one position, and the receiving transducer again steps along the length of the plate (b). This process is repeated until all transmitting transducer locations have been reached (c), an orientation referred to as projection 1. This scan is then performed in a second orientation that is 90° relative to the first (d), referred to as projection 2. Finally, the two datasets are combined for the full double-crosshole scan (e). Although only four transducers positions are show along each side here for clarity, the full scan used in this study contains 100 transducer positions along each side.

Table 5.1: Thickness of the flaw region corresponding to each double-crosshole scan, $D_1, \ldots, D_{14}$. A Starrett digital micrometer was used to measure to thickness in three locations within the flaw, with the average value reported here. The percent thickness loss relative to the full plate thickness is also presented.

Class	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Flaw Depth (mm)	0.00	0.16	0.33	0.56	0.67	0.94	1.14
Percent Thickness Loss	0%	6%	11%	18%	21%	30%	36%
Class	D_8	D_9	D_{10}	D_{11}	D_{12}	D_{13}	D_{14}
Average Flaw Depth (mm)	1.34	1.59	1.86	2.16	2.47	2.69	3.15
Percent Thickness Loss	43%	51%	59%	69%	78%	86%	100%

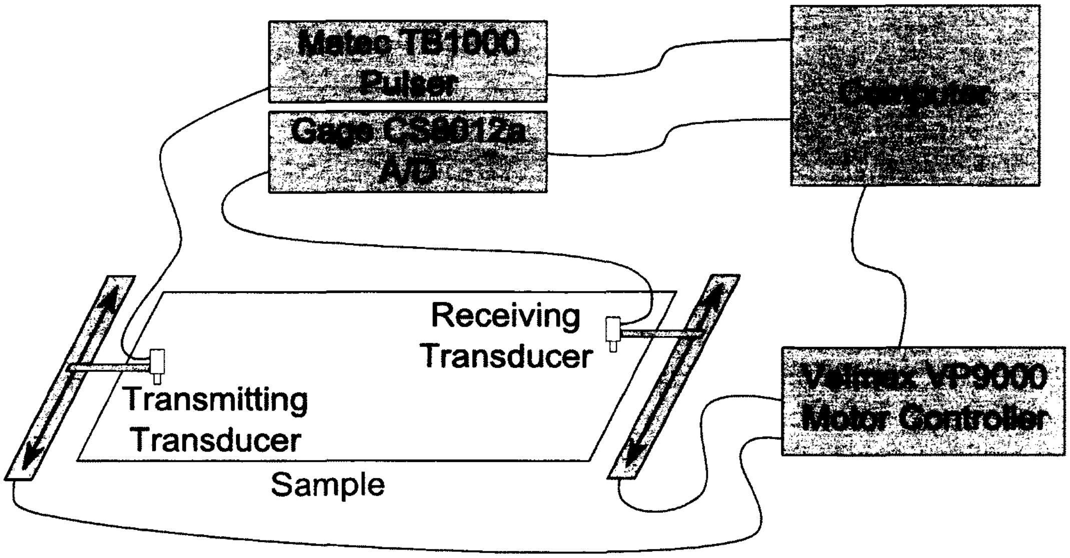
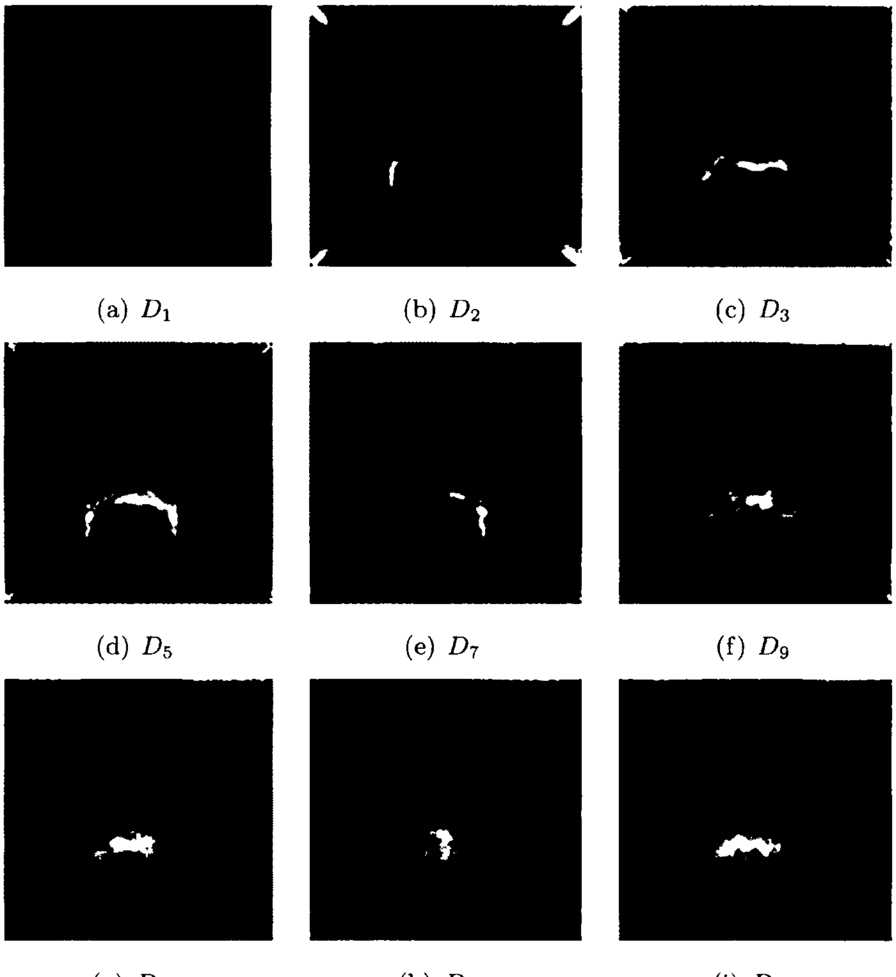


Figure 5.6: Experimental setup block diagram, in which the computer controlled pulser sends a voltage pulse to the transmitting transducer, exciting guided waves in the sample. The receiving transducer converts the sample's vibrations back into voltage charges that are sampled by the A/D converter and stored in the computer. Linear stepper motors allow for the transducers to move in controlled steps along the side of the sample.

5.4 Lamb Wave Tomography

Tomographic reconstructions were generated for each data set D_j using the raw double-crosshole scan data [45]. From each waveform, a specific mode arrival time was extracted, usually the first-arriving using an amplitude threshold. These arrival times were used as input for the simultaneous iterative reconstruction technique (SIRT) to tomographically reconstruct an image of the scanned region [143]. Each image is a gray-scale representation of the scanned region, where the color map of each image spans the range of mode velocities as determined by the SIRT algorithm. By normalizing the color map of each reconstruction this way, *any* areas of the scanned region with differing material properties are highlighted in the reconstruction. An over/under color intensity threshold was then applied to each of the reconstruction images that allowed the flaw to be highlighted and sized. This was done for all the images, building a database of extracted flaw dimensions. The overlapping area for all extracted flaw locations was determined, i.e. the area where every scan indicated a flaw was present. Every raypath from the double-crosshole scan that crossed this common flaw region was then identified and extracted into a "flaw-only" subset of waveforms, reducing the number of waveforms per scan from 20,000 down to 2,919, consistent across all 14 depths. It is important to note that this reduced set of waveforms does not correspond to the total number of raypaths that actually cross the flaw according to the known geometry of the flaw, but rather the raypaths that have a high confidence of having interacted with a flaw in each of the 14 reconstructions extracted automatically from the raw data. Since this process is automated, there is no need to physically access more than the perimeter of the scan area. This is an important advantage of the technique, allowing for areas that are covered by insulation or inaccessible due to complex structural geometries to be checked for flaws. While there exist many applications of SHM that do not require LWT analysis, it serves here as a convenient way to build a training data set to be used for classification. A variety of the raw tomographic reconstruction images from our plate sample can be seen in Figure 5.7. An example of the how the common flaw region is determined as well as a single raypath identified as passing through this common flaw region can be seen in Figure 5.8.

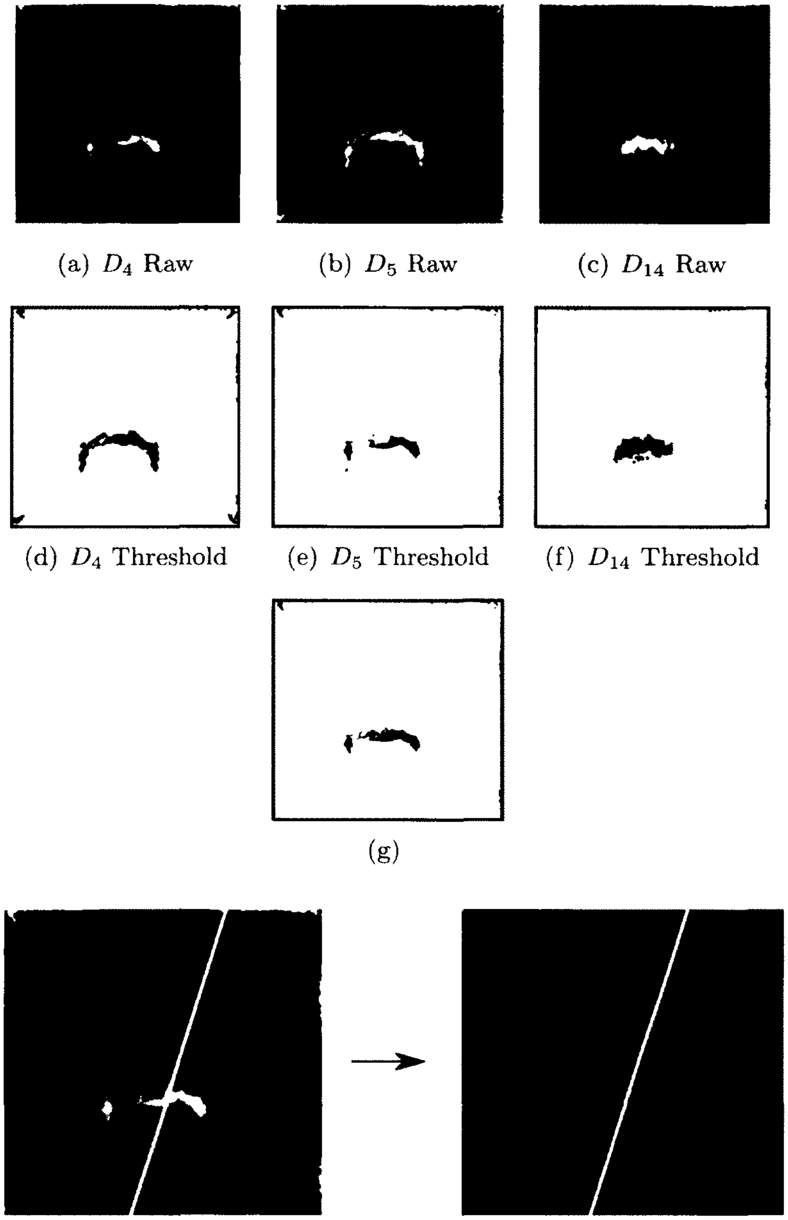

(a) D_1 (b) D_2 (c) D_3

(d) D_5 (e) D_7 (f) D_9

(g) D_{11} (h) D_{13} (i) D_{14}

Figure 5.7: Tomographic reconstructions of the plate at selected flaw depths (a)-(i).

Due to the computation time required to perform feature generation for each waveform, this set of 2,919 'flaw-only' raypaths was reduced further by selecting 100



(a) D_4 Raw

(b) D_5 Raw

(c) D_{14} Raw

(d) D_4 Threshold

(e) D_5 Threshold

(f) D_{14} Threshold

(g)

(h)

Figure 5.8: An example of finding the "common flaw" region. Raw tomographic reconstructions from selected flaw depths (a)-(c) are shown with their over/under threshold equivalents (d)-(f). These thresholded representations are stacked (g) into a composite image, with the overlapping flaw regions seen here as the darkest shade of gray. This overlapping area is sized in the x- and y-dimensions, and then used to identify the subset of raypaths that pass through the flaw. An example of an extracted raypath crossing the flaw (h) in both the raw reconstruction (left) and the common flaw region only (right).

raypaths at random, consistent across all depths. The random selection was also used to minimize geometric bias associated with this specific flaw. This process resulted in 100 waveforms per flaw depth, or 1,400 total waveforms, which were used for feature generation. Waveforms are denoted $w_{i,j}(t)$ where $i = 1, \ldots, 100$ corresponds to the raypath number relative to the scan and $j = 1, \ldots, 14$ corresponds to the flaw depth. A visualization of the raypaths selected can be seen in Figure 5.9.

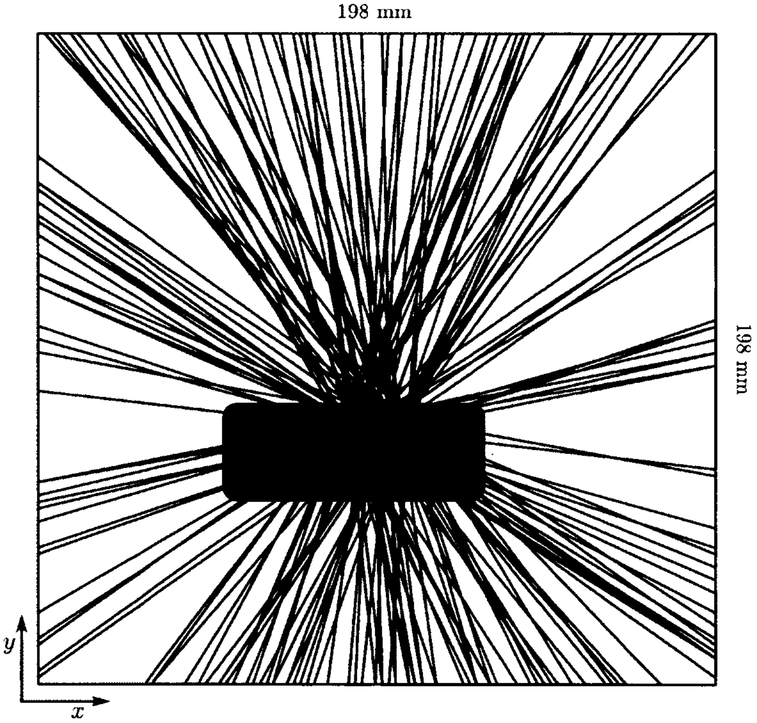


Figure 5.9: The 100 randomly selected raypaths that pass through the flaw region used in this study overlaying the location of the common flaw region (in gray). It is clear that there is no relation between the geometry of the flaw and the orientation of the raypaths being considered for classification.

5.5 Feature Generation

The tomographic reconstruction image analysis routines result in 1400 different waveforms $w_{i,j}(t)$ over the range of individual flaw depths. The next step in processing the raw data involves extracting features from the raw data and selecting M of them for classification. The resulting feature vector is denoted $x_{i,j}(k)$, $k = 1, \ldots, M$. In this application, we generate 78 individual feature values from each waveform ($M = 78$). Several methods are used to extract these features from the raw signals: higher-order statistics, Mellin transform statistics, Wavelet packet Decomposition, and Dynamic Wavelet Fingerprinting (DWFP). All of these methods provide a specific set of values that have direct physical relation back to the raw data, and each has shown promise in various applications for damage detection [23, 44, 64, 152, 153].

Several statistical features commonly used in ultrasonic signal analysis are generated from the raw waveforms [21]. Standard time-domain features include the mean of the raw waveform $w_{i,j}(t)$, the variance, the Shannon entropy, the second central moment, the skewness, and the kurtosis of each waveform [20, 118]. More details can be found in Section 4.5. Statistical moments provide insight by highlighting outliers due to any specific flaw-type signatures found in the data.

The Mellin transform is an integral transform closely related to the Fourier transform that can represent a signal in terms of a physical attribute similar to frequency known as scale. This transform has the key property of scale invariance. Small variations in mode velocity due to environmental changes, like temperature fluctuations or applied strain, can be approximated as a uniform time-scaling effect on the transmitted signals. By considering the Mellin domain, features can be compared that are robust to environmental effects. While there are complexities associated with discretizing the fast Mellin transform algorithm, a MATLAB-based implementation is used [119].

The Mellin transform was formally introduced in Section 4.5, and will be summarized here. The first step in implementing this is to define both an exponential sampling step along with the number of samples needed for a given signal in order to exponentially resample it. Once the exponential axis has been defined, an exponential point-by-point multiplication with the original signal is performed. A fast Fourier transform (FFT) is then computed, followed by an energy normalization step. This process is summarized in Figure 5.10.

After the Mellin transform is computed, features are extracted from the resulting Mellin domain for classification. These features include the mean of the Mellin transform, as well as the standard deviation, variance, second central moment, Shannon entropy, kurtosis, and skewness of the mean-removed Mellin transform [22, 23].

A set of N $(N < M)$ features are extracted from the waveforms using the DWFP technique [37]. By applying a wavelet transform to the raw waveforms, binary images often resembling fingerprints are generated. Image processing techniques are then applied to extract useful information from to make the feature vectors $x_{i,j}(k)$. With the intention of performing feature selection at a later step to reduce the feature space size, we set $N = 50$ here.

The first step in the DWFP algorithm applies a band-pass filter to each waveform $w_{i,j}(t)$ using a stationary wavelet transform. This removes a low-frequency feedback observed in our apparatus as well as high-frequency noise. Next, the DWFP technique applies a continuous wavelet transform to the filtered signals resulting in a set of wavelet coefficients $C_{i,j}(a, b)$, where $a > 0$ is a scaling parameter and $b \in \mathbb{R}$ is a translational parameter for the mother wavelet $\psi(t)$ being used. These coefficients are then sliced in a "thick" contour manner and projected onto the time-scale plane. These resulting contour lines are labeled in an alternating, binary manner, resulting in a "fingerprint" image for each waveform, $I_{i,j}(a, b)$, that are coincident in time with the raw signal. The choice of mother wavelets is application-specific, since certain choices of $\psi(t)$ will be more sensitive to certain types of signal features. Mother wavelets used in this study are the Daubechies-3 (db3), Symelet-5 (sym5), and Meyer

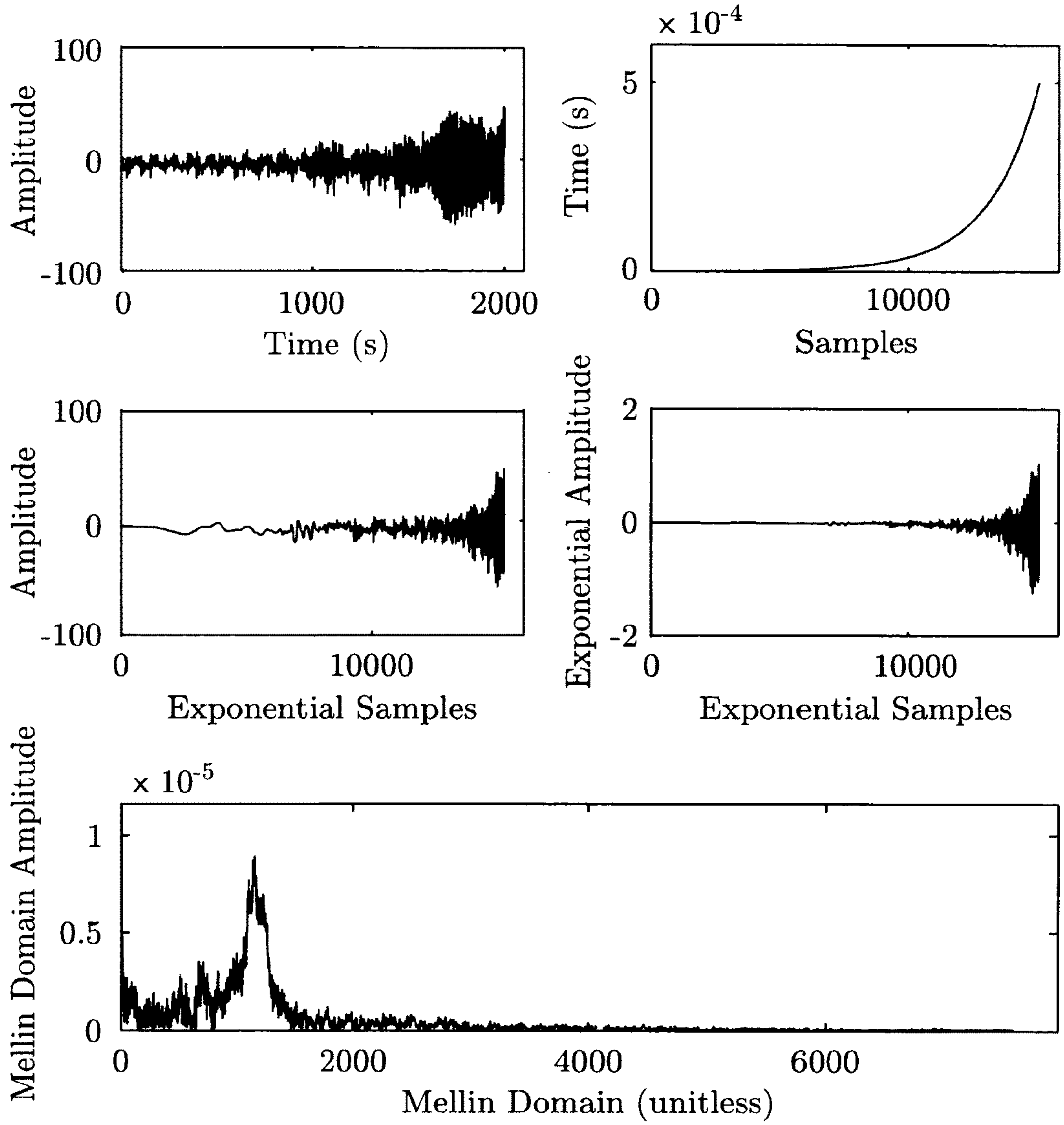


Figure 5.10: A raw signal $w_{i,j}(t)$ (top, left), the exponentially resampled axis (top, right), the signal resampled according to the exponential axis (middle, left), this same signal after point-by-point multiplication of the exponential axis (middle, right), and the resulting Mellin transformed signal (bottom).

(meyr) wavelets, chosen based on preliminary results as well as past experience [37, 39,42,46,61 64].

Since pattern classification uses combinations of one-dimensional feature vectors to develop decision boundaries for each group of observations, the dimension of the binary fingerprint images $I_{i,j}(a,b)$ that are generated for each waveform are first reduced. Each image $I_{i,j}(a,b)$ is composed of a series of individual fingerprint objects, spaced apart in time. Twenty-two measurements are made on each fingerprint object, including the area, centroid, diameter of a circle with the same area, Euler number, convex image, solidity, coefficients of 2nd and 4th degree polynomials fit to the fingerprint boundary, as well as major/minor axis length, eccentricity, and orientation of an ellipse that has the same normalized second central moment as the fingerprint. For more details on these features, see Reference [63]. The property measurements result in a sparse property array $\mathcal{P}_{i,j,n}[t]$, where n represents the property index $n = 1, \dots, 22$, since each extracted value is matched to the time value of the corresponding fingerprint's center of mass. Therefore, these sparse property vectors are linearly interpolated to produce a smoothed vector of property values, $\mathcal{P}_{i,j,n}(t)$.This process is shown for a typical time-domain waveform in Figure 5.11.

Once an array of fingerprint features for each waveform has been generated, it still needs to be reduced into a single vector of $N = 50$ individual values to be used for classification. The goal is to select features from $\mathcal{P}_{i,j,n}(t)$ that will have the best chance to classify each flaw depth. Since this is a multi-class system, feature values are desired that are consistent among each individual flaw depth, yet separable between different flaw depths. First, the dimensionality of the property array is reduced by averaging within each flaw depth, over index $i = 1, \dots, 100$, so $\mathcal{P} = \mathcal{P}_{j,n}(t)$. Then for

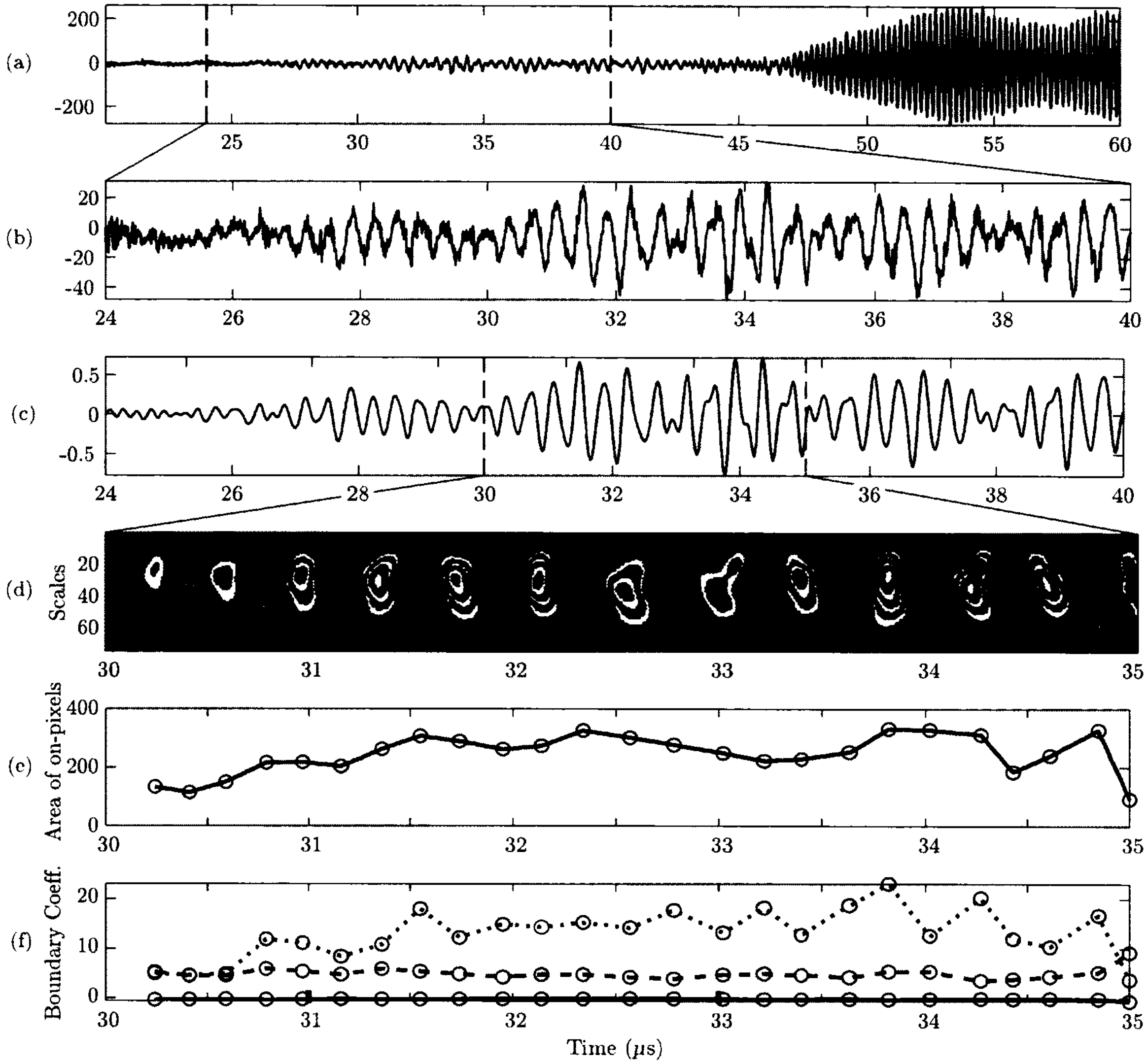


Figure 5.11: A visual summary of the dynamic wavelet fingerprinting (DWFP) [37] feature extraction technique. A typical ultrasonic signal is shown (a), with a close-up portion of the DWFP output shown in (b). Each waveform is first filtered (c), from which the fingerprint image (d) is generated, shown here with white peaks and gray valleys for distinction. Within each fingerprint image, a variety of measures are extracted from each fingerprint object and are interpolated in time, including the area of the on-pixels for each object (e), as well as the coefficients of a second-order polynomial ($p_1 \times x^2 + p_2 \times x + p_3$) fit to the boundary of each fingerprint object, with coefficients p_1 (solid line), p_2 (dashed line), and p_3 (dotted line) shown (f).

each property n, the mean (5.1) and standard deviation (5.2) are calculated by

$$\mu_n(t) = \frac{1}{N}\sum_{j=1}^{N} \mathcal{P}_{j,n}(t) \tag{5.1}$$

and

$$\sigma_n(t) = \sqrt{\frac{1}{N}\sum_{j=1}^{N} (\mathcal{P}_{j,n}(t) - \mu_n(t))^2}. \tag{5.2}$$

Times are then identified t_m, where $m = 1, \ldots, N$, at which the the values of the mean $\mu_n(t_m)$ vary for each flaw depth while the standard deviation $\sigma_n(t_m)$ remains low. The resulting DWFP feature vector for waveform $w_{i,j}(t)$ is

$$x_{i,j}(k_m) = \mathcal{P}_{i,j,n_m}(t_m).$$

Wavelet-based measurements provide the ability to decompose noisy and complex information and patterns into elementary components, making them favorable for multi-mode ultrasonic signal analysis where subtle features can be highlighted with the optimal choice of a mother wavelet.

Another wavelet-based feature used in classification is generated from wavelet packet decomposition [25]. Details of this algorithm can be found in Section 4.5. Wavelet packet decomposition uses redundant basis functions and can therefore provide arbitrary time-frequency resolution details, improving upon the wavelet transform when analyzing signals containing close, high frequency components. Wavelet Packet Decomposition features are included to provide insight into the generation of high-frequency modes as a result of mode conversion at flaw boundaries.

Several other features were included in the feature set. The first of which was the arrival time of the first arriving mode as determined by the Lamb wave tomography algorithms [146]. The tomographic reconstructions are generated from slowness maps, of which determining each waveform's mode arrival time in an intermediate step. These times were extracted and used as features. In addition, several methods were considered for computing the power spectral density (PSD) from which features are extracted [154]. The power spectral density of the signal is a measure of the the power carried by the guided wave, per unit frequency. Calculation techniques include the Welch PSD, the periodogram, the Thompson multitaper PSD, and the covariance PSD. For all of these, the maximum amplitude value and the corresponding frequency were included as features. While several of these measurements can be seen as redundant, for example Welch's method is simply an improvement on the periodogram in that it reduces noise in the power spectra in exchange for a reduction in frequency resolution, referring back to the Ugly Duckling Theorem provides us reason to not favor any specific variation and include all of them in the initial feature set.

A final feature included for consideration is the area under the instantaneous amplitude curve as defined by the Hilbert transform [91]. The Hilbert transform returns the original signal with a 90° phase shift, which means it has the same amplitude and frequency content as the original data. The magnitude of the Hilbert transform and the original signal is known as the instantaneous amplitude, which is another name for the envelope of the signal. The area under this curve was used here as a feature value.

5.6 Classifier Design

In designing a pattern classification routine, one can take one of two general routes: supervised classification in which predefined classes are used to identify new patterns, or unsupervised classification where the data's natural structure is used to determine class values for which new patterns are assigned. The choice of which approach to take is usually objective and depends on the application and the structure of the data at hand. Here, we have data that correspond to a variety of well-defined flaw depths, which makes a supervised approach a natural choice. Therefore, we first need to assign class labels to the waveforms. Since the goal of the classifier is to determine flaw severity, it follows that each flaw depth will be assigned a unique class label, where waveforms corresponding to the unflawed plate are grouped in a unique class (ω_1), waveforms from the shallowest flaw depth are in another class (ω_2), and so on resulting in 14 different classes with labels ω_j, $j = 1, 2, 3, \ldots, 14$.

In statistical pattern classification, each pattern is represented as a point in a multi-dimensional feature space. This space is referred to here as a feature matrix, an example of the feature matrix can be seen in Table 5.2. In this example, each row corresponds to an individual pattern, while each column represents an individual feature. We have generated 78 individual feature values for each waveform, resulting in a 78-dimensional feature space. Referring back to the curse of dimensionality, if we follow the generally accepted limit to use at least ten times as many training samples per class (N) as there are dimensions in the feature space (d), summarized as $N/d \geq 10$, then we should achieve the best classification results with a 10-dimensional feature space. Before we can begin classification, we therefore need to apply feature selection routines in order to reduce the size of our 78-dimensional feature space.

5.6.1 Feature Selection

The only feature selection routine guaranteed to return the optimal subset of a given feature space is an exhaustive search, where a criterion value is calculated for every possible subset of features [9]. This technique is extremely computationally intensive, and is not practical for our 78-dimensional feature set. We instead use the sequential floating backward search algorithm (SFBS) to perform an initial reduction in feature

Table 5.2: A limited selection of the feature space generated from raw waveforms $w_{i,j}(t)$ is presented here to illustrate features and classes. Each row corresponds to a different waveform $w_{i,j}(t)$, and each column corresponds to a different feature $k = 1, \ldots, M$. The waveforms are divided into classes ω_j, each composed of $i = 1, \ldots, 100$ different waveforms. The classifier's goal is to generate partitions in this feature space that accurately separate the classes from each other, allowing for new data to be assigned a class label based on where it lies relative to these partitions.

	Features $\mathbf{x}_{i,j}(k)$				
	k = 1	k = 2	···	k = M	
i	WPD	Mean of $w_{i,j}(t)$	···	DWFP #50	Class ω_j
1	1.1272	-2.8623		1.6116	ω_1
2	1.0704	-1.6196	···	1.2391	ω_1
3	1.1693	-1.6277		2.4858	ω_1
⋮	⋮	⋮		⋮	⋮
99	1.1693	-0.3029		1.7567	ω_1
100	-0.0372	3.4129		1.5786	ω_1
1	0.3731	1.0484	···	0.2062	ω_2
2	1.1332	1.0305		0.1272	ω_2
3	0.5196	0.2728		1.2579	ω_2
⋮	⋮	⋮		⋮	⋮
99	1.1259	1.4728		1.1702	ω_2
100	0.0745	-1.7211		1.0354	ω_2
1	-0.2638	0.9387	···	1.5261	ω_3
2	1.1693	1.0190		0.9328	ω_3
3	0.4310	-0.8075		-0.2722	ω_3
⋮	⋮	⋮		⋮	⋮
99	0.0813	0.1574		-1.0070	ω_{13}
100	-0.2383	0.1584		-0.9236	ω_{13}
1	-2.6853	0.1497	···	-1.0612	ω_{14}
2	0.2176	0.2463		-1.0426	ω_{14}
3	-3.4866	-0.5856		-0.9856	ω_{14}
⋮	⋮	⋮		⋮	⋮
99	-0.2638	0.3636	···	-1.0448	ω_{14}
100	-2.0391	0.2913		-1.0437	ω_{14}

space dimension. The SFBS technique was first introduced by Pudil et al. [155], and has been shown to provide close to the optimal solution while having a reasonable computational cost [9]. The SFBS technique is a version of the plus-l-takeaway-r algorithm, where the values of l and r are updated dynamically. We use classification accuracy of the 1-nearest neighbor classifier (1NN) as a criterion function, which the algorithm maximizes through an iterative search of feature subsets. Results from the SFBS feature selection routine can be seen in Figure 5.12 as a plot of criterion value versus the feature space dimension d for $d = 1, \ldots, 78$. It can be seen that the SFBS determines a subset of 10 features to be optimal, as expected, with the curse of dimensionality phenomenon clearly seen as an increase in feature space dimension leads to lower classification accuracies. From these results, we identify the best 25 features, indicated by the dotted line in Figure 5.12. This reduced space is more appropriately sized to perform an exhaustive search on, and will allow us to compare several feature selection routines in an attempt to identify a more optimal feature space of roughly 10 dimensions.

We explore feature selection metrics that use geometric measures on the class distribution within the feature space itself, for all feature space subsets of dimension $d = 1, \ldots, 25$. For each feature space subset of dimension d, we first calculate each of the 14 class centroid locations. These centroids are then used to measure several values, including average interclass distance, average angle between vectors constructed between successive class centroids, and the tortuosity of a line connecting all centroids sequentially.

We include the average distance between centroids under the assumption that classes that are spaced farther apart in the feature space will be easier to distinguish between, reducing the misclassification rate. This measure breaks down when the classes have large variance and still overlap significantly within the feature space. We

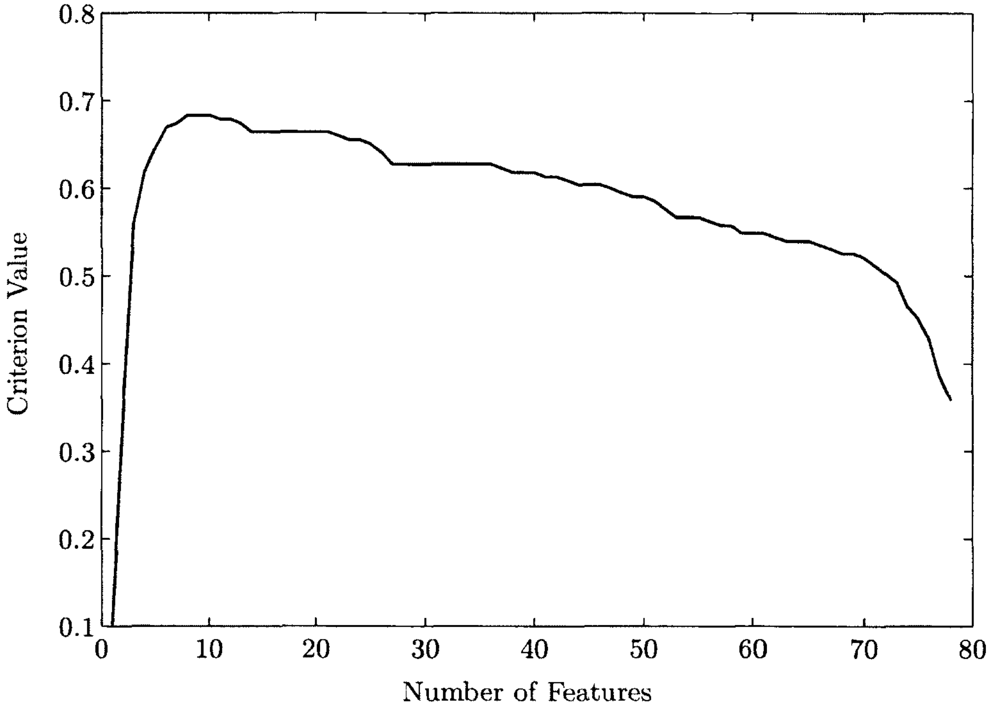


Figure 5.12: Results of the SFBS feature selection algorithm. The feature selection criterion here is the minimum nearest neighbor classification accuracy. The curse of dimensionality can be seen here, where the criterion value peaks at 10 features and then slowly drops off as more features are included.

first calculate the corresponding vectors between successive n-dimensional centroids, $C_j = [c_{j,1}, c_{j,2}, \ldots, c_{j,n}]$ where $j = 1, \ldots, 14$, using

$$\begin{aligned} \mathbf{v}_i &= C_{i+1} - C_i \\ &= (c_{i+1,1} - c_{i,1}, c_{i+1,2} - c_{i,2}, \ldots, c_{i+1,n} - c_{i,n}) \end{aligned} \tag{5.3}$$

for $i = 1, \ldots, |j| - 1$, where $|j|$ indicates the magnitude of j. We then take the Euclidean norm of each vector, $||\mathbf{v}_i||$, and average them to get a single average centroid distance value for each feature space,

$$\overline{||\mathbf{v}||} = \frac{1}{|j| - 1} \sum_i ||\mathbf{v}_i||. \tag{5.4}$$

These values are sorted by highest average distance, with the maximum value corresponding to the top feature subspace of each dimension.

We include the average angle between vectors constructed between successive class centroids under the assumption that a linear spread of classes would allow new data corresponding to an intermediate flaw depth to lie correctly between classes. For example, if data corresponding to a flaw 54% of thickness loss is introduced to this classifier, it would ideally fall somewhere in the feature space between classes D_9 (51% thickness loss) and D_{10} (59% thickness loss). If the classes are linearly distributed in sequential order within the feature space, then this new data would have the highest probability to be associated with either of these bounding classes. If the classes are not linearly distributed in the feature space, there is opportunity for an alternate class to lie in this intermediate area, which would result in a misclassification. From equations (5.3) and (5.4) we calculate the angle between successive vectors using the dot product

$$\theta_m = \cos^{-1}\left(\frac{\mathbf{v}_m \cdot \mathbf{v}_{m+1}}{\|\mathbf{v}_m\| \, \|\mathbf{v}_{m+1}\|}\right) \quad \text{for} \quad m = 1, \ldots, |i| - 1. \tag{5.5}$$

From these, we calculate the average angle between successive vectors

$$\bar{\theta} = \frac{1}{|i| - 1} \sum_m \theta_m \tag{5.6}$$

We sort these values by lowest average angle in order to identify a feature space where the classes are most linearly distributed, with the minimum value corresponding to the top feature subspace of each dimension.

Additionally, we calculate the tortuosity of each centroid curve. Tortuosity (τ) is defined here as the ratio of the length of the curve connecting the successive class centroids to the distance between the ends of it,

$$\tau = \frac{\sum_i \|\mathbf{v}_i\|}{\|C_{|j|} - C_1\|} \qquad \text{where} \qquad 1 \leq \tau < \inf. \tag{5.7}$$

A value of $\tau = 1$ corresponds to a linear sequential centroid curve. We sort tortuosity in ascending order, with the minimum value corresponding to the top feature subspace of each dimension. An example of these measures can be seen in Figure 5.13.

Since we have already calculated its results, we include the SFBS routine as well. One drawback of using a formal feature selection method is that it is directly dependent on the training data set as well as the criterion selection. We have reservations about relying on a feature selection method that uses formal classification accuracy as a criterion value due to the 'sequential' ordering of the classes in our data set. That is, formal classification accuracy is highest when class separation within the feature space is optimized, without regard to the specific ordering of those classes.

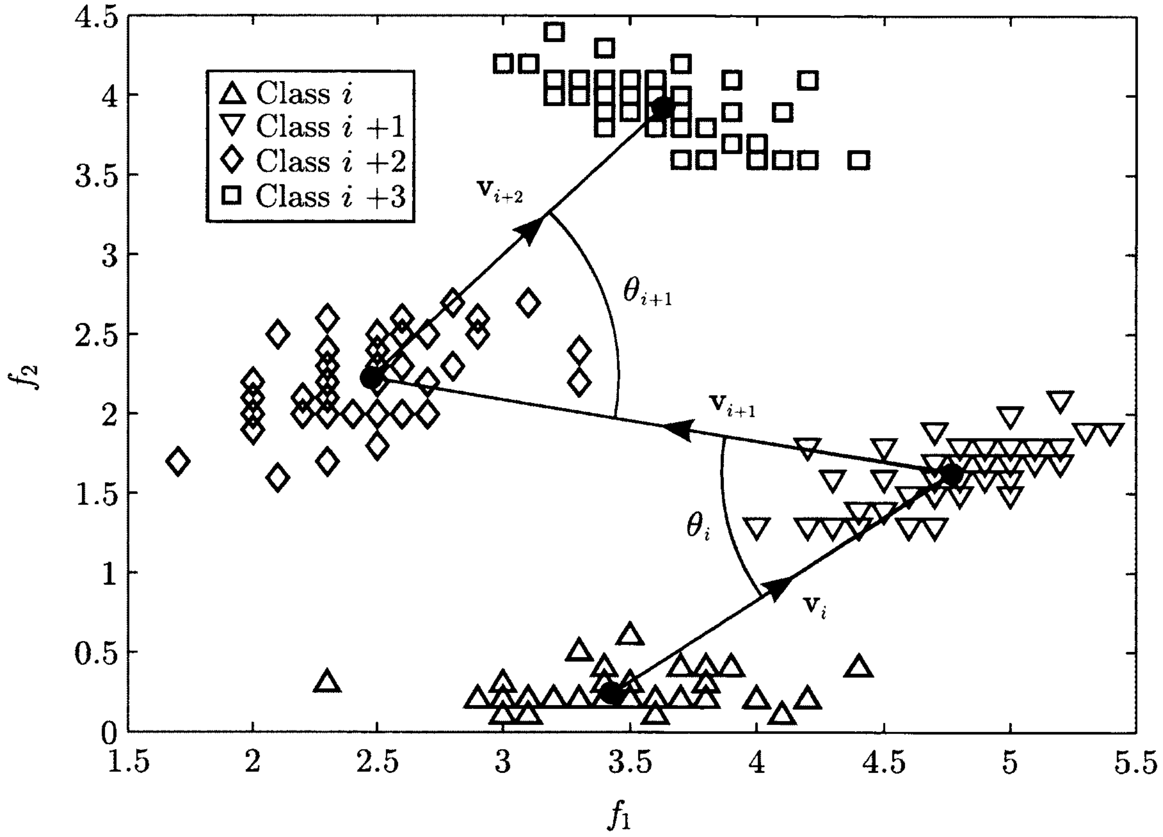


Figure 5.13: An example of a 4-class data distribution in a 2-dimensional feature space, with classes i ($\triangle$), $i+1$ ($\triangledown$), $i+2$ ($\Diamond$), and $i+3$ ($\Box$). Class centroids are denoted C_i, C_{i+1}, C_{i+2}, and C_{i+3}, respectively. Vectors $\mathbf{v}_i = C_{i+1} - C_i$, $\mathbf{v}_{i+1} = C_{i+2} - C_{i+1}$, and $\mathbf{v}_{i+2} = C_{i+3} - C_{i+2}$ are shown here. The angle between vectors $\mathbf{v}_i$ and $\mathbf{v}_{i+1}$ is θ_i, while that between vectors $\mathbf{v}_{i+1}$ and $\mathbf{v}_{i+2}$ is θ_{i+1}. The tortuosity (τ) is defined here as the sum of the length of vectors that make up the curve, $\|\mathbf{v}_i\| + \|\mathbf{v}_{i+1}\| + \|\mathbf{v}_{i+2}\|$, divided by the distance between end points of the curve, $\|C_{i+3} - C_i\|$.

The number of distinct d-feature subsets that can be formed is given by the binomial coefficient $\binom{25}{d}$. For each of these subsets, criterion values for each of these metrics are calculated. If a dimension d is being considered where $\binom{25}{d} > 200$, only the top 200 criterion values are stored. Once all calculated, the top metric criterion values are identified and their corresponding feature sets are used to determine an overall classification accuracy that can be compared between metrics and classifiers. Before we do this, however, we must select which classifiers to use.

5.6.2 Classifier Selection

The No Free Lunch Theorem states that there is no inherently best classifier for a particular application, and often times in practice several classifiers are compared and contrasted. As previously discussed, there exists a hierarchy of possible choices that are application dependent. We have previously decided that supervised, statistical pattern classification techniques using both parametric and nonparametric probability-based classifiers are appropriate for our application.

For parametric classifiers, we include a linear classifier using normal densities (LDC), a quadratic classifier using normal densities (QDC), an uncorrelated quadratic classifier using normal densities (UDC), a multi-class implementation of a logistic linear classifier (LOGLC), as well as a linear classifier using an initial principal component analysis projection step (PCLDC). For nonparametric classifiers, we include a nearest mean classifier (NMC), a k-nearest-neighbor classifier (KNNC) for $k = 1, 2, 3$, and a minimum least square linear classifier (FISHERC). The mathematical explanations for these classifiers can be found in [1,2,4–6,9,124–126]. For implementation of these classifier functions, we use routines from the MATLAB toolbox PRTools [126].

For the above classifiers that output densities (LDC, QDC, UDC, etc.) or unnormalized confidences (FISHERC), a function is applied that converts the output to a proper confidence interval, where the sum of the outcomes is one for every test object. This allows for comparison between classifier outputs. Since each waveform's feature vector is assigned a confidence value for each class, the final label is decided by the highest confidence of all the classes. Often there will exist several peaks in this confidence vector for a single feature vector, an indication that the classifier is having difficulty distinguishing between classes. This similarity is the reason complicated classification techniques are needed in the first place, and still often result in misclassifications. Since our motivation is structural health monitoring, the goal here is then to develop a classifier that is accurate enough to predict labels within a safe bound (e.g. ±1 flaw depth) of their true severity.

5.7 Results

As previously mentioned, the originial 78-dimensional feature set was reduced to a 25 feature subset using a sequential floating backward search (SFBS) algorithm. From these reduced features, all possible subsets of dimensions $d = 1, \dots, 25$ were used to calculate the three geometric feature selection criterion values: average distance between class centroids ($\overline{\|\mathbf{v}\|}$), average angle between successive connecting class centroid vectors ($\bar{\theta}$), and tortuosity of each centroid curve (τ). The feature subsets corresponding to the 'best' 200 criterion values (highest $\overline{\|\mathbf{v}\|}$, lowest $\bar{\theta}$, and lowest τ) for each dimension $d = 1, \dots, 25$ are compared using each of the classification algorithms presented in Section 5.6.2. The waveforms included in each training set T are submitted to the individual classifiers as training data using only the features identified by the feature space under consideration. A corresponding testing subset R is then submitted to the trained classifier, again using only those features defined by the feature space, and the resulting predicted labels are stored in an array. These labels can be easily viewed in a confusion matrix $\mathcal{L}(j_1, j_2)$, whose entries are the percent of testing waveforms from class D_{j_1} (known label ω_{j_1}) that are predicted to be in class D_{j_2} (predicted label ω_{j_2}). This is given by

$$\mathcal{L}(j_1, j_2) = \frac{|\omega_{j_1} = \omega_{j_2}|}{|\omega_{j_1}|} \quad \text{when} \quad j_1, j_2 \in j = 1, \ldots, 25 \tag{5.8}$$

A perfect classifier would therefore have values of $\mathcal{L} = 1$ whenever $\omega_{j_1} = \omega_{j_2}$ (on the diagonal) and values of $\mathcal{L} = 0$ when $\omega_{j_1} \neq \omega_{j_2}$ (off-diagonal). An example confusion matrix for a classifier can be seen in Table 5.3. In this confusion matrix, the number of waveforms accurately predicted to be their correct known label lie on the diagonal entries, while the number predicted to be within ±1 flaw depth also include the waveforms within one predicted label on either side of the diagonal. For example, this classifier corretly assigned labels of ω_{11} for only 9 waveforms from class D_{11} out of the 20 tested, resulting in 45% accuracy. When accuracy with ±1 flaw depth is considered, this accuracy rises to 85% since 17 total waveforms were given labels of ω_{10} - ω_{12} (5 labeled ω_{10}, 9 labeled ω_{11}, and 3 labeled ω_{12}).

Given the number of classifier configuration parameters used in this study, it does not make sense to compare individual confusion matrices to each other to determine classifier ability. Instead, we compute the overall classifier accuracy for all testing waveforms (Exact) as well as the accuracy of labels within ±1 flaw depth for all testing waveforms (±1 Depth). Results for each classifier using feature spaces corresponding to the 200 highest $\overline{\|\mathbf{v}\|}$, lowest $\bar{\theta}$, and lowest τ are presented in Tables 5.4 - 5.12. These results are given in terms of the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique, where randomly selected subsets of both the testing and training waveforms are used to test and train each classifier, respectively. This allows us to establish a sense of

Table 5.3: A confusion matrix $\mathcal{L}(j_1, j_2)$ for the QDC classifier corresponding to the 6-dimensional feature subspace identified by the SFBS algorithm. The values in the matrix are the number of waveforms from the true flaw depth j_1 that are labeled as flaw depth j_2. A perfect classifier would have 100% accurate labels on the diagonal ($j_1 = j_2$), and zero misclassifications for any off-diagonal elements ($j_1 \neq j_2$).

		Predicted Labels														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	Totals
	14	1	2	0	0	0	0	0	0	0	0	1	0	1	11	16
	13	0	0	2	0	0	3	0	0	0	0	0	3	11	1	20
	12	0	1	0	0	0	0	3	0	0	1	5	10	0	0	20
	11	0	0	2	0	0	0	0	0	0	5	9	3	0	1	20
True Labels	10	0	0	0	0	0	0	0	0	1	11	5	3	0	0	20
	9	0	0	0	1	2	0	0	0	13	1	1	2	0	0	20
	8	0	0	0	0	0	0	0	20	0	0	0	0	0	0	20
	7	0	0	1	0	2	0	11	0	0	1	1	3	1	0	20
	6	0	0	2	3	0	10	0	0	0	0	0	0	4	1	20
	5	0	0	0	0	14	0	0	0	0	0	0	3	0	0	17
	4	1	0	1	12	1	1	2	0	1	0	0	0	1	0	20
	3	0	0	13	2	0	1	0	0	0	0	0	0	3	1	20
	2	0	15	0	0	0	0	1	0	0	0	0	0	0	0	16
	1	20	0	0	0	0	0	0	0	0	0	0	0	0	0	20
Totals		22	18	21	18	19	15	17	20	15	19	22	27	21	15	269

bounds on the stability of the classification accuracy. Results with high standard deviations here correspond to classifiers that are unstable in their performance, while those with low standard deviations returned consistent classification performance. We have included classification accuracies for the optimal SFBS feature subset for each dimension $d = 1, \ldots, 25$ as well. This provides a basis for comparison to determine how well the geometric feature selection metrics perform against the formal SFBS routine. For each metric as well as the SFBS results, the feature subset dimension corresponding to the *highest* classification accuracy (for both exact and ±1 depth) is highlighted in gray.

For almost all of the classifiers tested, the maximum average distance between centroids ($\overline{\|\mathbf{v}\|}$) returns the highest classification accuracy of the three geometric metrics (tortuosity τ, average angle $\bar{\theta}$, average centroid distance $\overline{\|\mathbf{v}\|}$). Additionally, with the exception of the three k-NN classifiers, the average distance between centroids metric even outperformed the SFBS 'optimal' feature set. This follows from the fact that kNN classifier accuracy (specifically, 1NN) was used as a criterion function for developing the SFBS feature set. If an alternate classifier had been used as a criterion function, the results would change accordingly. This shows that even formal feature selection techniques may not always identify the 'best' feature space subsets.

Several of the individual classifiers performed noticeably poorer than the rest. The near mean classifier (NMC) computes the Euclidean distance from each feature vector to each of the j class mean vectors, assigning the feature vector a label to the class of the nearest mean. While similar in nature to the better-performing but more computationally intensive k-NN classifiers, overlapping classes in the feature space (as previously discussed) will clearly result in misclassifications that have potential to be avoided in the k-NN classifiers. This is seen as lower classification accuracies across the board for the NMC classifier relative to the rest of the classifiers.

Table 5.4: Classification accuracies for the LDC classifier using the top feature set of dimension d as determined by the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	LDC Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	17.5 ± 2.0	33.5 ± 3.6	17.5 ± 2.0	33.5 ± 3.6	17.5 ± 2.0	33.5 ± 3.6	15.0 ± 1.3	29.8 ± 2.6
2	32.1 ± 3.6	42.5 ± 4.6	32.1 ± 2.7	44.9 ± 4.1	32.1 ± 3.0	44.9 ± 3.9	26.9 ± 2.6	36.7 ± 3.4
3	28.7 ± 2.8	43.8 ± 3.4	29.7 ± 3.5	40.9 ± 4.4	43.2 ± 2.8	53.3 ± 4.2	41.2 ± 2.6	56.7 ± 2.9
4	23.5 ± 2.8	42.5 ± 3.3	31.7 ± 3.8	44.9 ± 4.4	48.8 ± 3.1	57.7 ± 3.7	44.1 ± 2.1	58.0 ± 2.1
5	21.5 ± 2.6	37.5 ± 3.1	33.0 ± 3.8	47.5 ± 4.7	50.8 ± 3.2	60.4 ± 3.8	43.6 ± 1.3	56.7 ± 0.9
6	21.6 ± 2.5	37.8 ± 3.4	34.6 ± 4.1	48.3 ± 4.7	52.4 ± 3.1	63.1 ± 4.0	44.3 ± 2.0	58.0 ± 1.8
7	24.5 ± 2.8	41.8 ± 3.4	37.1 ± 4.0	50.0 ± 4.7	55.1 ± 3.3	66.4 ± 4.0	44.0 ± 1.2	58.2 ± 1.5
8	25.1 ± 3.2	42.0 ± 3.5	37.5 ± 3.8	50.3 ± 4.7	58.1 ± 3.5	68.4 ± 3.7	50.3 ± 1.7	64.1 ± 2.7
9	25.1 ± 3.5	42.0 ± 3.6	39.3 ± 3.8	53.2 ± 4.3	59.4 ± 3.5	68.9 ± 3.8	51.8 ± 2.5	64.8 ± 2.4
10	29.1 ± 3.5	44.0 ± 3.6	39.8 ± 3.6	53.3 ± 4.2	60.5 ± 3.5	71.0 ± 3.5	51.8 ± 2.5	64.8 ± 2.4
11	30.1 ± 3.2	45.2 ± 3.5	40.0 ± 3.8	53.3 ± 4.4	61.3 ± 3.7	71.5 ± 3.6	55.8 ± 2.3	66.8 ± 2.8
12	33.5 ± 3.1	44.8 ± 3.4	42.3 ± 3.9	55.4 ± 4.4	62.4 ± 3.6	72.3 ± 3.4	55.8 ± 2.3	66.8 ± 2.8
13	37.3 ± 3.2	49.3 ± 3.4	46.0 ± 4.4	60.3 ± 5.0	63.1 ± 3.4	73.1 ± 2.9	59.8 ± 1.6	69.4 ± 1.5
14	38.0 ± 3.5	49.4 ± 3.2	47.5 ± 4.1	61.5 ± 4.5	63.8 ± 3.2	73.3 ± 2.8	61.0 ± 2.8	70.6 ± 2.0
15	38.7 ± 3.5	50.2 ± 3.2	51.7 ± 3.9	65.1 ± 4.3	64.3 ± 3.3	73.5 ± 2.8	60.8 ± 2.9	70.4 ± 2.1
16	41.9 ± 3.7	53.9 ± 3.6	52.0 ± 4.0	65.4 ± 3.8	64.7 ± 3.3	74.1 ± 2.7	55.3 ± 2.1	65.6 ± 2.6
17	47.3 ± 3.9	57.4 ± 3.6	52.0 ± 4.0	65.4 ± 3.2	64.8 ± 3.3	74.1 ± 2.9	55.4 ± 2.0	65.8 ± 2.4
18	54.1 ± 3.9	62.6 ± 3.2	52.1 ± 3.9	65.5 ± 3.3	65.1 ± 3.5	74.6 ± 2.9	55.4 ± 2.0	65.8 ± 2.4
19	54.1 ± 3.9	62.6 ± 3.0	57.4 ± 3.7	69.0 ± 3.1	65.1 ± 3.6	74.6 ± 2.9	59.5 ± 2.3	69.7 ± 2.1
20	59.3 ± 3.2	67.2 ± 2.9	58.1 ± 4.5	69.9 ± 3.1	65.2 ± 3.7	74.7 ± 3.2	59.5 ± 2.3	69.7 ± 2.1
21	62.0 ± 3.7	69.8 ± 2.8	63.0 ± 3.9	71.3 ± 2.8	**65.3 ± 3.8**	**74.8 ± 2.9**	59.4 ± 3.0	69.2 ± 2.8
22	64.2 ± 3.5	73.0 ± 3.1	**65.1 ± 3.5**	73.3 ± 2.7	65.1 ± 3.6	74.5 ± 3.1	63.4 ± 2.9	72.3 ± 2.7
23	**65.1 ± 3.4**	73.4 ± 2.6	**65.1 ± 3.7**	**73.8 ± 2.8**	64.9 ± 3.2	74.4 ± 2.9	61.4 ± 2.6	71.0 ± 1.9
24	65.0 ± 3.5	**73.8 ± 2.8**	65.0 ± 3.5	**73.8 ± 2.8**	65.0 ± 3.5	73.8 ± 2.8	64.1 ± 2.9	73.3 ± 2.4
25	64.3 ± 2.5	73.5 ± 2.4	64.3 ± 2.5	73.5 ± 2.4	64.3 ± 2.5	73.5 ± 2.4	**64.3 ± 2.5**	**73.5 ± 2.4**

Table 5.5: Classification accuracies for the QDC classifier using the top feature set of dimension d as determined by the lowest average tortuosity ($\overline{\tau}$), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	QDC Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	17.9 ± 2.2	33.5 ± 3.0	17.9 ± 2.2	33.5 ± 3.0	17.9 ± 2.2	33.5 ± 3.0	15.8 ± 1.7	30.6 ± 2.3
2	36.0 ± 3.1	48.5 ± 3.2	41.0 ± 2.9	54.1 ± 3.8	41.0 ± 4.1	54.1 ± 6.0	32.2 ± 1.7	44.3 ± 3.0
3	42.8 ± 3.1	55.5 ± 4.0	40.1 ± 3.4	52.3 ± 4.2	51.4 ± 7.7	64.2 ± 9.8	57.7 ± 2.3	68.7 ± 2.1
4	27.4 ± 3.5	43.9 ± 4.3	43.0 ± 3.0	53.6 ± 3.5	58.7 ± 12.2	68.2 ± 13.7	60.0 ± 1.6	70.0 ± 1.6
5	21.2 ± 3.4	40.3 ± 4.0	43.0 ± 3.1	55.0 ± 4.0	60.9 ± 16.1	70.7 ± 16.9	53.0 ± 8.7	64.9 ± 10.0
6	20.8 ± 3.0	39.8 ± 4.2	44.3 ± 3.1	56.0 ± 4.0	61.8 ± 16.1	71.9 ± 17.3	54.6 ± 5.6	65.6 ± 7.7
7	26.5 ± 3.2	41.4 ± 4.4	45.6 ± 3.0	57.1 ± 3.7	67.9 ± 15.6	75.7 ± 17.0	55.4 ± 5.5	64.8 ± 6.8
8	26.2 ± 2.8	42.2 ± 4.3	46.4 ± 3.1	57.9 ± 3.4	66.3 ± 12.5	74.2 ± 13.2	66.3 ± 4.7	74.3 ± 4.6
9	26.1 ± 2.9	43.1 ± 4.1	48.6 ± 3.0	60.7 ± 3.4	70.2 ± 11.5	77.4 ± 11.6	68.2 ± 4.1	75.5 ± 4.0
10	37.9 ± 2.5	51.3 ± 3.5	50.7 ± 3.0	61.0 ± 3.4	72.7 ± 9.7	78.8 ± 9.2	68.3 ± 4.1	75.6 ± 3.9
11	39.4 ± 3.1	53.0 ± 3.5	52.5 ± 3.1	62.2 ± 2.9	73.3 ± 8.6	79.9 ± 7.6	68.3 ± 3.8	74.8 ± 3.5
12	40.4 ± 2.9	53.5 ± 3.5	53.4 ± 3.2	62.8 ± 2.8	74.3 ± 7.2	80.3 ± 6.6	68.2 ± 3.7	74.8 ± 3.4
13	43.8 ± 3.4	55.9 ± 3.2	56.9 ± 3.3	67.2 ± 2.8	74.8 ± 5.9	80.3 ± 5.2	69.1 ± 4.0	75.7 ± 3.9
14	47.5 ± 3.1	58.3 ± 3.2	59.1 ± 3.4	68.7 ± 3.1	75.3 ± 5.9	80.4 ± 5.2	72.2 ± 4.7	79.1 ± 4.2
15	50.9 ± 3.4	62.2 ± 2.6	62.2 ± 3.5	71.6 ± 3.0	75.2 ± 5.1	80.4 ± 4.6	72.1 ± 4.7	**79.4 ± 4.0**
16	53.9 ± 3.9	64.1 ± 3.0	62.7 ± 3.5	71.7 ± 3.0	**75.6 ± 5.0**	**80.7 ± 4.9**	64.5 ± 5.3	72.5 ± 6.0
17	55.5 ± 3.9	65.9 ± 2.8	62.9 ± 3.3	71.7 ± 2.8	74.8 ± 5.1	80.5 ± 4.8	66.1 ± 3.9	74.5 ± 4.5
18	65.0 ± 3.7	71.8 ± 3.2	62.9 ± 3.4	71.4 ± 3.0	74.9 ± 5.0	80.4 ± 4.8	66.1 ± 4.0	74.5 ± 4.5
19	65.6 ± 3.8	72.1 ± 3.2	71.1 ± 2.8	77.6 ± 2.8	74.8 ± 4.8	**80.7 ± 4.5**	67.1 ± 4.2	74.7 ± 4.5
20	70.1 ± 4.1	77.0 ± 3.4	71.8 ± 2.8	77.7 ± 2.8	74.5 ± 4.6	80.4 ± 4.5	67.0 ± 4.2	74.6 ± 4.5
21	72.4 ± 3.3	79.1 ± 3.1	72.9 ± 3.3	78.5 ± 2.8	74.5 ± 4.6	80.3 ± 4.3	67.5 ± 3.7	74.8 ± 4.2
22	74.4 ± 4.2	80.0 ± 4.7	**75.5 ± 3.1**	**80.7 ± 2.9**	74.2 ± 4.1	80.4 ± 4.3	71.4 ± 4.1	77.9 ± 3.9
23	**75.2 ± 4.1**	**80.7 ± 4.6**	75.2 ± 3.9	**80.7 ± 4.6**	74.4 ± 3.9	80.3 ± 4.3	72.3 ± 3.1	79.1 ± 3.3
24	74.6 ± 3.8	80.4 ± 3.9	74.6 ± 3.8	80.4 ± 3.9	74.6 ± 3.8	80.4 ± 3.9	72.9 ± 3.4	79.1 ± 3.5
25	73.2 ± 3.3	79.2 ± 3.6	73.2 ± 3.3	79.2 ± 3.6	73.2 ± 3.3	79.2 ± 3.6	**73.2 ± 3.3**	79.2 ± 3.6

Table 5.6: Classification accuracies for the 1NN classifier using the top feature set of dimension d as determined by the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	1NN Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	19.3 ± 2.4	33.5 ± 2.6	19.3 ± 2.4	33.5 ± 2.6	19.3 ± 2.4	33.5 ± 2.6	11.2 ± 1.4	26.7 ± 2.1
2	35.6 ± 3.0	45.9 ± 3.4	38.2 ± 2.7	50.6 ± 3.3	38.2 ± 3.5	50.6 ± 3.7	34.1 ± 1.8	45.1 ± 2.6
3	38.7 ± 3.2	51.3 ± 3.7	44.1 ± 3.0	55.0 ± 3.4	56.1 ± 3.0	65.6 ± 3.6	59.2 ± 2.2	69.4 ± 1.8
4	30.2 ± 3.0	45.5 ± 3.8	45.2 ± 3.3	55.9 ± 4.1	58.9 ± 3.3	69.6 ± 3.9	62.6 ± 2.0	72.5 ± 2.1
5	26.1 ± 3.1	42.1 ± 4.0	45.9 ± 3.5	58.8 ± 4.3	62.3 ± 3.1	71.8 ± 3.0	64.2 ± 2.8	74.1 ± 2.8
6	25.4 ± 3.3	41.2 ± 3.7	44.5 ± 3.5	55.2 ± 4.0	64.2 ± 2.9	73.9 ± 3.0	65.8 ± 2.3	75.4 ± 2.4
7	28.8 ± 3.3	44.3 ± 4.1	46.2 ± 3.4	58.9 ± 3.8	65.6 ± 2.7	75.2 ± 2.9	65.9 ± 2.2	75.5 ± 2.4
8	29.5 ± 3.4	46.2 ± 3.9	44.2 ± 3.5	56.8 ± 4.0	66.4 ± 2.7	75.4 ± 3.2	67.3 ± 2.1	76.8 ± 2.1
9	29.7 ± 3.5	46.3 ± 4.0	43.6 ± 3.4	55.0 ± 4.0	66.5 ± 2.5	**75.5 ± 2.6**	67.7 ± 2.0	77.0 ± 2.1
10	34.1 ± 3.8	48.7 ± 4.0	44.6 ± 3.5	57.0 ± 3.8	**66.6 ± 2.5**	75.3 ± 2.5	68.0 ± 2.0	77.1 ± 2.0
11	35.0 ± 4.0	49.0 ± 3.9	44.3 ± 3.4	56.6 ± 3.5	63.8 ± 2.5	74.7 ± 2.5	67.7 ± 2.0	76.9 ± 1.9
12	37.9 ± 3.8	50.4 ± 4.0	46.1 ± 3.5	58.2 ± 3.4	64.0 ± 2.5	75.0 ± 2.6	67.7 ± 1.9	76.8 ± 1.8
13	39.7 ± 3.3	51.4 ± 3.4	45.4 ± 3.4	58.3 ± 3.6	64.1 ± 2.3	74.8 ± 2.5	**68.3 ± 2.7**	**77.2 ± 2.1**
14	39.9 ± 3.4	51.7 ± 3.7	50.0 ± 3.4	61.3 ± 3.7	64.3 ± 2.4	74.7 ± 2.4	62.1 ± 1.9	74.2 ± 2.5
15	41.7 ± 3.2	53.5 ± 3.5	54.7 ± 3.6	67.5 ± 3.4	64.3 ± 2.5	74.7 ± 2.4	61.9 ± 2.1	74.0 ± 2.3
16	44.8 ± 3.2	57.5 ± 3.7	56.1 ± 3.7	67.6 ± 3.4	64.4 ± 2.6	75.0 ± 2.4	61.3 ± 2.8	73.3 ± 2.9
17	49.7 ± 3.7	61.4 ± 3.7	56.0 ± 4.1	68.4 ± 3.5	64.5 ± 2.6	75.1 ± 2.6	61.3 ± 2.8	73.3 ± 2.8
18	50.4 ± 3.3	62.2 ± 3.8	55.9 ± 4.1	68.2 ± 3.5	64.3 ± 2.6	74.9 ± 2.9	61.3 ± 2.8	73.4 ± 2.8
19	52.3 ± 2.8	64.2 ± 3.2	55.2 ± 4.0	68.4 ± 3.3	64.1 ± 2.4	74.9 ± 2.9	61.4 ± 2.7	73.7 ± 2.9
20	53.8 ± 2.8	65.8 ± 3.4	57.4 ± 3.7	69.8 ± 3.0	63.5 ± 2.8	74.3 ± 2.9	61.2 ± 2.8	73.5 ± 2.9
21	56.3 ± 2.5	67.0 ± 3.2	60.5 ± 4.1	70.8 ± 3.4	62.8 ± 2.9	74.1 ± 2.9	61.2 ± 2.8	73.3 ± 2.9
22	60.1 ± 2.6	71.3 ± 3.0	61.5 ± 3.3	72.3 ± 2.8	62.6 ± 3.0	74.2 ± 2.9	61.2 ± 2.6	73.0 ± 3.0
23	62.0 ± 2.5	73.1 ± 2.9	**62.3 ± 3.0**	**73.5 ± 2.8**	62.5 ± 2.9	73.9 ± 2.6	60.3 ± 2.3	71.7 ± 2.1
24	**62.2 ± 2.7**	**73.4 ± 2.6**	62.2 ± 2.7	73.4 ± 2.6	62.2 ± 2.7	73.4 ± 2.6	60.4 ± 2.3	71.6 ± 1.9
25	60.1 ± 2.3	71.2 ± 2.0	60.1 ± 2.3	71.2 ± 2.0	60.1 ± 2.3	71.2 ± 2.0	60.1 ± 2.3	71.2 ± 2.0

Table 5.7: Classification accuracies for the 2NN classifier using the top feature set of dimension d as determined by the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	2NN Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	17.3 ± 2.7	32.7 ± 3.1	17.3 ± 2.7	32.7 ± 3.1	17.3 ± 2.7	32.7 ± 3.1	11.0 ± 1.6	25.2 ± 2.0
2	38.1 ± 2.7	47.4 ± 3.7	37.7 ± 2.8	50.0 ± 3.4	37.4 ± 3.0	50.0 ± 3.6	35.3 ± 2.1	45.9 ± 1.9
3	36.9 ± 3.2	49.3 ± 3.7	43.8 ± 2.9	54.8 ± 3.1	55.5 ± 3.0	64.7 ± 3.8	59.3 ± 2.5	69.3 ± 2.5
4	30.9 ± 2.7	44.9 ± 3.4	44.3 ± 3.0	54.5 ± 3.3	58.8 ± 3.1	67.8 ± 3.2	61.4 ± 1.8	71.3 ± 2.1
5	25.8 ± 2.6	41.3 ± 3.4	44.4 ± 3.4	56.8 ± 3.5	63.0 ± 3.2	71.7 ± 3.2	62.9 ± 1.8	73.1 ± 1.9
6	24.9 ± 2.7	40.9 ± 3.4	42.1 ± 3.5	53.8 ± 3.7	64.3 ± 3.6	73.0 ± 2.8	63.2 ± 1.9	73.3 ± 2.0
7	28.1 ± 2.6	41.6 ± 3.3	44.1 ± 3.5	56.4 ± 3.8	64.4 ± 3.1	73.0 ± 2.8	63.2 ± 1.9	73.3 ± 2.0
8	28.5 ± 2.6	43.6 ± 3.5	42.9 ± 3.4	55.4 ± 3.8	**66.1 ± 2.4**	**74.0 ± 2.6**	65.4 ± 1.3	75.1 ± 1.6
9	29.7 ± 2.9	43.3 ± 4.0	42.4 ± 3.7	54.3 ± 3.8	65.5 ± 2.2	73.9 ± 2.2	65.9 ± 1.6	75.5 ± 1.8
10	31.7 ± 3.0	45.0 ± 3.3	43.4 ± 3.3	56.1 ± 3.8	65.6 ± 2.1	**74.0 ± 2.5**	65.8 ± 1.4	75.5 ± 1.7
11	33.0 ± 3.3	46.4 ± 3.2	43.0 ± 3.3	56.1 ± 3.8	61.7 ± 2.5	72.3 ± 2.5	66.2 ± 1.5	**75.8 ± 1.7**
12	35.8 ± 3.3	48.0 ± 3.0	45.6 ± 3.5	57.9 ± 3.9	61.9 ± 3.3	72.6 ± 3.1	66.1 ± 1.6	75.7 ± 1.6
13	37.9 ± 3.1	49.9 ± 3.1	43.2 ± 3.3	56.1 ± 3.5	61.8 ± 2.7	72.3 ± 2.8	**66.4 ± 1.7**	75.0 ± 1.6
14	39.1 ± 3.5	51.0 ± 3.7	47.2 ± 3.5	59.2 ± 3.6	61.6 ± 2.0	71.9 ± 2.5	60.3 ± 1.2	71.5 ± 1.5
15	41.2 ± 3.8	52.1 ± 4.3	52.6 ± 3.7	64.5 ± 3.6	61.8 ± 2.7	72.1 ± 2.5	59.6 ± 1.6	70.5 ± 1.6
16	44.1 ± 3.5	55.0 ± 4.0	52.6 ± 3.2	64.8 ± 3.4	61.8 ± 2.9	72.3 ± 2.7	59.2 ± 2.3	70.1 ± 2.0
17	46.6 ± 3.6	57.7 ± 4.1	53.5 ± 2.8	65.4 ± 3.2	61.9 ± 3.0	72.2 ± 2.6	59.1 ± 2.3	70.0 ± 2.1
18	49.2 ± 3.6	59.6 ± 3.9	52.5 ± 2.9	64.6 ± 3.1	61.9 ± 3.0	72.3 ± 2.9	58.8 ± 2.3	69.7 ± 1.9
19	50.1 ± 3.5	60.8 ± 3.0	52.8 ± 2.9	64.5 ± 2.7	61.9 ± 3.0	72.1 ± 2.9	58.8 ± 2.2	69.7 ± 1.6
20	51.9 ± 3.3	62.9 ± 2.9	55.1 ± 2.6	66.8 ± 3.0	61.6 ± 3.0	71.9 ± 3.0	58.7 ± 2.1	69.6 ± 1.5
21	55.1 ± 2.9	64.7 ± 2.7	**60.3 ± 2.6**	68.4 ± 2.6	61.3 ± 2.9	71.6 ± 2.9	58.7 ± 2.3	69.5 ± 1.7
22	58.5 ± 2.9	67.7 ± 2.5	59.1 ± 2.8	68.6 ± 2.7	61.0 ± 2.9	71.2 ± 2.6	58.8 ± 1.7	69.6 ± 1.8
23	**60.4 ± 3.0**	70.2 ± 2.6	60.1 ± 3.1	**71.0 ± 2.5**	60.8 ± 2.8	70.6 ± 2.5	59.0 ± 1.9	69.3 ± 1.3
24	59.9 ± 2.6	**70.3 ± 2.5**	59.9 ± 2.6	70.3 ± 2.5	59.9 ± 2.6	70.3 ± 2.5	58.3 ± 2.1	68.9 ± 1.5
25	58.4 ± 1.9	68.8 ± 1.2	58.4 ± 1.9	68.8 ± 1.2	58.4 ± 1.9	68.8 ± 1.2	58.4 ± 1.9	68.8 ± 1.2

Table 5.8: Classification accuracies for the 3NN classifier using the top feature set of dimension d as determined by the lowest average tortuosity ($\overline{\tau}$), lowest average angle ($\overline{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	3NN Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity ($\overline{\tau}$)		Avg. Angle ($\overline{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	17.6 ± 2.5	30.9 ± 3.1	17.6 ± 2.5	30.9 ± 3.1	17.6 ± 2.5	30.9 ± 3.1	12.4 ± 1.7	25.9 ± 1.3
2	38.9 ± 3.0	47.2 ± 3.7	39.6 ± 3.2	50.4 ± 3.1	39.6 ± 3.3	50.4 ± 3.9	37.0 ± 1.9	47.0 ± 1.6
3	37.5 ± 2.9	48.1 ± 3.3	44.1 ± 3.6	55.3 ± 3.4	56.8 ± 3.3	66.4 ± 3.5	59.3 ± 1.4	69.7 ± 2.0
4	30.8 ± 2.9	43.8 ± 3.5	45.4 ± 3.0	55.7 ± 3.6	59.3 ± 3.3	68.9 ± 3.3	61.6 ± 1.9	71.4 ± 2.2
5	25.2 ± 3.0	40.0 ± 3.1	46.1 ± 3.4	57.3 ± 4.1	62.5 ± 3.1	71.3 ± 3.2	62.9 ± 2.2	72.6 ± 2.4
6	24.5 ± 3.0	39.3 ± 3.0	45.2 ± 3.1	55.5 ± 3.7	63.3 ± 3.2	72.3 ± 2.7	63.5 ± 2.0	73.3 ± 2.5
7	28.3 ± 3.1	41.6 ± 3.6	46.1 ± 3.4	57.8 ± 4.2	63.8 ± 2.7	72.4 ± 2.8	63.6 ± 2.0	73.3 ± 2.1
8	28.9 ± 2.6	42.9 ± 3.6	44.3 ± 3.3	56.3 ± 3.7	**64.7 ± 2.6**	72.6 ± 2.6	65.2 ± 2.5	74.2 ± 2.3
9	29.1 ± 2.8	42.0 ± 3.4	44.4 ± 3.3	55.1 ± 3.7	64.4 ± 2.5	**73.0 ± 2.6**	65.4 ± 2.4	74.1 ± 2.3
10	33.1 ± 3.0	44.9 ± 3.6	44.3 ± 3.1	56.8 ± 3.7	64.1 ± 2.6	72.6 ± 2.8	65.5 ± 2.4	74.3 ± 2.2
11	34.3 ± 3.4	46.3 ± 3.3	44.5 ± 3.1	56.8 ± 3.7	61.4 ± 2.5	71.9 ± 2.9	**66.0 ± 2.7**	**74.6 ± 2.5**
12	36.5 ± 3.2	48.4 ± 3.6	46.8 ± 3.0	58.9 ± 3.6	61.4 ± 2.4	72.0 ± 2.9	65.7 ± 2.9	74.4 ± 2.8
13	37.7 ± 3.5	48.8 ± 3.4	45.4 ± 3.3	57.5 ± 3.4	61.4 ± 2.6	72.0 ± 2.9	65.0 ± 1.5	73.7 ± 1.8
14	39.3 ± 3.0	50.4 ± 3.4	48.8 ± 3.8	60.2 ± 3.9	61.4 ± 2.5	72.0 ± 3.0	59.9 ± 1.5	70.9 ± 1.7
15	42.0 ± 3.2	52.2 ± 3.9	54.3 ± 3.8	65.4 ± 3.5	61.5 ± 2.5	72.0 ± 2.9	60.4 ± 1.5	70.8 ± 1.2
16	44.0 ± 3.2	53.1 ± 3.8	54.6 ± 3.8	65.6 ± 3.5	61.6 ± 2.7	72.1 ± 2.9	59.5 ± 1.8	70.4 ± 1.5
17	48.1 ± 3.1	58.1 ± 3.8	55.1 ± 3.6	65.6 ± 3.6	61.8 ± 2.6	72.2 ± 2.8	59.9 ± 1.8	70.9 ± 1.7
18	50.3 ± 3.3	60.7 ± 3.6	54.5 ± 2.9	65.4 ± 3.5	61.7 ± 2.8	72.0 ± 3.0	59.8 ± 1.9	70.7 ± 1.7
19	51.6 ± 3.3	61.9 ± 3.2	54.4 ± 3.0	65.3 ± 3.4	61.6 ± 2.8	72.2 ± 3.0	59.4 ± 1.8	70.3 ± 1.6
20	53.6 ± 3.3	64.2 ± 3.6	56.5 ± 3.3	67.6 ± 3.0	61.3 ± 2.7	72.1 ± 3.1	59.5 ± 1.7	70.4 ± 1.4
21	56.5 ± 3.1	65.4 ± 3.4	60.4 ± 3.3	69.4 ± 3.0	61.3 ± 2.6	71.8 ± 2.9	59.6 ± 1.6	70.4 ± 1.4
22	59.5 ± 2.7	69.3 ± 3.0	59.3 ± 3.2	69.2 ± 3.0	61.0 ± 2.7	71.8 ± 3.0	59.9 ± 1.6	70.6 ± 0.7
23	**61.0 ± 2.9**	**70.9 ± 3.3**	**60.6 ± 2.9**	**71.0 ± 3.1**	60.6 ± 2.3	71.0 ± 2.7	59.6 ± 1.1	69.2 ± 1.7
24	60.4 ± 2.6	70.7 ± 3.1	60.4 ± 2.6	70.7 ± 3.1	60.4 ± 2.6	70.7 ± 3.1	59.1 ± 0.9	69.1 ± 1.7
25	59.2 ± 1.0	69.3 ± 1.8	59.2 ± 1.0	69.3 ± 1.8	59.2 ± 1.0	69.3 ± 1.8	59.2 ± 1.0	69.3 ± 1.8

Table 5.9: Classification accuracies for the PCLDC classifier using the top feature set of dimension d as determined by the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	PCLDC Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	17.5 ± 2.0	33.5 ± 3.6	17.5 ± 2.0	33.5 ± 3.6	17.5 ± 2.0	33.5 ± 3.6	15.0 ± 1.3	29.8 ± 2.6
2	32.1 ± 3.6	42.5 ± 4.6	32.1 ± 2.7	44.9 ± 4.1	32.1 ± 3.0	44.9 ± 3.9	26.9 ± 2.6	36.7 ± 3.4
3	28.7 ± 2.8	43.8 ± 3.4	29.7 ± 3.5	40.9 ± 4.4	43.2 ± 2.8	53.3 ± 4.2	41.2 ± 2.6	56.7 ± 2.9
4	23.5 ± 2.8	42.5 ± 3.3	31.7 ± 3.8	44.9 ± 4.4	48.8 ± 3.1	57.7 ± 3.7	44.1 ± 2.1	58.0 ± 2.1
5	21.5 ± 2.6	37.5 ± 3.1	33.0 ± 3.8	47.5 ± 4.7	50.8 ± 3.2	60.4 ± 3.8	43.6 ± 1.3	56.7 ± 0.9
6	21.6 ± 2.5	37.8 ± 3.4	34.6 ± 4.1	48.3 ± 4.7	52.4 ± 3.1	63.1 ± 4.0	44.3 ± 2.0	58.0 ± 1.8
7	24.5 ± 2.8	41.8 ± 3.4	37.1 ± 4.0	50.0 ± 4.7	55.1 ± 3.3	66.4 ± 4.0	44.0 ± 1.2	58.2 ± 1.5
8	25.1 ± 3.2	42.0 ± 3.5	37.5 ± 3.8	50.3 ± 4.7	58.1 ± 3.5	68.4 ± 3.7	50.3 ± 1.7	64.1 ± 2.7
9	25.1 ± 3.5	42.0 ± 3.6	39.3 ± 3.8	53.2 ± 4.3	59.4 ± 3.5	68.9 ± 3.8	51.8 ± 2.5	64.8 ± 2.4
10	29.1 ± 3.5	44.0 ± 3.6	39.8 ± 3.6	53.3 ± 4.2	60.5 ± 3.5	71.0 ± 3.5	51.8 ± 2.5	64.8 ± 2.4
11	30.1 ± 3.2	45.2 ± 3.5	40.0 ± 3.8	53.3 ± 4.4	61.3 ± 3.7	71.5 ± 3.6	55.8 ± 2.3	66.8 ± 2.8
12	33.5 ± 3.1	44.8 ± 3.4	42.3 ± 3.9	55.4 ± 4.4	62.4 ± 3.6	72.3 ± 3.4	55.8 ± 2.3	66.8 ± 2.8
13	37.3 ± 3.2	49.3 ± 3.4	46.0 ± 4.4	60.3 ± 5.0	63.1 ± 3.4	73.1 ± 2.9	59.8 ± 1.6	69.4 ± 1.5
14	38.0 ± 3.5	49.4 ± 3.2	47.5 ± 4.1	61.5 ± 4.5	63.8 ± 3.2	73.3 ± 2.8	61.0 ± 2.8	70.6 ± 2.0
15	38.7 ± 3.5	50.2 ± 3.2	51.7 ± 3.9	65.1 ± 4.3	64.3 ± 3.3	73.5 ± 2.8	60.8 ± 2.9	70.4 ± 2.1
16	41.9 ± 3.7	53.9 ± 3.6	52.0 ± 4.0	65.4 ± 3.8	64.7 ± 3.3	74.1 ± 2.7	55.3 ± 2.1	65.6 ± 2.6
17	47.3 ± 3.9	57.4 ± 3.6	52.0 ± 4.0	65.4 ± 3.2	64.8 ± 3.3	74.1 ± 2.9	55.4 ± 2.0	65.8 ± 2.4
18	54.1 ± 3.9	62.6 ± 3.2	52.1 ± 3.9	65.5 ± 3.3	65.1 ± 3.5	74.6 ± 2.9	55.4 ± 2.0	65.8 ± 2.4
19	54.1 ± 3.9	62.6 ± 3.0	57.4 ± 3.7	69.0 ± 3.1	65.1 ± 3.6	74.6 ± 2.9	59.5 ± 2.3	69.7 ± 2.1
20	59.3 ± 3.2	67.2 ± 2.9	58.1 ± 4.5	69.9 ± 3.1	65.2 ± 3.7	74.7 ± 3.2	59.5 ± 2.3	69.7 ± 2.1
21	62.0 ± 3.7	69.8 ± 2.8	63.0 ± 3.9	71.3 ± 2.8	**65.3 ± 3.8**	**74.8 ± 2.9**	59.4 ± 3.0	69.2 ± 2.8
22	64.2 ± 3.5	73.0 ± 3.1	**65.1 ± 3.5**	73.3 ± 2.7	65.1 ± 3.6	74.5 ± 3.1	63.4 ± 2.9	72.3 ± 2.7
23	**65.1 ± 3.4**	73.4 ± 2.6	**65.1 ± 3.7**	**73.8 ± 2.8**	64.9 ± 3.2	74.4 ± 2.9	61.4 ± 2.6	71.0 ± 1.9
24	65.0 ± 3.5	**73.8 ± 2.8**	65.0 ± 3.5	**73.8 ± 2.8**	65.0 ± 3.5	73.8 ± 2.8	64.1 ± 2.9	73.3 ± 2.4
25	64.3 ± 2.5	73.5 ± 2.4	64.3 ± 2.5	73.5 ± 2.4	64.3 ± 2.5	73.5 ± 2.4	**64.3 ± 2.5**	**73.5 ± 2.4**

Table 5.10: Classification accuracies for the LOGLC classifier using the top feature set of dimension d as determined by the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\|\mathbf{v}\|$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray. The ($\cdots$) symbol represents a classification algorithm that failed to converge.

	LOGLC Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\|\mathbf{v}\|$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	17.1 ± 2.2	32.9 ± 3.6	17.1 ± 2.2	32.9 ± 3.6	17.1 ± 2.2	32.9 ± 3.6	14.5 ± 1.3	30.7 ± 2.2
2	32.5 ± 2.6	42.5 ± 3.8	36.0 ± 2.9	48.4 ± 4.1	35.7 ± 2.9	48.4 ± 4.1	30.1 ± 2.2	41.2 ± 3.7
3	39.1 ± 2.7	47.5 ± 3.6	37.1 ± 13.9	47.7 ± 11.5	45.0 ± 2.8	57.2 ± 3.8	49.1 ± 2.9	59.3 ± 2.9
4	26.4 ± 2.7	44.1 ± 3.2	38.3 ± 11.2	48.1 ± 9.4	50.5 ± 2.7	61.6 ± 3.3	52.9 ± 3.5	61.7 ± 3.0
5	23.0 ± 3.0	40.2 ± 3.6	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
6	22.8 ± 2.9	39.4 ± 3.3	43.2 ± 12.9	54.1 ± 10.6	57.1 ± 3.0	67.1 ± 3.5	54.1 ± 8.3	63.6 ± 7.6
7	25.8 ± 2.6	42.8 ± 3.5	39.8 ± 13.2	50.4 ± 11.4	60.9 ± 15.2	69.7 ± 12.9	42.6 ± 13.6	52.8 ± 12.3
8	27.0 ± 3.0	43.2 ± 3.4	40.3 ± 14.1	51.3 ± 12.1	62.2 ± 19.2	71.6 ± 16.2	61.6 ± 1.9	70.6 ± 2.3
9	27.5 ± 3.2	43.2 ± 3.5	40.7 ± 13.8	51.0 ± 12.3	62.8 ± 18.4	72.8 ± 15.2	58.1 ± 15.7	65.9 ± 16.8
10	39.3 ± 3.2	48.3 ± 3.8	43.5 ± 13.7	53.3 ± 12.2	64.0 ± 20.4	73.6 ± 16.6	58.1 ± 15.7	65.9 ± 16.8
11	39.6 ± 3.8	48.9 ± 4.2	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
12	40.8 ± 7.4	49.8 ± 6.0	44.5 ± 16.4	54.3 ± 14.8	66.3 ± 25.4	75.8 ± 24.8	49.4 ± 20.5	60.3 ± 16.9
13	40.8 ± 10.0	49.9 ± 8.5	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
14	42.7 ± 10.6	53.6 ± 9.1	51.9 ± 17.8	61.8 ± 16.4	64.2 ± 24.6	73.0 ± 22.9	61.9 ± 10.8	72.7 ± 7.5
15	45.5 ± 14.2	54.7 ± 13.4	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
16	43.5 ± 14.3	52.8 ± 13.0	58.6 ± 22.2	68.1 ± 19.7	63.3 ± 24.7	72.9 ± 22.9	39.7 ± 15.5	52.5 ± 14.5
17	43.5 ± 16.3	52.9 ± 14.3	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
18	50.7 ± 17.9	60.5 ± 15.4	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
19	50.7 ± 18.4	60.5 ± 15.3	59.0 ± 18.5	68.3 ± 17.6	68.6 ± 22.2	76.6 ± 20.1	46.9 ± 18.0	58.7 ± 17.5
20	56.3 ± 19.7	66.2 ± 18.3	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
21	60.3 ± 21.4	68.8 ± 18.9	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
22	66.1 ± 24.0	74.0 ± 23.7	68.4 ± 20.3	76.1 ± 20.3	$\cdots$	$\cdots$	$\cdots$	$\cdots$
23	64.2 ± 21.2	72.6 ± 18.7	64.7 ± 22.5	73.4 ± 21.8	68.6 ± 20.8	76.7 ± 21.8	56.4 ± 17.3	66.5 ± 14.5
24	60.9 ± 22.5	70.4 ± 21.6	60.9 ± 22.5	70.4 ± 21.6	60.9 ± 22.5	70.4 ± 21.6	54.7 ± 20.1	62.6 ± 21.6
25	56.0 ± 14.3	65.2 ± 13.5	56.0 ± 14.3	65.2 ± 13.5	56.0 ± 14.3	65.2 ± 13.5	56.0 ± 14.3	65.2 ± 13.5

Table 5.11: Classification accuracies for the FISHERC classifier using the top feature set of dimension d as determined by the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	FISHERC Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	17.6 ± 2.2	32.2 ± 3.1	17.6 ± 2.2	32.2 ± 3.1	17.6 ± 2.2	32.2 ± 3.1	13.1 ± 1.3	29.8 ± 2.2
2	28.8 ± 2.5	41.1 ± 3.9	28.8 ± 2.6	42.7 ± 4.0	28.8 ± 2.5	42.7 ± 3.5	22.6 ± 1.6	34.4 ± 2.3
3	25.1 ± 2.9	41.0 ± 3.6	26.3 ± 2.6	38.9 ± 2.9	41.2 ± 3.2	53.2 ± 3.5	37.6 ± 2.4	51.7 ± 2.1
4	23.1 ± 2.8	39.0 ± 3.4	30.4 ± 3.2	42.8 ± 3.6	44.5 ± 3.0	57.1 ± 3.6	39.2 ± 2.4	51.7 ± 2.4
5	22.0 ± 2.7	37.8 ± 3.0	32.2 ± 3.9	45.1 ± 3.6	46.7 ± 3.1	60.1 ± 3.2	41.8 ± 2.0	55.0 ± 2.0
6	22.0 ± 2.8	37.8 ± 3.3	33.9 ± 3.5	47.9 ± 3.7	49.1 ± 3.5	61.7 ± 3.7	44.6 ± 1.5	57.7 ± 1.6
7	23.5 ± 2.8	39.7 ± 3.0	35.7 ± 3.1	49.8 ± 3.7	51.1 ± 3.5	64.1 ± 4.3	42.8 ± 1.6	56.7 ± 1.5
8	23.7 ± 3.2	39.6 ± 3.7	35.7 ± 3.1	50.3 ± 3.5	52.7 ± 3.1	66.3 ± 4.2	46.8 ± 2.6	62.0 ± 3.5
9	23.6 ± 3.2	39.6 ± 3.6	36.2 ± 3.3	50.7 ± 3.5	54.6 ± 3.3	67.5 ± 3.9	47.4 ± 2.6	62.2 ± 3.1
10	26.5 ± 3.2	40.2 ± 3.6	37.4 ± 3.4	51.9 ± 3.8	56.4 ± 3.1	68.9 ± 3.5	47.4 ± 2.6	62.2 ± 3.1
11	27.3 ± 3.0	41.4 ± 3.4	39.0 ± 3.2	52.2 ± 3.8	57.0 ± 3.2	68.9 ± 3.8	50.6 ± 2.9	65.1 ± 2.9
12	32.1 ± 3.3	44.3 ± 3.5	40.0 ± 3.3	53.8 ± 3.8	58.4 ± 3.1	70.7 ± 3.6	50.6 ± 2.9	65.1 ± 2.9
13	34.6 ± 3.3	47.4 ± 3.5	43.7 ± 3.3	58.4 ± 3.8	59.6 ± 2.8	71.4 ± 2.8	54.7 ± 2.4	66.5 ± 2.6
14	35.5 ± 3.5	48.4 ± 3.5	45.9 ± 3.3	59.4 ± 3.6	60.7 ± 2.8	72.7 ± 2.8	56.8 ± 2.0	68.4 ± 1.7
15	36.4 ± 3.3	49.4 ± 3.2	45.8 ± 3.9	60.6 ± 3.6	60.7 ± 3.0	72.7 ± 2.4	56.2 ± 2.1	67.8 ± 1.9
16	39.5 ± 3.1	52.3 ± 3.3	47.8 ± 4.0	60.9 ± 3.5	60.7 ± 3.0	$\mathbf{72.8 \pm 2.5}$	50.9 ± 2.2	62.1 ± 2.5
17	43.0 ± 3.2	54.7 ± 3.3	48.0 ± 4.0	61.0 ± 3.1	60.7 ± 3.2	$\mathbf{72.8 \pm 2.7}$	51.0 ± 2.4	61.7 ± 1.9
18	49.2 ± 3.4	59.6 ± 3.3	47.8 ± 3.9	60.9 ± 3.1	$\mathbf{60.9 \pm 3.3}$	72.7 ± 2.6	51.0 ± 2.4	61.7 ± 1.9
19	49.2 ± 3.3	59.6 ± 3.4	51.6 ± 3.7	63.5 ± 3.1	60.7 ± 3.4	72.4 ± 2.4	54.8 ± 1.8	66.5 ± 1.8
20	52.7 ± 3.4	62.6 ± 3.2	52.5 ± 3.8	64.2 ± 3.0	60.6 ± 3.3	71.7 ± 2.4	54.8 ± 1.8	66.5 ± 1.8
21	55.7 ± 3.2	64.9 ± 3.0	56.1 ± 3.6	66.4 ± 3.2	60.4 ± 3.4	71.1 ± 2.3	54.6 ± 1.8	65.9 ± 2.0
22	58.0 ± 3.5	67.9 ± 2.7	58.7 ± 3.4	68.4 ± 2.7	60.1 ± 3.3	70.9 ± 2.2	$\mathbf{59.2 \pm 2.6}$	$\mathbf{69.8 \pm 2.0}$
23	58.4 ± 3.6	69.3 ± 2.7	$\mathbf{59.1 \pm 3.9}$	$\mathbf{69.6 \pm 2.7}$	60.0 ± 3.4	70.5 ± 2.1	56.4 ± 2.0	66.9 ± 1.8
24	$\mathbf{59.1 \pm 3.1}$	$\mathbf{69.6 \pm 2.0}$	$\mathbf{59.1 \pm 3.1}$	$\mathbf{69.6 \pm 2.0}$	59.1 ± 3.1	69.6 ± 2.0	58.6 ± 3.0	68.9 ± 1.5
25	58.4 ± 2.9	68.7 ± 1.4	58.4 ± 2.9	68.7 ± 1.4	58.4 ± 2.9	68.7 ± 1.4	58.4 ± 2.9	68.7 ± 1.4

Table 5.12: Classification accuracies for the NMC classifier using the top feature set of dimension d as determined by the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential floating backward search (SFBS). Results are presented as the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling technique for both an exact classification label (Exact), as well as the accuracy to within ±1 flaw depth (±1 Depth). The highest classification accuracy per column is highlighted in gray.

	NMC Classification Accuracy ($\mu \pm \sigma$) [%]							
	Tortuosity (τ)		Avg. Angle ($\bar{\theta}$)		Centroid Dist. ($\overline{\|\mathbf{v}\|}$)		SFBS - 1NN	
Dimension	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth	Exact	±1 Depth
1	18.2 ± 2.3	31.6 ± 3.2	18.2 ± 2.3	31.6 ± 3.2	18.2 ± 2.3	31.6 ± 3.2	16.7 ± 1.8	31.0 ± 2.8
2	25.3 ± 2.8	40.1 ± 3.9	25.8 ± 3.5	38.3 ± 4.4	26.7 ± 3.5	40.1 ± 4.4	25.0 ± 2.7	35.8 ± 3.9
3	24.8 ± 2.7	41.6 ± 3.7	26.5 ± 3.4	37.7 ± 4.5	31.5 ± 3.3	43.9 ± 3.6	28.1 ± 1.9	40.2 ± 2.7
4	23.3 ± 2.9	39.3 ± 3.7	25.0 ± 2.8	36.5 ± 3.8	33.7 ± 3.0	45.0 ± 3.6	28.0 ± 2.7	40.1 ± 2.8
5	20.6 ± 2.6	34.6 ± 3.6	25.5 ± 2.7	36.9 ± 3.9	34.8 ± 2.7	46.3 ± 3.7	27.5 ± 2.1	39.7 ± 1.6
6	20.7 ± 2.5	34.7 ± 3.6	24.5 ± 2.4	36.7 ± 4.2	35.2 ± 2.8	46.8 ± 3.5	27.4 ± 2.3	39.6 ± 1.5
7	23.3 ± 2.5	39.3 ± 3.4	24.4 ± 2.6	36.4 ± 4.3	35.7 ± 2.7	**47.3 ± 3.3**	27.4 ± 2.4	39.4 ± 1.9
8	23.7 ± 2.5	39.6 ± 3.6	24.3 ± 2.7	36.5 ± 4.3	**35.9 ± 2.7**	47.2 ± 3.1	28.8 ± 3.2	41.6 ± 3.0
9	24.1 ± 2.4	41.2 ± 3.1	24.7 ± 2.8	36.6 ± 4.1	35.7 ± 2.7	46.5 ± 3.2	28.4 ± 2.8	41.2 ± 2.9
10	25.8 ± 2.1	41.1 ± 2.8	25.5 ± 2.6	36.7 ± 4.1	35.4 ± 2.8	46.6 ± 3.3	28.3 ± 3.0	40.9 ± 3.2
11	26.1 ± 2.3	42.5 ± 2.8	26.0 ± 2.7	37.1 ± 3.8	35.4 ± 2.8	46.5 ± 3.0	28.8 ± 3.1	41.4 ± 3.2
12	26.7 ± 2.6	43.6 ± 2.6	26.1 ± 3.0	37.5 ± 3.7	35.1 ± 2.6	46.2 ± 2.8	29.1 ± 2.7	41.6 ± 2.9
13	27.8 ± 2.6	43.6 ± 2.7	26.0 ± 3.0	37.6 ± 4.0	35.2 ± 2.7	46.2 ± 2.9	29.8 ± 1.8	39.9 ± 2.7
14	29.1 ± 2.5	44.0 ± 2.7	26.1 ± 3.3	37.5 ± 4.0	35.4 ± 2.8	46.1 ± 2.9	34.9 ± 2.6	45.8 ± 2.7
15	28.8 ± 2.7	43.5 ± 2.6	28.2 ± 3.5	40.2 ± 3.8	35.5 ± 2.6	46.0 ± 2.8	34.9 ± 2.5	45.8 ± 2.6
16	31.3 ± 3.4	45.2 ± 3.6	28.2 ± 3.6	40.4 ± 3.7	35.5 ± 2.7	46.2 ± 2.8	34.9 ± 2.4	45.7 ± 2.6
17	31.7 ± 3.5	45.6 ± 3.4	28.3 ± 3.6	40.4 ± 3.7	35.4 ± 2.8	45.9 ± 2.9	35.0 ± 2.3	45.9 ± 2.5
18	32.5 ± 3.4	45.2 ± 3.5	28.1 ± 3.5	40.4 ± 3.9	35.5 ± 2.7	45.9 ± 2.8	35.2 ± 2.4	**46.0 ± 2.7**
19	33.2 ± 3.3	43.0 ± 3.4	28.0 ± 3.2	40.6 ± 3.9	35.5 ± 2.9	45.9 ± 3.0	35.4 ± 2.5	45.9 ± 2.5
20	33.5 ± 2.9	43.1 ± 3.2	27.9 ± 3.3	40.5 ± 3.8	35.6 ± 2.9	45.9 ± 2.9	35.5 ± 2.5	45.8 ± 2.6
21	33.6 ± 2.9	43.0 ± 3.3	29.7 ± 3.1	42.2 ± 3.5	35.6 ± 2.8	45.9 ± 2.9	**35.6 ± 2.6**	45.9 ± 2.8
22	35.7 ± 3.0	45.8 ± 3.2	34.2 ± 2.9	45.3 ± 3.4	35.6 ± 2.8	45.9 ± 2.9	34.9 ± 2.8	45.3 ± 2.9
23	**36.0 ± 2.7**	46.5 ± 3.3	**35.5 ± 3.5**	**47.4 ± 3.5**	35.5 ± 2.8	46.2 ± 3.0	35.0 ± 2.5	45.5 ± 2.7
24	35.5 ± 3.1	**46.7 ± 3.2**	**35.5 ± 3.1**	46.7 ± 3.2	35.5 ± 3.1	46.7 ± 3.2	35.3 ± 2.7	45.7 ± 2.8
25	35.3 ± 2.7	45.7 ± 2.8	35.3 ± 2.7	45.7 ± 2.8	35.3 ± 2.7	45.7 ± 2.8	35.3 ± 2.7	45.7 ± 2.8

The logistic linear classifier is an inherently binary classifier that implements logistic regression to find a linear classifier between two classes. It is extended to our multi-class scenario by computing j individual classifiers between each of the j classes in the data set and the remaining $j - 1$ classes (one-vs-all). We observe convergence issues here for many of the class distributions using the logistic linear classifier, a well-documented occasional problem with logistic regression [156]. Since we have included the logistic classifier for completeness and had no reason to depend on the LOGLC results specifically, we do not explore alternate methods for detection of false convergence in linear regression methods in this work. We will no longer explore the logistic linear classifier for this application because of this.

The quadratic discriminant classifier (QDC) returned the overall highest classification accuracies, reaching 75.6% correct classification (Exact) and 80.7% classification within ±1 depths (±1 Depth) with a 16-dimensional feature set identified using the top centroid distance metric ($\overline{\|\mathbf{v}\|}$). The entire set of classifiers were all reasonably consistent in their performance, with accuracy standard deviation (σ) values all below 5%, ranging from 1.7% to 4.9%. The three k-NN classifiers show clear signs of the curse of dimensionality in Tables 5.6 - 5.8, where the classification accuracy peaks around $d = 10$ features and then drops off as the feature size increase. The other classifiers seem to not lose accuracy with increased dimension as quickly, although they do not improve much either. This is possibly due to the fact that the original feature reduction from 78 to 25 features was done using 1-NN accuracy as a criterion, meaning that the starting 25-feature set is not necessarily optimal for the alternate classifiers. There may exist alternate feature space subsets of the original 78-dimensional feature space that would result in higher classification accuracies for these classifiers. As previously discussed, identifying these without performing a brute-force search may not be possible. Rather, we are exploring the potential for simple geometric measurements to identify the more optimal feature subsets given an initial starting set (the 25-feature SFBS results here).

A summary of these results can be found in Table 5.13, where only the top classification accuracy among all feature subspace dimensions is presented for each classifier and feature selection metric combination. This table considers only the accuracy within ±1 flaw depth (±1 Depth), and each entry corresponds to the highlighted values from Tables 5.4 - 5.12.

Table 5.13: Classification accuracy for the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between class centroids ($\overline{\|\mathbf{v}\|}$) feature selection metrics, along with the 'optimal' feature space classification accuracy as determined by the sequential backward floating backward search (SFBS) for each classifier. Results are presented in terms of the mean (μ) plus or minus the standard deviation (σ) of the classifier accuracy under ten repetitions of a bagging resampling method, where randomly selected subsets of both the testing (T) and training (R) sets are used to test and train each classifier, respectively. The corresponding feature space dimension (d) is provided in parenthesis.

	Classification Accuracy $\mu \pm \sigma$ (d) [% within ±1 flaw depth]			
Classifier	τ	$\bar{\theta}$	$\overline{\|\mathbf{v}\|}$	SFBS
LDC	73.8 ± 2.8 (24)	73.8 ± 2.8 (23)	74.8 ± 2.9 (21)	73.5 ± 2.4 (25)
QDC	80.6 ± 4.6 (23)	80.6 ± 2.9 (22)	80.7 ± 4.9 (16)	79.4 ± 4.0 (15)
1NN	73.4 ± 2.6 (24)	73.5 ± 2.8 (23)	75.5 ± 2.6 (9)	77.2 ± 2.1 (13)
2NN	70.3 ± 2.5 (24)	71.0 ± 2.5 (23)	74.0 ± 2.6 (8)	75.8 ± 1.7 (11)
3NN	70.9 ± 3.3 (23)	71.0 ± 3.1 (23)	73.0 ± 2.6 (9)	74.6 ± 2.5 (11)
PDLCD	73.8 ± 2.8 (24)	73.8 ± 2.8 (23)	74.8 ± 2.9 (21)	73.5 ± 2.4 (25)
FISHERC	69.6 ± 2.0 (24)	69.6 ± 2.7 (23)	72.8 ± 2.5 (16)	69.8 ± 2.0 (22)
NMC	46.7 ± 3.2 (24)	47.4 ± 3.5 (23)	47.3 ± 3.3 (7)	46.0 ± 2.7 (18)

Although the No Free Lunch theorem states that there is no benefit in "simple" classifiers over "complex" ones, the principle of *satisficing* creating an adequate though possibly non-optimal solution is accepted and widely practiced in pattern recognition [1]. To that extent, the 1NN classifier is both simple and provides reasonable accuracy compared to the other classifiers. Since computational complexity and speed are usually restrictions, then this could be the most appropriate classification technique to use. Otherwise, the quadratic discriminant classifier (QDC) returns the highest general classification accuracies here.

5.8 Independent Sample Validation

The above analysis is useful to explore how well a classifier can be expected to perform given a limited data set. We next seek to test the resulting classification abilities when *new* data are used to test the classifiers. We use a training set composed of the full 14-classes outlined in Section 5.3. Once trained, we test each classifier with a series of independent plate samples, each with a milled flaw of varying depth and shape. The classifier algorithms considered are those discussed in Section 5.6.

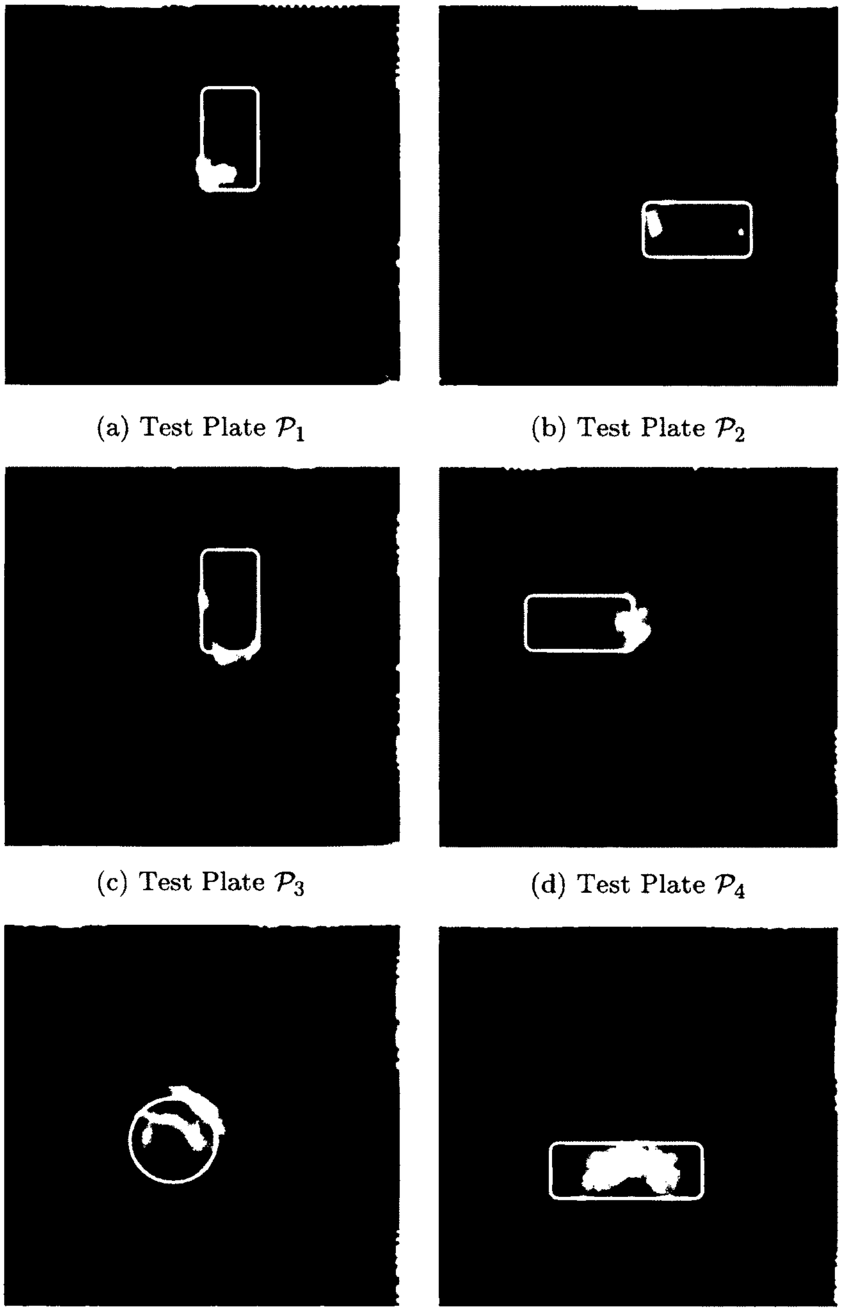
We introduce six new testing plates $\mathcal{P}_x$ for $x = 1, \ldots, 6$ [148] with flat-bottom hole flaws, and collect a double-crosshole scan data set from each. These plates are all identical in material and size as the plate used in the training data set. Plates $\mathcal{P}_1$, $\mathcal{P}_2$, $\mathcal{P}_3$, and $\mathcal{P}_4$ all contain flaws that are rectangular in shape and are approximately the same size as the flaw used to train the classifiers, although not necessarily in the same position or orientation as that used to train the classifiers. Plate $\mathcal{P}_5$ contains a flaw that is not rectangular, but circular in shape. Plate $\mathcal{P}_6$ is physically the same plate as that used in the training data set, however it is here scanned by a different person many months after the initial training data set was collected, making the waveforms distinct. Details of these flaws are summarized in Table 5.14. Each plate's double-crosshole scan is tomographically reconstructed, results of which can be seen in Figure 5.14. The reconstruction image over/under color intensity threshold described in Section 5.4 is applied to each new reconstruction, automatically identifying the subset of raypaths which pass through each of these flaws. It should be noted again that this process requires no user input other than the physical scanning of the plate, as the reconstruction and image thresholding parameters have already been established in Section 5.4.

Table 5.14: Testing plates used in the independent sample test, $\mathcal{P}_1, \ldots, \mathcal{P}_6$. The closest corresponding class from the training set is provided for each plate for reference, where a range indicates the testing flaw severity lies between those of the training set.

Plate Number	Average Flaw Depth	Corresponding Training Class	Flat-Bottom Flaw Shape
$\mathcal{P}_1$	10 %	D_2-D_3	Rectangular (3 × 5 cm)
$\mathcal{P}_2$	10 %	D_2-D_3	Rectangular (5 × 3 cm)
$\mathcal{P}_3$	45 %	D_8-D_9	Rectangular (3 × 5 cm)
$\mathcal{P}_4$	45 %	D_8-D_9	Rectangular (5 × 3 cm)
$\mathcal{P}_5$	60 %	D_{10}-D_{11}	Circular (4 cm radius)
$\mathcal{P}_6$	100 %	D_{14}	Rectangular (2.9 × 7.6 cm)

Features are extracted from waveforms identified as crossing a flaw for each test plate. The statistical, Mellin transform, and WPD features are extracted as before. For the DWFP features used previously in training the classifier, an intermediate feature selection step was performed that identified the times and DWFP measurements for which the largest inter-class separation was present. These same times and DWFP measurements are used to collect the DWFP features from the new plates, remaining consistent with the previous analysis as well.

Our training set is composed of the 14 classes $D_1, \ldots, D_{14}$ to include a generalization of the varying degrees of material loss possible in a plate. As previously discussed, it is unreasonable to assume a training set can contain a comprehensive set of flaw depths, shapes, sizes, etc. and therefore should not be assumed that a testing plate will necessarily fall into one of the existing training classes. While several of new these testing plates have flaws similar in shape to that used for training, their flaw depths fall inbetween those in the training set. This is a result of the inter-class continuity previously discussed. For example, testing plate $\mathcal{P}_3$'s flaw severity of 45% material loss lies between classes D_8 and D_9 (43% and 51% material loss, respectively) of the training data. Predicted labels of ω_8 or ω_9 would therefore be



(a) Test Plate $\mathcal{P}_1$

(b) Test Plate $\mathcal{P}_2$

(c) Test Plate $\mathcal{P}_3$

(d) Test Plate $\mathcal{P}_4$

(e) Test Plate $\mathcal{P}_5$

(f) Test Plate $\mathcal{P}_6$

Figure 5.14: Tomographic reconstructions of testing plates $\mathcal{P}_1$ (a) - $\mathcal{P}_6$ (f). Known flaw locations are indicated by white outlines.

considered correct labels, while ω_7 or ω_{10} are considered to be within ±1 flaw depth.

The features spaces used here consist of an average of the top ten 'best' subspaces of each dimension $d = 1, \ldots, 25$ as determined by the three geometric feature selection metrics. By best, we mean the feature spaces corresponding to the ten lowest average tortuosity (τ), ten lowest average angle ($\bar{\theta}$), and ten highest average distance between class centroid ($\overline{\|\mathbf{v}\|}$) class distributions. Each set of ten feature spaces were individually used to train/test the classifier, and the resulting accuracies were averaged. While this averaging is expected to reduce the accuracies presented here (since the feature spaces are already sorted based on how well we expect them to perform), that assumption relies on the accuracy of our intuition behind the geometric measures. We have already described how each metric could (individually) identify non-optimal feature subspaces, so there is no reason to expect the 'best' feature space identified by each metric to identify the single best feature subspace of that dimension. We therefore average the ten best feature subspaces for each dimension to produce a general accuracy result that can be used to compare the performance of each feature selection metrics. For the SFBS feature selection results, we use only the single feature space of dimension d that has been identifed by the SFBS algorithm.

We again expect highest accuracies to be around dimension $d = 10$ due to the curse of dimensionality rule-of-thumb. A classification configuration consisting of both an individual classifier and a feature set of dimension d would be considered ideal here if it returned 100% accuracy for each of the six test plates $\mathcal{P}_1$ - $\mathcal{P}_6$. This is not a requirement for an acceptable result, nor is it a realistic result since 100% accuracy would indicate this to be a much simpler problem that we believe it to be. Rather, only a majority vote (or some other voting method) needs to by implemented to identify a correct class from each test set.

Table 5.15 provides a summary of these results, where the classifier configurations sorted by the highest average accuracy (within ±1 flaw depth) across all six testing plates are shown. Results are presented for the lowest average tortuosity (τ), lowest average angle ($\bar{\theta}$), and highest average distance between centroid ($\overline{\|\mathbf{v}\|}$) metric feature selection routines as well as the sequential floating backward search (SFBS) feature selection routine.

Several general trends can be seen in these results. For the geometric feature selection metrics, the average angle between centroids ($\bar{\theta}$) and the tortuosity (τ) provide lower average accuracy values across all six testing plates than the average distance between centroids ($\overline{\|\mathbf{v}\|}$). These peak at accuracies of 38.0%, 38.4%, and 44.3%, respectively. This is in agreement with earlier results in Section 5.7, where both of these metrics performed worse than the average centroid distance metric. Since this generally lower performance has translated to the validation data sets, we will no longer consider either the tortuosity or average angle metrics as optimal measures of classifier performance, and do not explore their results further.

The highest six-plate average classification accuracy comes from the 1NN SFBS results, where 51.7% of the testing waveforms are correctly identified to be within ±1 flaw depth of the actual testing flaw. Again, this result makes sense considering that the SFBS routine uses the maximum 1NN accuracy as a criterion function. It can be seen, however, that these results almost all favor high classification accuracies for plates $\mathcal{P}_1$ - $\mathcal{P}_5$ while returning very low accuracy values for plate $\mathcal{P}_6$. This is an unexpected result, since plate $\mathcal{P}_6$ corresponds directly to class D_{14} of the training set. Scanning with individual contact transducers is prone to variation in the signal, however, from unintentional changes in pressure and coupling of the guided wave energy in and out of the plate. This is a known drawback of using a simulated double-crosshole scanning technique [45]. In service, using the full array of transducers we mimic here would eliminate inconsistencies in coupling and pressure, resulting in more

Table 5.15: Classifier configurations sorted by the highest average accuracy (within ±1 flaw depth) across all six testing plates for feature subspaces of dimension $d = 1, \ldots, 25$. Results are presented for the lowest average tortuosity ($\overline{\tau}$), lowest average angle ($\bar{\theta}$), and highest average distance between centroid ($\overline{\|\mathbf{v}\|}$) metric feature selection routines as well as the sequential floating backward search (SFBS) feature selection routine. For each classifier, the feature subset dimension corresponding to the highest average accuracy is presented as d.

	Dim.	Feature	Accuracy [% within ±1 flaw depth]						
Classifier	d	Selection	$\mathcal{P}_1$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_2$	$\mathcal{P}_6$	Avg.
LDC	2	τ	78.6	77.0	22.0	17.9	22.0	6.1	37.3
QDC	16	τ	90.6	94.0	14.4	12.8	0.2	14.8	37.8
1NN	20	τ	7.9	53.9	50.2	30.7	72.6	1.9	36.2
2NN	19	τ	29.4	81.4	35.9	24.3	56.0	0.9	38.0
3NN	2	τ	22.2	15.6	68.3	54.3	42.7	6.4	34.9
PCLDC	2	τ	78.6	77.0	22.0	17.9	22.0	6.1	37.3
FISHERC	2	τ	80.1	78.7	19.3	13.9	23.5	7.4	37.1
NMC	2	τ	87.4	89.2	10.0	14.2	14.2	12.5	37.9
LDC	25	θ	9.0	28.0	57.0	63.0	52.0	2.0	35.2
QDC	19	$\bar{\theta}$	94.8	93.2	22.4	17.1	0.0	0.9	38.1
1NN	13	$\bar{\theta}$	89.9	90.6	9.4	7.3	25.9	7.4	38.4
2NN	21	$\bar{\theta}$	22.7	67.0	37.3	35.5	61.6	3.4	37.9
3NN	12	$\bar{\theta}$	89.4	91.4	6.4	8.7	7.1	6.8	35.0
PCLDC	25	$\bar{\theta}$	9.0	28.0	57.0	63.0	52.0	2.0	35.2
FISHERC	2	$\bar{\theta}$	61.8	47.7	30.5	12.7	27.8	2.6	30.5
NMC	25	$\bar{\theta}$	61.0	46.0	39.0	4.0	29.0	27.0	34.3
LDC	3	$\overline{\|\mathbf{v}\|}$	49.0	38.0	54.5	28.8	52.5	3.3	37.7
QDC	12	$\overline{\|\mathbf{v}\|}$	68.4	69.0	31.7	41.5	12.4	19.7	40.5
1NN	14	$\overline{\|\mathbf{v}\|}$	38.7	52.3	52.8	34.0	49.6	10.5	39.6
2NN	5	$\overline{\|\mathbf{v}\|}$	25.6	30.2	76.4	52.9	68.2	12.3	44.3
3NN	5	$\overline{\|\mathbf{v}\|}$	24.7	27.4	78.7	49.5	67.9	14.1	43.7
PCLDC	3	$\overline{\|\mathbf{v}\|}$	49.0	38.0	54.5	28.8	52.5	3.3	37.7
FISHERC	3	$\overline{\|\mathbf{v}\|}$	49.9	39.8	56.9	31.4	54.7	4.4	39.5
NMC	6	$\overline{\|\mathbf{v}\|}$	67.2	51.1	49.6	22.2	43.2	5.6	39.8
LDC	23	SFBS	1.0	25.0	62.0	63.0	63.0	2.0	36.0
QDC	13	SFBS	85.0	90.0	31.0	60.0	0.0	3.0	44.8
1NN	13	SFBS	38.0	39.0	63.0	83.0	78.0	9.0	51.7
2NN	13	SFBS	44.0	47.0	69.0	84.0	63.0	1.0	51.3
3NN	13	SFBS	41.0	47.0	66.0	80.0	62.0	4.0	50.0
PCLDC	23	SFBS	1.0	25.0	62.0	63.0	63.0	2.0	36.0
FISHERC	8	SFBS	77.0	39.0	31.0	23.0	47.0	1.0	36.3
NMC	14	SFBS	74.0	57.0	40.0	6.0	29.0	18.0	37.3

consistent scans between different users. For the feature spaces identified using the centroid distance ($\overline{\|\mathbf{v}\|}$) feature selection routine, the highest average accuracy across all six plates is from the 2NN classifier at 44.3%.

The centroid distance ($\overline{\|\mathbf{v}\|}$) feature selection results appear to be the best at evenly distributing the classification accuracies across all six plates, and even then only with a couple of the classification algorithms. This supports our suspicion that formal feature selection routines which optimize classification accuracies based on a specific training set may not be appropriate for damage characterization, which involve data sets containing sequentially distributed classes. Another pattern observed in the results is that the centroid distance ($\overline{\|\mathbf{v}\|}$) feature selection routine identified lower-dimensional (d) feature spaces when compared to the formal SFBS routine, even though many of the highest classifier accuracies are comparable between the two routines.

We now take a closer look at the average centroid distance results and the top SFBS results for testing plates $\mathcal{P}_1$ - $\mathcal{P}_6$ in Tables 5.16 - 5.23. Each table shows the classification accuracies of the given classifiers for all dimensions and all testing plates using both the average centroid distance and the top SFBS results as feature selection metrics. For ease of analysis, we highlight any configurations that correctly classify at least 50% of a test set's waveforms as being within ± 1 depth of the actual flaw severity for a given plate. Correctly classifying over 50% of the waveforms guarantees an accurate final damage characterization assuming a simple majority vote among class probabilities is used.

There is a trend in these results that supports the curse of dimensionality. As feature space dimension d increases, there is always one or two plates whose accuracy drops to below 10%. It is most often plate $\mathcal{P}_6$ that has these low accuracies, however the NMC accuracies actually increase for $\mathcal{P}_6$ as d increases while $\mathcal{P}_4$ accuracy decreases

Table 5.16: Validation plate results using the LDC classifier with feature spaces identified by the highest average distance between centroid ($\|\mathbf{v}\|$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

Dim.	LDC Classification Accuracy (±1Depth) [%]													
	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	51.0	50.0	23.0	15.1	31.4	3.6	29.0	72.0	61.0	12.0	4.0	15.0	10.0	29.0
2	54.7	48.3	39.8	20.6	33.2	1.7	33.1	55.0	33.0	18.0	5.0	9.0	16.0	22.7
3	49.0	38.0	54.5	28.8	52.5	3.3	37.7	24.0	30.0	42.0	16.0	36.0	9.0	26.2
4	61.6	43.3	49.6	22.6	43.8	2.5	37.2	66.0	51.0	34.0	7.0	7.0	10.0	29.2
5	66.4	48.9	32.8	19.3	41.3	4.9	35.6	82.0	56.0	23.0	2.0	5.0	28.0	32.7
6	71.6	51.5	27.1	15.8	35.7	7.0	34.8	83.0	55.0	23.0	1.0	2.0	17.0	30.2
7	65.8	43.4	25.7	20.1	41.5	7.5	34.0	40.0	29.0	31.0	9.0	20.0	2.0	21.8
8	51.4	25.4	27.4	20.5	45.4	5.3	29.2	57.0	35.0	30.0	28.0	49.0	6.0	34.2
9	48.5	25.9	24.8	20.4	45.0	7.6	28.7	55.0	35.0	37.0	27.0	48.0	0.0	33.7
10	37.1	18.8	29.2	24.9	47.5	4.8	27.1	55.0	35.0	37.0	27.0	48.0	0.0	33.7
11	52.5	28.1	27.1	16.0	39.1	7.9	28.4	53.0	28.0	39.0	29.0	55.0	0.0	34.0
12	31.2	16.1	30.0	28.5	48.4	3.7	26.3	53.0	28.0	39.0	29.0	55.0	0.0	34.0
13	22.7	11.8	26.7	34.0	49.6	2.8	24.6	1.0	13.0	26.0	32.0	60.0	2.0	22.3
14	13.3	10.0	30.2	40.0	52.4	3.2	24.9	26.0	9.0	38.0	26.0	48.0	4.0	25.2
15	13.4	13.0	32.6	39.6	53.8	3.8	26.0	33.0	19.0	38.0	23.0	47.0	4.0	27.3
16	14.2	11.4	33.2	43.0	53.8	3.0	26.4	7.0	3.0	61.0	60.0	54.0	8.0	32.2
17	8.6	18.8	34.2	47.0	55.2	3.0	27.8	8.0	4.0	64.0	58.0	56.0	7.0	32.8
18	8.1	18.7	34.4	47.5	55.1	3.2	27.8	8.0	4.0	64.0	58.0	56.0	7.0	32.8
19	8.2	15.8	35.3	48.9	54.3	2.4	27.5	8.0	3.0	62.0	57.0	48.0	7.0	30.8
20	8.4	18.8	35.3	49.8	55.4	2.5	28.4	8.0	3.0	62.0	57.0	48.0	7.0	30.8
21	8.6	19.1	36.1	50.2	55.8	2.6	28.7	9.0	23.0	66.0	57.0	50.0	7.0	35.3
22	10.2	21.2	38.6	50.8	55.7	2.7	29.9	6.0	32.0	46.0	45.0	52.0	3.0	30.7
23	10.6	22.7	44.2	53.3	55.3	1.8	31.3	1.0	25.0	62.0	63.0	63.0	2.0	36.0
24	9.6	27.9	54.8	60.9	53.3	1.9	34.7	4.0	26.0	60.0	64.0	53.0	2.0	34.8
25	9.0	28.0	57.0	63.0	52.0	2.0	35.2	9.0	28.0	57.0	63.0	52.0	2.0	35.2

Table 5.17: Validation plate results using the QDC classifier with feature spaces identified by the highest average distance between centroid ($\overline{\|\mathbf{v}\|}$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

Dim.	QDC Classification Accuracy (±1Depth) [%]													
	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	83.6	82.5	13.6	9.1	14.8	4.6	34.7	79.0	73.0	6.0	1.0	12.0	3.0	29.0
2	62.4	57.9	25.3	5.7	16.9	16.0	30.7	58.0	39.0	34.0	8.0	23.0	9.0	28.5
3	32.3	30.3	39.6	23.0	28.3	23.1	29.4	21.0	50.0	19.0	18.0	40.0	13.0	26.8
4	27.3	25.4	28.1	18.4	17.5	43.3	26.7	42.0	31.0	21.0	6.0	33.0	28.0	26.8
5	34.9	36.1	29.6	20.6	25.2	36.3	30.4	24.0	29.0	28.0	26.0	38.0	28.0	28.8
6	18.1	24.4	45.9	30.8	34.1	42.5	32.6	11.0	16.0	67.0	21.0	6.0	25.0	24.3
7	9.9	15.9	66.0	53.8	34.2	42.4	37.0	43.0	70.0	51.0	23.0	13.0	28.0	38.0
8	18.7	15.0	73.6	58.1	21.3	30.1	36.1	43.0	18.0	80.0	63.0	2.0	4.0	35.0
9	21.0	12.8	82.1	74.4	14.4	21.0	37.6	40.0	9.0	87.0	88.0	0.0	6.0	38.3
10	28.6	15.1	82.3	69.0	10.8	14.3	36.7	40.0	9.0	87.0	88.0	0.0	6.0	38.3
11	44.4	39.7	55.9	58.6	16.0	6.5	36.8	26.0	4.0	91.0	92.0	0.0	0.0	35.5
12	68.4	69.0	31.7	41.5	12.4	19.7	40.5	26.0	4.0	91.0	92.0	0.0	0.0	35.5
13	70.8	67.2	30.7	36.9	9.1	21.8	39.4	85.0	90.0	31.0	60.0	0.0	3.0	44.8
14	73.9	70.6	28.5	36.9	8.3	17.0	39.2	84.0	69.0	30.0	11.0	8.0	51.0	42.2
15	86.6	94.8	13.9	23.5	0.8	1.2	36.8	45.0	48.0	30.0	15.0	10.0	38.0	31.0
16	87.7	94.3	16.4	22.1	0.1	0.5	36.9	55.0	57.0	80.0	35.0	27.0	0.0	42.3
17	91.5	96.0	18.2	18.4	0.0	0.7	37.5	96.0	91.0	22.0	15.0	1.0	0.0	37.5
18	90.2	91.4	15.3	17.3	0.0	1.1	35.9	96.0	90.0	22.0	15.0	1.0	0.0	37.3
19	91.6	94.1	18.4	17.1	0.0	0.8	37.0	96.0	90.0	25.0	13.0	1.0	0.0	37.5
20	92.8	95.3	15.3	16.0	0.0	1.0	36.7	96.0	90.0	25.0	14.0	1.0	0.0	37.7
21	92.9	93.6	13.0	16.9	0.0	1.1	36.3	98.0	96.0	18.0	9.0	0.0	0.0	36.8
22	94.4	94.2	6.7	16.7	0.0	1.2	35.5	98.0	99.0	15.0	15.0	1.0	1.0	38.2
23	96.5	98.0	3.2	15.3	0.0	1.0	35.7	98.0	100.0	8.0	14.0	0.0	1.0	36.8
24	97.7	99.3	1.8	11.9	0.0	1.0	35.3	98.0	100.0	5.0	12.0	0.0	1.0	36.0
25	98.0	100.0	1.0	11.0	0.0	1.0	35.2	98.0	100.0	1.0	11.0	0.0	1.0	35.2

Table 5.18: Validation plate results using the 1NN classifier with feature spaces identified by the highest average distance between centroid ($\overline{\|\mathbf{v}\|}$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

	1NN Classification Accuracy (±1Depth) [%]													
Dim.	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	38.0	38.5	36.9	34.7	38.9	17.5	34.1	38.0	42.0	29.0	24.0	28.0	23.0	30.7
2	42.9	32.7	35.5	23.1	25.4	24.3	30.7	31.0	21.0	32.0	18.0	21.0	23.0	24.3
3	24.2	19.5	**60.6**	29.8	46.5	23.5	34.0	17.0	23.0	**62.0**	36.0	37.0	21.0	32.7
4	34.9	32.6	31.6	21.2	46.0	34.9	33.5	36.0	35.0	38.0	23.0	14.0	28.0	29.0
5	35.1	33.9	24.3	27.6	48.4	32.2	33.6	8.0	2.0	**88.0**	**85.0**	49.0	23.0	42.5
6	31.2	33.6	24.7	30.6	44.7	26.0	31.8	8.0	2.0	**90.0**	**87.0**	**54.0**	11.0	42.0
7	32.4	33.2	23.9	28.3	40.2	23.5	30.3	8.0	2.0	**89.0**	**88.0**	**56.0**	5.0	41.3
8	37.3	34.5	32.8	29.5	40.6	21.1	32.6	29.0	27.0	**65.0**	39.0	30.0	8.0	33.0
9	42.3	35.1	34.6	19.0	37.4	14.1	30.4	27.0	26.0	**66.0**	39.0	31.0	6.0	32.5
10	43.1	40.2	41.4	21.7	41.6	13.4	33.6	27.0	27.0	**69.0**	39.0	32.0	6.0	33.3
11	39.4	39.8	45.9	20.7	45.5	10.5	33.6	28.0	27.0	**68.0**	37.0	29.0	6.0	32.5
12	25.5	47.1	48.7	31.1	**53.3**	7.0	35.4	29.0	27.0	**66.0**	34.0	30.0	6.0	32.0
13	26.1	45.0	48.8	29.7	**52.5**	7.3	34.9	38.0	39.0	**63.0**	**83.0**	**78.0**	9.0	**51.7**
14	38.7	**52.3**	**52.8**	34.0	49.6	10.5	39.6	17.0	20.0	**58.0**	13.0	46.0	18.0	28.7
15	35.8	49.1	**50.8**	29.9	**50.4**	9.5	37.6	39.0	30.0	**59.0**	15.0	**61.0**	27.0	38.5
16	25.3	48.0	49.5	37.1	**53.9**	7.0	36.8	**57.0**	40.0	**78.0**	24.0	**80.0**	17.0	49.3
17	14.7	47.7	47.5	45.9	**57.4**	4.4	36.3	**56.0**	39.0	**79.0**	25.0	**80.0**	17.0	49.3
18	19.1	46.1	47.8	40.6	**56.9**	5.3	36.0	**56.0**	39.0	**79.0**	24.0	**80.0**	17.0	49.2
19	15.2	36.4	**51.8**	36.9	**56.6**	7.2	34.0	**56.0**	40.0	**79.0**	25.0	**80.0**	18.0	49.7
20	21.9	47.9	43.0	35.7	**57.3**	6.6	35.4	**57.0**	40.0	**79.0**	26.0	**79.0**	16.0	49.5
21	17.6	41.4	48.7	42.7	**61.2**	5.6	36.2	**59.0**	40.0	**79.0**	27.0	**79.0**	15.0	49.8
22	17.3	38.3	**50.3**	36.5	**65.2**	4.8	35.4	**56.0**	43.0	**58.0**	22.0	**68.0**	16.0	43.8
23	13.4	41.0	46.6	33.5	**68.9**	3.2	34.4	5.0	24.0	**62.0**	**61.0**	**73.0**	2.0	37.8
24	10.5	36.8	46.9	30.6	**71.0**	3.2	33.2	6.0	24.0	**62.0**	**56.0**	**72.0**	3.0	37.2
25	9.0	37.0	44.0	28.0	**71.0**	3.0	32.0	9.0	37.0	44.0	28.0	**71.0**	3.0	32.0

Table 5.19: Validation plate results using the 2NN classifier with feature spaces identified by the highest average distance between centroid ($\overline{\|\mathbf{v}\|}$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

Dim.	2NN Classification Accuracy (±1Depth) [%]													
	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	61.8	60.9	31.7	29.7	21.3	5.9	35.2	57.0	67.0	24.0	16.0	18.0	10.0	32.0
2	48.4	37.4	39.7	23.5	26.1	9.1	30.7	51.0	36.0	35.0	10.0	17.0	9.0	26.3
3	44.9	32.6	51.4	24.0	39.9	12.6	34.2	22.0	27.0	67.0	42.0	18.0	11.0	31.2
4	36.1	34.6	61.4	39.6	52.0	14.9	39.8	40.0	36.0	43.0	20.0	10.0	19.0	28.0
5	25.6	30.2	76.4	52.9	68.2	12.3	44.3	8.0	2.0	88.0	85.0	50.0	19.0	42.0
6	39.1	39.1	60.1	23.7	49.4	12.6	37.3	10.0	4.0	90.0	86.0	53.0	6.0	41.5
7	46.6	40.8	48.8	20.8	38.6	12.2	34.6	9.0	7.0	90.0	85.0	56.0	4.0	41.8
8	51.6	42.5	49.7	26.1	37.8	10.1	36.3	32.0	29.0	65.0	31.0	28.0	5.0	31.7
9	55.9	42.9	44.7	19.1	35.3	11.7	34.9	31.0	30.0	65.0	30.0	29.0	3.0	31.3
10	57.3	40.7	44.0	17.1	35.2	11.5	34.3	30.0	30.0	68.0	31.0	31.0	2.0	32.0
11	49.1	42.9	45.8	24.9	41.8	10.9	35.9	32.0	29.0	67.0	27.0	29.0	4.0	31.3
12	40.9	57.1	49.0	37.0	45.3	8.4	39.6	32.0	29.0	66.0	24.0	30.0	4.0	30.8
13	44.4	43.6	44.8	28.0	43.6	10.0	35.7	44.0	47.0	69.0	84.0	63.0	1.0	51.3
14	47.9	55.5	49.7	32.1	41.3	9.8	39.4	50.0	41.0	46.0	10.0	47.0	15.0	34.8
15	46.6	53.7	49.0	30.3	41.9	9.3	38.5	58.0	40.0	47.0	10.0	61.0	13.0	38.2
16	32.9	56.0	48.7	43.9	50.2	6.0	39.6	69.0	48.0	70.0	22.0	63.0	6.0	46.3
17	36.3	61.7	42.3	33.8	46.1	6.9	37.9	69.0	48.0	70.0	22.0	64.0	6.0	46.5
18	25.9	50.7	47.1	42.7	50.9	4.3	36.9	70.0	48.0	70.0	21.0	64.0	6.0	46.5
19	24.5	55.7	40.8	37.6	51.2	4.8	35.8	71.0	48.0	70.0	19.0	63.0	6.0	46.2
20	21.6	52.2	43.4	40.8	54.1	2.1	35.7	72.0	47.0	70.0	19.0	62.0	7.0	46.2
21	20.4	50.7	41.7	38.3	58.8	1.6	35.3	71.0	48.0	69.0	20.0	64.0	7.0	46.5
22	26.4	55.7	39.4	32.7	59.1	2.0	35.9	66.0	49.0	60.0	17.0	59.0	6.0	42.8
23	26.0	53.9	38.4	30.3	62.9	1.0	35.4	11.0	36.0	35.0	39.0	73.0	1.0	32.5
24	25.7	57.9	38.0	28.7	63.7	1.5	35.9	11.0	36.0	35.0	38.0	72.0	1.0	32.2
25	26.0	57.0	40.0	30.0	64.0	1.0	36.3	26.0	57.0	40.0	30.0	64.0	1.0	36.3

Table 5.20: Validation plate results using the 3NN classifier with feature spaces identified by the highest average distance between centroid ($\|\mathbf{v}\|$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

Dim.	3NN Classification Accuracy (±1Depth) [%]													
	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	71.6	67.5	24.6	21.0	14.3	7.7	34.4	63.0	74.0	19.0	8.0	13.0	11.0	31.3
2	45.5	37.7	43.2	25.5	25.9	14.4	32.0	55.0	35.0	33.0	7.0	12.0	9.0	25.2
3	31.8	26.1	59.0	27.9	50.5	17.0	35.4	27.0	25.0	60.0	38.0	26.0	12.0	31.3
4	30.0	29.2	67.0	36.2	54.7	19.8	39.5	39.0	37.0	38.0	21.0	8.0	22.0	27.5
5	24.7	27.4	78.7	49.5	67.9	14.1	43.7	7.0	2.0	86.0	85.0	49.0	22.0	41.8
6	32.4	33.0	54.0	21.0	47.0	16.7	34.0	9.0	5.0	89.0	85.0	51.0	10.0	41.5
7	40.0	36.5	54.2	20.7	43.2	15.5	35.0	8.0	8.0	90.0	87.0	56.0	5.0	42.3
8	46.8	40.3	51.6	24.0	39.9	14.6	36.2	33.0	27.0	62.0	32.0	28.0	10.0	32.0
9	53.1	41.6	47.4	16.3	37.0	13.7	34.8	32.0	29.0	65.0	31.0	30.0	8.0	32.5
10	51.2	39.5	47.9	16.0	38.1	13.9	34.4	31.0	29.0	65.0	28.0	31.0	5.0	31.5
11	49.1	38.3	46.2	18.2	42.7	12.8	34.5	31.0	28.0	65.0	25.0	29.0	5.0	30.5
12	58.5	39.9	44.4	14.1	42.9	16.0	36.0	32.0	28.0	65.0	25.0	30.0	4.0	30.7
13	57.6	39.7	44.7	14.4	41.3	16.0	35.6	41.0	47.0	66.0	80.0	62.0	4.0	50.0
14	29.2	31.4	52.8	30.2	45.6	8.4	32.9	51.0	40.0	48.0	9.0	44.0	15.0	34.5
15	29.0	39.7	43.0	19.7	45.5	8.8	31.0	57.0	42.0	48.0	10.0	52.0	15.0	37.3
16	30.6	44.6	48.2	29.8	49.4	6.4	34.8	80.0	47.0	64.0	16.0	54.0	6.0	44.5
17	27.3	46.2	48.8	32.5	50.3	6.8	35.3	80.0	47.0	64.0	15.0	53.0	6.0	44.2
18	23.9	41.4	48.7	34.0	50.7	5.7	34.1	81.0	48.0	64.0	14.0	53.0	6.0	44.3
19	29.8	43.3	47.3	33.5	52.9	5.4	35.4	81.0	47.0	63.0	14.0	53.0	6.0	44.0
20	27.3	55.4	38.0	34.7	54.1	3.6	35.5	81.0	45.0	63.0	16.0	55.0	6.0	44.3
21	21.6	47.8	39.9	36.7	54.5	2.6	33.9	80.0	46.0	63.0	16.0	54.0	6.0	44.2
22	37.4	60.6	36.8	23.2	49.9	3.2	35.2	78.0	51.0	49.0	14.0	48.0	6.0	41.0
23	24.7	53.7	36.7	23.0	59.2	1.5	33.1	25.0	59.0	30.0	22.0	53.0	1.0	31.7
24	25.1	48.8	37.0	22.0	55.4	1.3	31.6	25.0	58.0	29.0	23.0	48.0	1.0	30.7
25	24.0	47.0	39.0	24.0	54.0	1.0	31.5	24.0	47.0	39.0	24.0	54.0	1.0	31.5

Table 5.21: Validation plate results using the PCLDC classifier with feature spaces identified by the highest average distance between centroid ($\overline{\|\mathbf{v}\|}$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

Dim.	PCLDC Classification Accuracy (±1Depth) [%]													
	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	51.0	50.0	23.0	15.1	31.4	3.6	29.0	72.0	61.0	12.0	4.0	15.0	10.0	29.0
2	54.7	48.3	39.8	20.6	33.2	1.7	33.1	55.0	33.0	18.0	5.0	9.0	16.0	22.7
3	49.0	38.0	54.5	28.8	52.5	3.3	37.7	24.0	30.0	42.0	16.0	36.0	9.0	26.2
4	61.6	43.3	49.6	22.6	43.8	2.5	37.2	66.0	51.0	34.0	7.0	7.0	10.0	29.2
5	66.4	48.9	32.8	19.3	41.3	4.9	35.6	82.0	56.0	23.0	2.0	5.0	28.0	32.7
6	71.6	51.5	27.1	15.8	35.7	7.0	34.8	83.0	55.0	23.0	1.0	2.0	17.0	30.2
7	65.8	43.4	25.7	20.1	41.5	7.5	34.0	40.0	29.0	31.0	9.0	20.0	2.0	21.8
8	51.4	25.4	27.4	20.5	45.4	5.3	29.2	57.0	35.0	30.0	28.0	49.0	6.0	34.2
9	48.5	25.9	24.8	20.4	45.0	7.6	28.7	55.0	35.0	37.0	27.0	48.0	0.0	33.7
10	37.1	18.8	29.2	24.9	47.5	4.8	27.1	55.0	35.0	37.0	27.0	48.0	0.0	33.7
11	52.5	28.1	27.1	16.0	39.1	7.9	28.4	53.0	28.0	39.0	29.0	55.0	0.0	34.0
12	31.2	16.1	30.0	28.5	48.4	3.7	26.3	53.0	28.0	39.0	29.0	55.0	0.0	34.0
13	22.7	11.8	26.7	34.0	49.6	2.8	24.6	1.0	13.0	26.0	32.0	60.0	2.0	22.3
14	13.3	10.0	30.2	40.0	52.4	3.2	24.9	26.0	9.0	38.0	26.0	48.0	4.0	25.2
15	13.4	13.0	32.6	39.6	53.8	3.8	26.0	33.0	19.0	38.0	23.0	47.0	4.0	27.3
16	14.2	11.4	33.2	43.0	53.8	3.0	26.4	7.0	3.0	61.0	60.0	54.0	8.0	32.2
17	8.6	18.8	34.2	47.0	55.2	3.0	27.8	8.0	4.0	64.0	58.0	56.0	7.0	32.8
18	8.1	18.7	34.4	47.5	55.1	3.2	27.8	8.0	4.0	64.0	58.0	56.0	7.0	32.8
19	8.2	15.8	35.3	48.9	54.3	2.4	27.5	8.0	3.0	62.0	57.0	48.0	7.0	30.8
20	8.4	18.8	35.3	49.8	55.4	2.5	28.4	8.0	3.0	62.0	57.0	48.0	7.0	30.8
21	8.6	19.1	36.1	50.2	55.8	2.6	28.7	9.0	23.0	66.0	57.0	50.0	7.0	35.3
22	10.2	21.2	38.6	50.8	55.7	2.7	29.9	6.0	32.0	46.0	45.0	52.0	3.0	30.7
23	10.6	22.7	44.2	53.3	55.3	1.8	31.3	1.0	25.0	62.0	63.0	63.0	2.0	36.0
24	9.6	27.9	54.8	60.9	53.3	1.9	34.7	4.0	26.0	60.0	64.0	53.0	2.0	34.8
25	9.0	28.0	57.0	63.0	52.0	2.0	35.2	9.0	28.0	57.0	63.0	52.0	2.0	35.2

Table 5.22: Validation plate results using the FISHRC classifier with feature spaces identified by the highest average distance between centroid ($\|\mathbf{v}\|$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

Dim.	FISHERC Classification Accuracy (±1Depth) [%]													
	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	63.0	61.6	11.3	8.5	24.8	0.7	28.3	79.0	56.0	15.0	8.0	6.0	0.0	27.3
2	70.5	62.5	27.6	21.6	27.9	2.7	35.5	65.0	39.0	17.0	4.0	10.0	1.0	22.7
3	49.9	39.8	56.9	31.4	54.7	4.4	39.5	24.0	30.0	45.0	14.0	34.0	7.0	25.7
4	63.6	46.5	45.1	24.5	44.9	4.1	38.1	74.0	53.0	29.0	8.0	0.0	0.0	27.3
5	64.0	48.7	25.8	17.3	42.5	6.2	34.1	81.0	56.0	15.0	0.0	0.0	33.0	30.8
6	69.1	57.9	23.1	19.8	44.9	10.8	37.6	78.0	55.0	12.0	0.0	3.0	23.0	28.5
7	66.8	51.2	24.2	24.5	44.5	10.0	36.9	37.0	22.0	30.0	5.0	15.0	1.0	18.3
8	69.2	49.0	22.3	21.5	45.6	8.5	36.0	77.0	39.0	31.0	23.0	47.0	1.0	36.3
9	52.6	33.3	16.5	17.9	43.5	10.2	29.0	73.0	37.0	36.0	22.0	44.0	0.0	35.3
10	48.4	27.2	23.2	23.8	46.1	7.1	29.3	73.0	37.0	36.0	22.0	44.0	0.0	35.3
11	62.4	37.1	18.5	20.0	36.0	9.1	30.5	68.0	32.0	40.0	25.0	52.0	0.0	36.2
12	44.0	25.4	29.0	30.8	45.4	4.9	29.9	68.0	32.0	40.0	25.0	52.0	0.0	36.2
13	35.4	22.5	28.6	33.1	43.6	3.8	27.8	1.0	0.0	23.0	33.0	52.0	5.0	19.0
14	19.6	9.2	34.2	38.4	46.2	3.0	25.1	12.0	6.0	41.0	29.0	44.0	4.0	22.7
15	12.8	8.6	35.2	41.1	46.1	3.0	24.5	36.0	28.0	39.0	24.0	43.0	7.0	29.5
16	9.3	5.7	34.8	41.9	46.0	4.4	23.7	8.0	3.0	67.0	62.0	58.0	9.0	34.5
17	14.3	6.5	37.0	41.5	45.8	3.7	24.8	8.0	3.0	65.0	60.0	54.0	8.0	33.0
18	12.3	7.8	34.3	41.6	45.8	3.5	24.2	8.0	3.0	65.0	60.0	54.0	8.0	33.0
19	12.7	5.7	39.7	44.6	46.8	3.9	25.6	8.0	3.0	70.0	60.0	41.0	7.0	31.5
20	9.7	2.3	49.5	50.9	48.5	6.6	27.9	8.0	3.0	70.0	60.0	41.0	7.0	31.5
21	12.9	2.2	50.3	52.0	49.0	6.8	28.9	7.0	4.0	72.0	59.0	44.0	9.0	32.5
22	8.0	3.4	41.5	50.4	45.7	4.4	25.6	8.0	6.0	48.0	45.0	47.0	8.0	27.0
23	6.5	2.3	49.6	53.3	45.5	5.3	27.1	2.0	1.0	59.0	57.0	58.0	7.0	30.7
24	5.1	1.6	57.9	56.8	45.4	4.8	28.6	2.0	1.0	63.0	58.0	45.0	5.0	29.0
25	4.0	1.0	60.0	58.0	44.0	5.0	28.7	4.0	1.0	60.0	58.0	44.0	5.0	28.7

Table 5.23: Validation plate results using the NMC classifier with feature spaces identified by the highest average distance between centroid ($\|\mathbf{v}\|$) metric as well as the sequential floating backward search (SFBS) feature selection routines. Accuracies (within ±1 flaw depth) for each plate $\mathcal{P}_1$ - $\mathcal{P}_6$ as well as the average accuracy across all six plates are shown for each feature subspace of dimension $d = 1, \ldots, 25$. Any classification accuracy above 50% is highlighted in grey.

Dim.	NMC Classification Accuracy (±1Depth) [%]													
	Top Centroid Dist							Top SFBS						
d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
1	69.1	70.5	13.9	7.9	20.7	11.2	32.2	68.0	58.0	8.0	3.0	24.0	12.0	28.8
2	87.8	86.8	9.1	12.5	12.0	12.9	36.9	57.0	42.0	15.0	6.0	6.0	12.0	23.0
3	73.9	76.1	23.7	19.4	15.8	14.4	37.2	45.0	44.0	20.0	7.0	18.0	8.0	23.7
4	78.4	74.1	22.0	12.3	18.9	13.2	36.5	58.0	55.0	27.0	6.0	2.0	11.0	26.5
5	64.4	48.5	52.2	25.0	38.6	5.2	39.0	10.0	2.0	89.0	76.0	2.0	19.0	33.0
6	67.2	51.1	49.6	22.2	43.2	5.6	39.8	9.0	2.0	89.0	76.0	2.0	15.0	32.2
7	73.1	55.6	42.7	15.7	42.0	6.8	39.3	9.0	2.0	90.0	76.0	9.0	12.0	33.0
8	73.8	55.5	40.4	12.1	41.2	8.6	38.6	47.0	38.0	45.0	1.0	7.0	11.0	24.8
9	74.6	56.8	32.4	7.2	31.8	13.1	36.0	47.0	38.0	46.0	1.0	9.0	9.0	25.0
10	75.9	56.9	36.7	6.7	29.0	14.2	36.6	48.0	38.0	44.0	1.0	11.0	7.0	24.8
11	76.0	58.5	32.6	6.0	23.1	15.7	35.3	48.0	38.0	44.0	1.0	9.0	9.0	24.8
12	75.4	56.0	39.5	6.0	27.2	19.2	37.2	49.0	38.0	45.0	2.0	7.0	9.0	25.0
13	75.0	55.7	39.5	6.0	27.8	19.2	37.2	63.0	46.0	43.0	8.0	32.0	6.0	33.0
14	74.6	56.1	40.3	5.0	28.4	16.8	36.9	74.0	57.0	40.0	6.0	29.0	18.0	37.3
15	73.3	55.2	39.7	4.1	28.4	18.0	36.5	74.0	57.0	39.0	6.0	25.0	18.0	36.5
16	72.8	54.6	39.2	4.2	27.8	18.6	36.2	67.0	48.0	42.0	5.0	21.0	19.0	33.7
17	72.3	52.0	39.6	5.0	29.2	18.4	36.1	68.0	49.0	42.0	5.0	22.0	19.0	34.2
18	72.4	52.4	40.0	4.8	29.4	18.8	36.3	68.0	49.0	41.0	6.0	22.0	19.0	34.2
19	70.6	50.5	39.4	4.6	29.7	20.2	35.8	68.0	49.0	42.0	5.0	21.0	19.0	34.0
20	72.0	51.3	39.0	4.7	30.4	18.9	36.0	68.0	49.0	43.0	5.0	25.0	19.0	34.8
21	70.7	50.0	39.0	4.5	30.3	20.3	35.8	68.0	47.0	42.0	4.0	28.0	19.0	34.7
22	67.7	48.1	39.1	4.2	29.1	23.3	35.3	71.0	54.0	39.0	4.0	31.0	20.0	36.5
23	64.0	46.2	39.0	4.3	27.1	26.7	34.5	65.0	46.0	40.0	5.0	30.0	26.0	35.3
24	62.4	45.0	39.5	4.0	26.1	26.5	33.9	65.0	46.0	40.0	4.0	29.0	26.0	35.0
25	61.0	46.0	39.0	4.0	29.0	27.0	34.3	61.0	46.0	39.0	4.0	29.0	27.0	34.3

instead. This apparent trade-off supports the concept that feature spaces of dimension d that follow $n/d \geq 10$ are among the best performing.

There exist several feature spaces that are able to better classify plate $\mathcal{P}_6$. For example, the QDC results for $\overline{\|\mathbf{v}\|}$ feature space of dimension $d = 4$ returns 43.3% accuracy for plate $\mathcal{P}_6$. There is again a trade-off here as accuracies across the other five plates are somewhat lower, however, resulting in a lower average accuracy of 26.7% for the six plates together.

To explore the differences between the $\overline{\|\mathbf{v}\|}$ feature selection metric and the SFBS feature selection routine, we present Table 5.24. Ideally, all 6 plates would be classified evenly and accurately. However we have already shown that is not the case. We therefore consider the top 5 plate subsets in our results, followed by the top 4 plate subsets, etc. for all classifiers. We present in Table 5.24 the top n-plate subset accuracy shown as accuracy-per-plate for $n = 1, \ldots, 6$. For each value of n, the feature subset dimensions $d = 1, \ldots, 25$ are considered from Tables 5.16 - 5.23 for both the CD metric and the SFBS results.

The trend of decreasing average classifier performance as the number of plates is increased from $n = 1$ to $n = 6$ is due to the fact that no classifier was able to correctly identify all six plates simultaneously. Individual plate performance was highest ($n = 1$), with two plate performance second highest ($n = 2$), and so on. We see here that the highest 6 plate classifier performance is with the 1NN and 2NN classifiers with the SFBS feature set of dimension $d = 13$. Looking back to Tables 5.18 and 5.19, we can see that these correspond to a rather high accuracy for plates $\mathcal{P}_1$ - $\mathcal{P}_5$ but essentially zero accuracy on plate $\mathcal{P}_6$.

Another trend that we see here is that of the performance of the $\overline{\|\mathbf{v}\|}$ metric compared to the standard SFBS feature selection routine. In general, the full six plate test set ($n = 6$) shows split results between the $\overline{\|\mathbf{v}\|}$ and the SFBS feature

Table 5.24: average distance between centroid ($\overline{\|\mathbf{v}\|}$) metric classification accuracies compared to the top SFBS classification accuracies. Each row represents an n-plate subset, while each group of columns corresponds to an individual classifier. Results are presented as the top n-plate accuracy (with the corresponding feature subset dimensions d).

n-plates	LDC		QDC		1NN		2NN	
	$\overline{\|\mathbf{v}\|}$	SFBS	$\overline{\|\mathbf{v}\|}$	SFBS	$\overline{\|\mathbf{v}\|}$	SFBS	$\overline{\|\mathbf{v}\|}$	SFBS
1	71.6 (6)	83.0 (6)	100.0 (25)	100.0 (23)	71.0 (24)	90.0 (6)	76.4 (5)	90.0 (6)
2	61.5 (6)	69.0 (5)	99.0 (25)	99.0 (23)	59.0 (24)	88.5 (6)	72.3 (5)	88.0 (6)
3	57.3 (25)	62.7 (23)	69.9 (23)	78.3 (13)	52.2 (23)	77.7 (7)	65.8 (5)	77.0 (7)
4	50.0 (25)	53.3 (23)	56.0 (17)	66.5 (13)	49.6 (17)	65.8 (13)	56.9 (5)	65.8 (13)
5	44.6 (3)	43.0 (23)	46.1 (12)	53.8 (13)	45.5 (14)	60.2 (13)	50.7 (5)	61.4 (13)
6	37.7 (3)	36.0 (23)	40.5 (12)	44.8 (13)	39.6 (14)	51.7 (13)	44.3 (5)	51.3 (13)

n-plates	3NN		PCLDC		FISHERC		NMC	
	$\overline{\|\mathbf{v}\|}$	SFBS	$\overline{\|\mathbf{v}\|}$	SFBS	$\overline{\|\mathbf{v}\|}$	SFBS	$\overline{\|\mathbf{v}\|}$	SFBS
1	78.7 (5)	90.0 (7)	71.6 (6)	83.0 (6)	70.5 (2)	81.0 (5)	87.8 (2)	90.0 (7)
2	73.3 (5)	88.5 (7)	61.5 (6)	69.0 (5)	66.5 (2)	68.5 (5)	87.3 (2)	83.0 (7)
3	65.4 (5)	77.7 (7)	57.3 (25)	62.7 (23)	57.3 (6)	62.3 (16)	62.5 (2)	61.3 (5)
4	55.9 (5)	63.7 (13)	50.0 (25)	53.3 (23)	50.3 (3)	49.0 (16)	53.3 (7)	50.0 (14)
5	49.6 (5)	59.2 (13)	44.6 (3)	43.0 (23)	46.5 (3)	43.4 (8)	46.7 (6)	43.6 (14)
6	43.7 (5)	50.0 (13)	37.7 (3)	36.0 (23)	39.5 (3)	36.3 (8)	39.8 (6)	37.3 (14)

selection routines in terms of higher accuracy for a given classifier. The greatest disparity occurs for the k-NN classifiers, where the SFBS accuracies are noticeably higher than the $\overline{\|\mathbf{v}\|}$. This is not an unexpected result, since the k-NN accuracy was used as a criterion function (with $k = 1$) to select the SFBS features. The resulting feature set should be close to optimal for those classifiers. It is encouraging that for the remaining classifiers, the $\overline{\|\mathbf{v}\|}$ metric results are higher than the corresponding SFBS feature sets. This indicates that there is potential for a geometric feature space-based selection routine to use with flaw depth classification applications.

The kNN classifiers and the QDC perform best for overall higher-dimensional feature spaces ($d > 3$). The kNN classifiers return the highest average accuracies for all six plates at 51.7%, as well as the highest accuracies for 5-out-of-6 plates at an average accuracy per plate of 61.4%. However we have shown that this corresponds to an uneven spread in classifier accuracy across the six plates. The QDC results are more evenly spread out across the six plates, and therefore represent a better classifier configuration for the problem at hand.

Since the goal of this classification system is to be able to identify *all* testing plate samples with relative accuracy, the most even spread of accuracies across all six plates is of interest to us as well. Table 5.25 presents the highest *minimum* individual plate accuracy across all six plates for each classifier over all feature space dimensions considered. By sorting the results in this manner, we remove those classifiers which sacrifice an individual plate performance to achieve better average accuracies. This table is presented only for the centroid distance feature selection routine ($\overline{\|\mathbf{v}\|}$) results.

We can see that average accuracies drop to just below 40% when considering the highest minimum performance across all plates. This represents a more even classifier performance, where feature spaces have been identified that returned accuracies for all individual plates of at least 24.7% (1NN, $d = 6$). It is important to note that while

Table 5.25: Classifier configurations for the highest minimum accuracy for each classifier under consideration for the centroid distance feature selection routine ($\overline{\|\mathbf{v}\|}$). As the goal of this classifier is to perform well for any testing plate submitted, we sort the classifier configuration results by the highest minimum accuracy to show classifiers with the most even spread of accuracies across all six testing plates. For each classifier, the feature subset dimension corresponding to the highest average accuracy is presented as d.

	Dim.	Accuracy [% within ±1 flaw depth]						
Classifier	d	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	Avg.
LDC	11	52.5	28.1	27.1	16.0	39.1	7.9	28.4
QDC	3	32.3	30.3	39.6	23.0	28.3	23.1	29.4
1NN	6	31.2	33.6	24.7	30.6	44.7	26.0	31.8
2NN	4	36.1	34.6	61.4	39.6	52.0	14.9	39.8
3NN	4	30.0	29.2	67.0	36.2	54.7	19.8	39.5
PCLDC	11	52.5	28.1	27.1	16.0	39.1	7.9	28.4
FISHERC	6	69.1	57.9	23.1	19.8	44.9	10.8	37.6
NMC	3	73.9	76.1	23.7	19.4	15.8	14.4	37.2

we present classification accuracies for each plate individually here, these values do not provide information on the distribution of the incorrectly classified labels. As discussed in Section 5.6.2, the classifier outputs are converted to confidence intervals across the full span of possible testing classes D_1 - D_{14}. We have reduced these to a final predicted label for each waveform, decided by which class has the highest predicted confidence. These predicted labels for each waveform therefore span the range of possible classes ($D_1 = D_{14}$), resulting in an output label distribution for each testing plate $\mathcal{P}_1$ - $\mathcal{P}_6$. It is natural to think that a majority-vote is appropriate to further reduce this output label distribution to a single class D_j. This may have an effect on these results.

Determining whether or not a majority vote is appropriate for final predicted label reduction depends on the final application of a classifier. For example, technicians will not be interested in considering the raw output label distribution among the classes D_1 - D_{14}. Instead, their interest would be in a single output from the classifier, where a combination algorithm reduces the predicted labels to a single flaw depth value. Alternatively, they may be interested in the percent of waveforms given a label corresponding to a flaw that is more severe than a specific threshold, i.e. only those labeled $\omega_{n,\ldots,14}$ where n is determined by the user.

5.8.1 Example Demonstration

We now illustrate the classifier capabilities in the context of an example demonstration of SHM where Lamb waves are used to determine structural integrity throughout a lifetime of use. The capacity to carry a specific load is unique to each individual structure, and is usually over-engineered with a factor of safety (FoS) to account for unexpected loads and inherent defects in the structure. The FoS is formally defined as the ratio of the structure's strength over the expected maximum load. In the aviation industry, a factor of safety standard[4] is 1.5, a relatively low value that is the result of engineering o reduce weight and increase efficiency [157]. As a result, aircraft structures require frequent inspection due to the narrow margin of error associated with material loss from damage.

By identifying material losses that reach a given threshold level, we can evaluate the classifier's potential for use as a damage identification system. We present over/under classification accuracy threshold results for the six testing plates using two different damage thresholds of 15% and 30% material loss using the QDC classifier. For a structure with an initial FoS value of 1.5, any damage resulting in material loss will reduce the effective FoS. If enough damage occurs to reduce the FoS down to an effective value of 1.0, then the damage can be thought of as critical since the structure's strength would only be equal to the expected maximum load level. Any

[4] The FAA standard currently used states "Unless otherwise specified, a factor of safety of 1.5 must be applied to the prescribed limit load which are considered external loads on the structure." *Airworthiness Standards: Transport Category Airplanes*, FAA Federal Aviation Regulations, Sec. 25.303 "Factor of Safety", (1970).

unexpected load fluctuations would then result in failure. Thresholds of 15% and 30% material loss explored here correspond to effective FoS reductions in aerospace applications of 1.25 and 1.0, respectively. For each threshold value, we determine the number of waveforms from each testing plate that get identified as being above and below the threshold thickness. This translates to determining the number of predicted labels of ω_n where $n = 4, \ldots, 14$ and $n = 7, \ldots, 14$ for the 15% and 30% thresholds, respectively. If over 50% of the waveforms from a testing plate are classified as being more severe than the threshold, we feel confident assigning a 'more severe' label to that plate. If less than 50% of the waveforms are classified as being more severe than the threshold, we assign a 'less severe' label to that plate. For both 15% and 30% material loss threshold levels, we expect plates $\mathcal{P}_1$ and $\mathcal{P}_2$ (both 10% thickness loss) to be identified as 'less severe' while plates $\mathcal{P}_3$, $\mathcal{P}_4$, $\mathcal{P}_5$, and $\mathcal{P}_6$ (45%, 45%, 60%, and 100% thickness loss, respectively) should be labeled as 'more severe'. Results are presented in Table 5.26.

Plates $\mathcal{P}_1$ and $\mathcal{P}_2$, which are known to have only 10% thickness loss, are both correctly identified as having damage that is less severe than the 15% damage threshold for feature spaces of dimension $d = 13$ and $d = 14$. Similarly, plates $\mathcal{P}_3$, $\mathcal{P}_4$, $\mathcal{P}_5$, and $\mathcal{P}_6$ have thickness losses of 45%, 45%, 60 %, and 100% and are all correctly identified as having flaws that are beyond the 15% damage threshold limit. Similar results exist for the 30% damage threshold limit, where all six plates are correctly determined to be either above or below each damage threshold considered. We have demonstrated here a classification routine that accurately identifies six individual test plates as being either under or over the damage threshold limits of 15% and 30% material loss, corresponding to roughly a 50% and 100% reduction in the FoS used in aerospace applications.

Table 5.26: Over/under flaw severity thresholding results using the QDC classifier for feature spaces of dimension $d = 13, 14$ selected using the average centroid distance ($\overline{\|\mathbf{v}\|}$) metric for plates $\mathcal{P}_1$ - $\mathcal{P}_6$. Two thresholds are considered, 15% material loss and 30% material loss. The percentage of waveforms labeled above each threshold from a given test plate is shown here.

Dimension	QDC											
	> 15% Damage Threshold						> 30% Damage Threshold					
	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$	$\mathcal{P}_1$	$\mathcal{P}_2$	$\mathcal{P}_3$	$\mathcal{P}_4$	$\mathcal{P}_5$	$\mathcal{P}_6$
1	39.0	44.1	51.0	38.2	52.3	62.1	16.4	17.5	24.1	16.0	27.1	25.2
2	43.3	49.6	61.0	32.9	54.5	70.7	37.6	42.1	52.9	29.3	44.2	50.5
3	70.5	73.2	84.3	53.7	80.1	80.7	67.7	69.7	81.6	52.9	77.6	79.6
4	77.6	76.0	90.0	73.4	90.9	83.1	72.7	74.6	87.4	71.5	86.9	81.6
5	71.1	67.1	86.1	68.0	91.8	83.4	65.1	63.9	81.2	62.4	85.1	81.4
6	88.2	79.9	87.9	73.9	92.5	81.6	81.9	75.6	82.5	67.5	85.6	79.5
7	95.2	90.2	95.6	86.4	93.9	80.5	90.1	84.1	92.9	79.0	89.2	78.6
8	94.5	90.4	96.8	89.7	95.2	78.5	81.3	85.0	93.1	73.4	89.9	75.2
9	88.4	91.1	92.6	92.3	91.2	85.3	79.0	87.2	90.9	81.2	88.4	81.8
10	83.0	91.3	92.1	87.7	90.1	82.0	71.4	84.9	90.0	75.0	88.2	76.4
11	74.6	74.2	74.6	81.5	73.6	84.7	55.6	60.3	66.6	65.9	70.9	74.2
12	48.4	50.8	59.9	59.6	60.6	83.4	31.6	31.0	48.8	51.6	58.9	67.0
13	43.4	50.0	72.6	61.1	67.1	94.2	29.2	32.8	60.0	55.2	65.5	79.6
14	42.3	49.3	71.7	60.8	68.8	91.6	26.1	29.4	55.6	53.6	66.0	72.3
15	20.3	21.1	34.9	34.5	55.9	86.5	13.4	5.2	27.7	31.2	54.1	52.7
16	23.1	22.4	36.5	35.1	57.1	84.6	12.3	5.7	25.9	29.5	54.0	51.9
17	27.6	28.6	47.7	49.6	53.9	71.9	8.5	4.0	23.5	35.7	45.8	49.2
18	21.5	24.8	40.8	54.3	47.7	65.6	9.8	8.6	25.8	37.5	41.7	49.5
19	25.2	28.6	44.4	55.6	49.1	67.2	8.4	5.9	26.2	36.3	41.3	48.0
20	22.3	27.1	38.4	59.0	37.5	56.3	7.2	4.7	21.7	35.6	30.4	44.1
21	17.9	23.3	29.5	60.6	29.4	47.2	7.1	6.4	19.0	40.2	23.4	38.9
22	10.6	12.4	17.1	63.2	18.3	28.7	5.6	5.8	11.9	51.2	13.1	26.2
23	5.7	3.6	8.4	56.7	8.9	12.9	3.5	2.0	7.4	50.8	4.5	12.9
24	2.5	0.7	4.5	47.4	5.6	7.0	2.3	0.7	4.4	42.9	2.9	7.0
25	2.0	0.0	3.0	42.0	2.0	6.0	2.0	0.0	3.0	40.0	1.0	6.0

5.9 Discussion

We have presented here a study of multiple-mode Lamb wave interactions with simulated corrosion thinnings for autonomous classification of flaw severity. Lamb wave tomography was used to first reconstruct an image of the flaw and thereby locate and size it. Waveforms of interest were identified automatically using raypaths that had passed through the flaw area in each tomographic reconstruction. Using these waveforms, several feature selection methods were explored to extract information about the complex scattering and mode conversion that can happen. These waveform features were then evaluated using formal classification accuracies as a measure of performance. The fundamental concept of class distribution within each feature space with respect to the sequential ordering of classes was discussed. Measurements using the class centroid locations within the various feature space subsets were found that allow intelligent selection of those subsets which result in high classification accuracies. Using this selection metric, six validation data sets were submitted to the classifier to assess the classifier's ability to correctly label new data when it falls between existing flaw depths (classes). We found that most of these validation sets were correctly identified with respect to their flaw severity, however most classifier configurations struggled to simultaneously label all of the validations sets correctly. Classifiers often sacrificed the performance on one validation set to increase the accuracy of the remaining sets. Classifiers which performed best for all validation sets were then used in an example application more relevant to structural health monitoring, where the multi-class problem was reduced to a binary output of 'more' or 'less' severe than a given damage threshold. All validation data sets were correctly assessed using this thresholding, indicating the potential use of this classification scheme to identify damaged structures in a straightforward way.

We have concluded that formal feature selection routines rely too heavily on the concept that the testing set is a representative collection of data for a particular problem domain. In the case of damage detection, it is not feasible to provide training sets representing all possible flaw types to the classifiers. Classifiers are guaranteed to be tested on data from unknown flaws, yet are still expected to accurately asses their severity. Our work suggests exploring topological measures of the class distribution within the feature space as a way to predict classifier accuracies for testing new data. Our work provides the framework for a classification routine which is able to pull features from complex, multi-mode Lamb wave signals to identify signatures related to flaw severity.

With the promising performance of the average centroid distance ($\overline{\|\mathbf{v}\|}$) metric relative to the formal SFBS routines for feature selection, we next explore the concept of class separation within a feature space using techniques that do not involve the comparison of formal classification results as an intermediate step. It makes sense that a topological analysis of the class distribution within the feature space could provide further insight into a more ideal feature selection routine for this analysis. To explore this concept, we introduce the field of computational homology for use as a feature selection routine in the next chapter.

Chapter 6

Computational Homology Feature Selection

With the promising performance of the average distance between class centroid feature selection metric relative to the formal SFBS routine, we further explore the concept of class separation within a feature space using techniques that do not involve the comparison of formal classification results as an intermediate step. A topological analysis of the class distribution within the feature space could provide further insight into a more ideal feature selection routine for this analysis. To explore this concept, we introduce computational homology for use as a feature selection routine.

6.1 Computational Homology

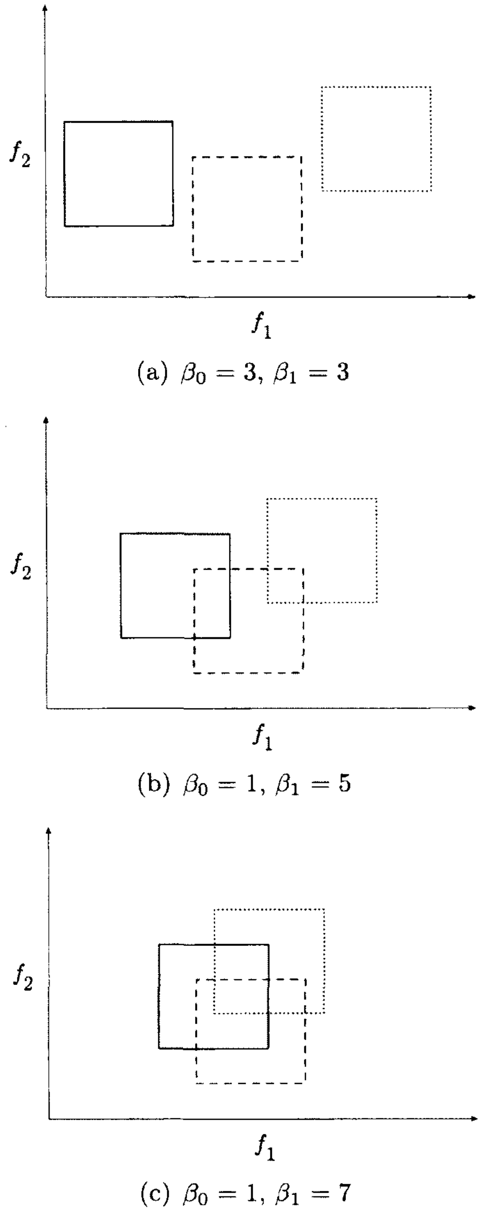
The application of computational homology as a feature selection tool continues from the work of Chapter 5, where several geometric measures on the class distribution within the feature space were used as feature selection metrics for ultrasonic inspection flaw classification. None of the feature selection metrics used previously were ideal in the sense that there exist cases for each method where the metric value is high yet the class distribution is poor. For example, using the average distance between class centroids is appropriate if the classes are linearly distributed within the feature space. If, however, a feature value increases for intermediate values and then decreases for higher values, one would end up with classes that 'turn back' on themselves and overlap significantly. The centroid distance may remain high between classes, but there is no measure of class overlap taken into account using this metric. Therefore, rather than simply calculate direction or magnitude of the vectors from class centroids within a given feature space, we wish to extract a value which compares a measure of how much the classes in the feature space overlap each other. We turn to the calculation of Betti numbers (β), where the nth Betti number is given by

$$\beta_n = \dim H_n. \tag{6.1}$$

Here H_n is the nth homology group of a topological space. The elements of H_n are equivalence classes of n-cycles that do not bound any $n+1$ chains, which is another way of characterizing n-dimensional holes. Therefore, for $n = 0, \ldots, d-1$, the nth Betti number of a d-dimensional space is a coarse measurement of the number of disconnected n-dimensional surfaces in that space. For example, β_0 measures the number of individual connected components, β_1 measures the number of circular 'loops' (two-dimensional holes), β_2 measure the number of spherical voids (three-dimensional holes), etc. effectively measuring the higher dimensional connectedness of a space. It is important to point out here that a d-dimensional space can have at most d nonzero Betti numbers, from $\beta_0 \ldots \beta_{d-1}$. For example, a space of dimension $d = 2$ cannot contain any 2-holes, or spherical voids, which require three dimensions to define. A formal introduction to computational homology can be found in Appendix B, which follows [158]. There, we formally define homology groups and their elementary properties. Specifically, we define topological spaces that are built out of d-dimensional unit cubes with vertices on the integer lattice, known as *cubical sets*. These will be the building blocks of our work.

A simple example of how Betti numbers correlate to the spatial distribution of classes within a feature space is presented in Figure 6.1. Here, three objects represented as squares are presented in a two-dimensional space for three different degrees of overlap: no overlap, slight overlap, and severe overlap. Each object can be thought of as a bounding box representation of an individual class. Because this space is two-dimensional, $\beta_n = 0$ for all values of $n > 1$. As the degree of object overlap increases, so does the number of two-dimensional holes (β_1), while the number of individual connected components (β_0) decreases. Similarly, as classes overlap increasingly within a feature space, they will be more difficult to distinguish between, reducing classification accuracy. This results in an inverse correlation between β_1 and classification accuracy. In a three-dimensional feature space, each class would be bounded by a cube. The number of cavities (three-dimensional holes) indicated by β_2 would then provide the relevant measure of class overlap. This concept can be extended to d-dimensional feature spaces, where each class is represented by a d-dimensional hypercube. In d-dimensions, it is the value of β_{d-1} that we expect to provide a relevant measure of class overlap within the feature space.

The mathematical tools we use here are not novel. Rather, we present a unique application of them in the context of feature selection. Several publicly available software packages exist for the computation of homologies and their Betti numbers, including CHomP [159], the GAP homology package [160], JavaPlex [161], Dionysus [162], and the RedHom library [163]. Several of these packages build topological spaces using simplical complexes, while others allow the use of cubical complexes. For



(a) $\beta_0 = 3$, $\beta_1 = 3$

(b) $\beta_0 = 1$, $\beta_1 = 5$

(c) $\beta_0 = 1$, $\beta_1 = 7$

Figure 6.1: Three examples of Betti numbers β_0 and β_1 for three objects in a two-dimensional feature space (f_1 vs. f_2). Each object can be thought of as a class centered at its mean vector and bounded by its standard deviation in each dimension. Betti number β_0 corresponds to the number of connected objects in the space, while β_1 corresponds to the number of 1-dimensional holes (enclosures) in the space. If the three objects are disjoint, then there exist three separate connected components, each of which is a 1-dimensional hole (a). As the objects begin to overlap, the number of connected objects decreases and the number of 1-dimensional holes increases (b). Further overlap again increases the number of 1-dimensional holes again (c).

a given class distribution in a d-dimensional feature space, each class can easily be centered at its centroid and bound in each dimension by its standard deviation. This results in a d-dimensional hypercube, which can be represented easily by a cubical complex. We therefore restrict our software choice to those that allow cubical complex inputs. Specifically, we use settled on software developed by the computational homology project (CHomP). The CHomP software used here applies several reduction algorithms that are combinatorial in nature, allowing for the rapid computation of homology groups based on input of an elementary cubical complexes. Specifically, the algorithm used here is written by Pawel Pilarczyk [159] and computes the homologies over the ring of integers.

6.2 Cubical Set Representation of Data

Before we can use the CHomP algorithms, we first need to transform our data set from clusters of points in the feature space into a form that can be read by the computational homology algorithms. The goal is to turn this data into a cubical set, where each class is represented by a hyper-rectangle embedded in the space $\mathbb{R}^d$. The first step in this transformation is to identify each class's centroid and its standard deviation in each dimension of a d-dimensional feature space. Each class can then be represented by a d-dimensional hyper-rectangle (referred to here as a d-rectangle), centered at the centroid and bound by the standard deviation in each dimension. Each d-rectangle is composed of $2 \times d$ individual $(d-1)$-dimensional faces. On this set of d-rectangles in a given feature space, we apply a homology-preserving map that transforms the bounds defined by the standard deviations onto an integer lattice. Next, each $2 \times d$ individual $(d-1)$-face is divided into a set of d-dimensional *elementary cubes*, which themselves are the finite product of d *elementary intervals*, one of which is degenerate and $d-1$ of which are non-degenerate. An example of this process for a two-dimensional feature space ($d = 2$) can be seen in Figure 6.2. The number of elementary cubes required to define each $(d-1)$-face depends on the size of that face within the integer lattice representation. In other words, if a transformed hyper-rectangle has a dimension larger than unit length, it is defined in terms of several smaller elementary components.

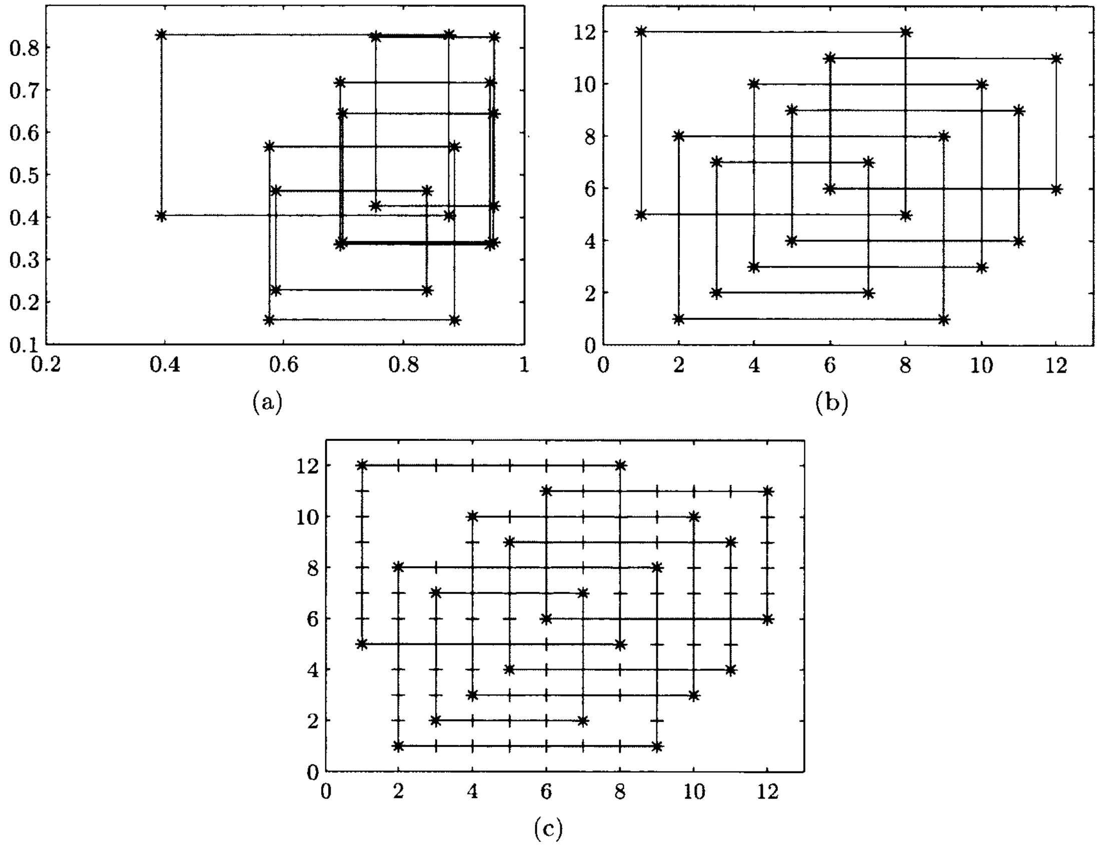


Figure 6.2: The first step in computing the homology is to establish the class bounds in the feature space (a). Then, a mapping is applied that maintains the homology of the space and sets these bounds to the integers $\mathbb{Z}$ (b). Finally, faces of each class are divided into elementary cubes of dimension $d-1$ (c).

To summarize, each d-rectangle is represented by a set of $2 \times d$ individual $(d-1)$-faces, each of which is composed of elementary cubes, where each elementary cube is defined as the Cartesian product of one degenerate and $d-1$ non-degenerate elementary intervals.

For each feature space, the set of elementary intervals defining the hyper-rectangles in $\mathbb{R}^d$ is written into a temporary file which is used as input for CHomP. The resulting output of Betti numbers is extracted from the CHomP program and saved in an array, along with the corresponding features that make up the given feature space. The class distribution in this feature space is then used to train and test the formal classification functions described previously, with the classifier accuracy saved in the same array. This process is repeated for every feature space subset of dimension d, building an array of feature space subsets and their corresponding classification accuracies and Betti numbers. Any correlation present between the selected feature space's classification accuracy and any trends present in the Betti numbers can be made from these values.

6.2.1 Computational Restrictions

There are computational restrictions that need to be taken into account before the CHomP algorithms are used. For the most part, the restrictions are based on the fact that each class has to be composed of elementary cubes, defined on the set of integers. It follows that a large number of classes results in a large number of elementary components required to build the resulting hyper-rectangles in the feature space. This number increases exponentially with dimension. For example, a 3-dimensional box has $2^3 = 8$ corner nodes, while a 15-dimensional hyper-rectangle has $2^{15} = 32768$ corner nodes. This exponentially increasing number of elementary cubes required to define a higher-dimensional feature space translates to an exponential increase in the memory required to define the space. Additionally, the number of elementary cubes required in each dimension depends on number of classes present, which defines the range in each dimension for the feature space. For example, a 6 class feature space will have 12 boundaries in each dimensions (an upper and lower bound per class). This further increases the memory required to define a class distribution as a cubical set. It is easy to see how the size of this problem increases drastically as the number of classes and dimensions increase.

We therefore explore a subset of the original problem instead of the entire original 14-class feature space. We consider a subset of 6 classes, taken from the original data set. From the original 100 waveforms per class, 50 were selected randomly to reduce the problem space further. The 6 classes were selected to provide a full spread of flaw thicknesses while still being a manageable set of data. A summary of these reduced classes can be found in Table 6.1

Table 6.1: The flaw thickness corresponding to the subset of double-crosshole scans used to explore the computational homology feature selection results. A Starrett digital micrometer was used to measure to thickness in three locations within the flaw, with the average value reported here. The percent thickness loss relative to the full plate thickness is also presented.

Data set	D_1	D_5	D_8	D_{10}	D_{12}	D_{14}
Average Flaw Depth (mm)	0.00	0.67	1.34	1.86	2.47	3.15
Percent Thickness Loss	0%	21%	43%	59%	78%	100%

Because we have reduced our data to 50 waveforms per class, the $N/10 > d$ expected 'optimal' feature set dimension estimation suggests that we can find a feature subset of dimension $d = 5$ or less with reasonable performance. Therefore, we explore all possible feature subsets of dimensions $d = 2, \ldots, 5$ here. To make the computation time reasonable, we must first reduce the original $d = 78$ feature set again, since $\binom{78}{5} > 21 \times 10^6$ possible combinations which, at 3 minutes per computation, would take over 120 years to compute on a single machine. Theoretically this could be reduced through a combination of serial and parallel processing techniques down to approximately 6 months of continuous runtime if the full power of the SciClone[1] computing cluster was reserved exclusively for this project. This is not a realistic request, however, so we reduce the size of our original feature set down to $d = 10$ features using the previously defined SFBS feature selection routine. With these top ten features, we calculate both the classification accuracy as well as the Betti numbers for the class distribution in that feature space for all possible feature subsets of dimension $d = 2, \ldots, 5$ in order analyze the relationship between accuracy and Betti numbers. Dimension $d = 1$ is not considered here because the only non-zero Betti number $\beta_0 = 1$ for all feature spaces of dimension $d = 1$, and no trends can be extracted from a constant value.

For each feature space, classification accuracy is determined in a method similar to before. Of the 300 waveforms in the data set (50 waveforms per class, 6 classes), 20% is randomly selected and withheld for testing, preserving class distributions in the original data set. That means that a proportional number of waveforms are randomly selected from each class compared to the original distribution. The remaining 80% of the data is used to train the classifier, using only the features identified by the feature space under consideration. Once trained, the testing set is submitted to the classifier, again using the features defined by the given feature space, and the resulting predicted labels are stored. Classification accuracy is determined to be the number of labels predicted to be within ±1 step of their actual label, presented here as a percentage of the total waveforms tested.

[1] `http://www.compsci.wm.edu/SciClone/`

6.3 Results

Initial Betti number results for the 3NN and QDC classifiers, previously identified as performing superior to the rest, can be seen in Figures 6.3 and 6.4. For each classifier, a series of plots is presented. Each row of plots corresponds to feature space dimension $d = 2, \ldots, 5$ from top to bottom, while each column corresponds to Betti numbers of dimension β_n where $n = 0, \ldots, d-1$ from left to right. Each subplot contains the Betti number of dimension n (y-axis) as a function of classifier accuracy (%) (x-axis) for each possible feature space combination of dimension d. We expect the number of d-dimensional holes present in each d-dimensional feature space, measured by β_{d-1}, to provide a measure of class separability as a linear relationship. Roughly speaking, the higher the number of n-holes, the more class overlap present in the feature space, and the lower the expected classification accuracy.

The first thing to note in these results is that the range of accuracies for the various feature spaces here are generally higher than the previous larger analysis due to the reduced number of classes (and overall dimension of the problem). By including only 6 classes, there is statistically higher chance of separation within the feature space.

Both classifiers return similar sets of results. With one exception, the values of $\beta_0 = 1$, indicating that there is always overlap between classes in the feature space. This is verification that a sophisticated analysis technique, like pattern classification, is required for analysis of this data in the first place. If the values of β_0 were consistently larger than 1, that would mean there are disjoint classes in the feature space, and a simple clustering analysis would be more appropriate.

For plots of β_{d-1} over all $d = 2, \ldots, 5$ (the left-hand subplot of each row), there seems to be an unusual pattern in the data. It appears that for these β_{d-1} plots, a linear fit may not have been the best choice to analyze the data, but rather a higher-

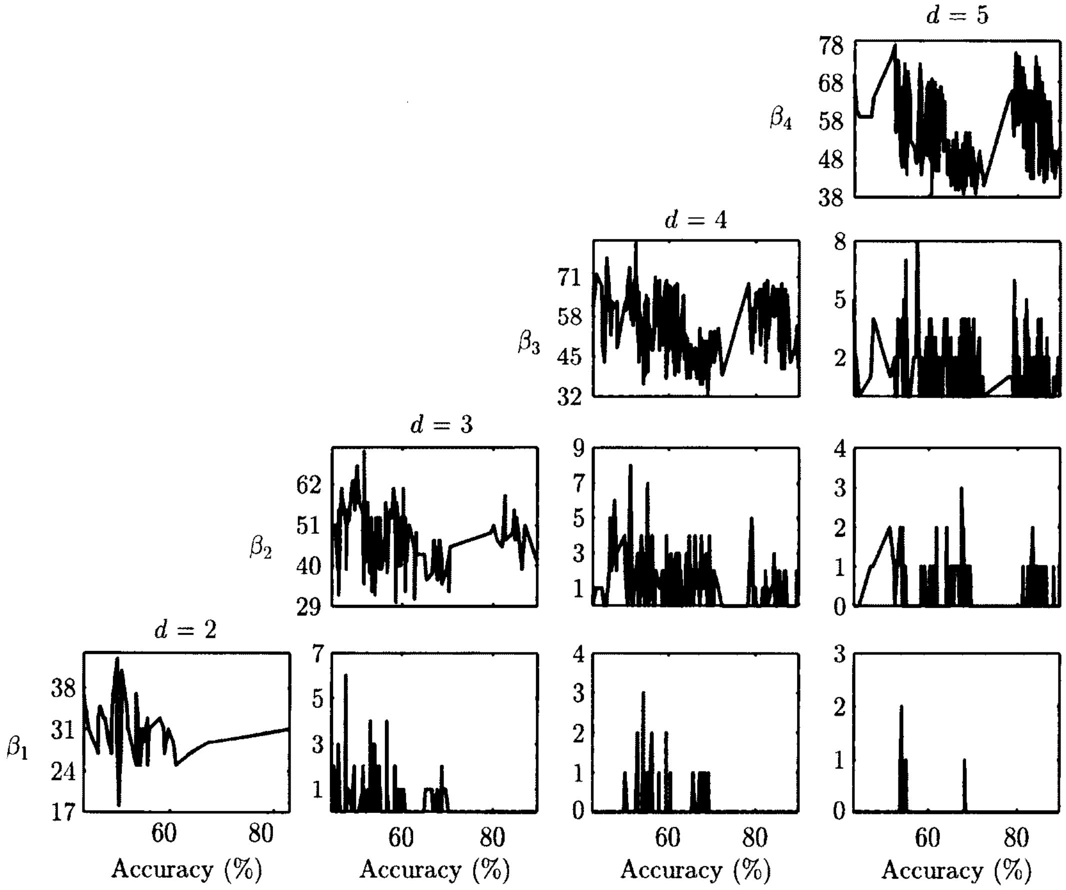


Figure 6.3: Betti number results for the QDC classifier. Each row corresponds to dimension d where $d = 2, 3, 4, 5$ from top to bottom, while the columns represent the Betti number β_n where $n = 0, \ldots, d-1$ from left to right. Each plot shows two lines: the blue line is the Betti number of dimension n (y-axis) as a function of classifier accuracy (%) (x-axis), while the black dotted line is a linear best-fit line for the Betti numbers provided for easy visualization of any relationship present between the classifier accuracy and the Betti numbers. There is a general trend of an increasing slope for the best-fit line, indicating a relationship between lower Betti number and higher classification accuracy.

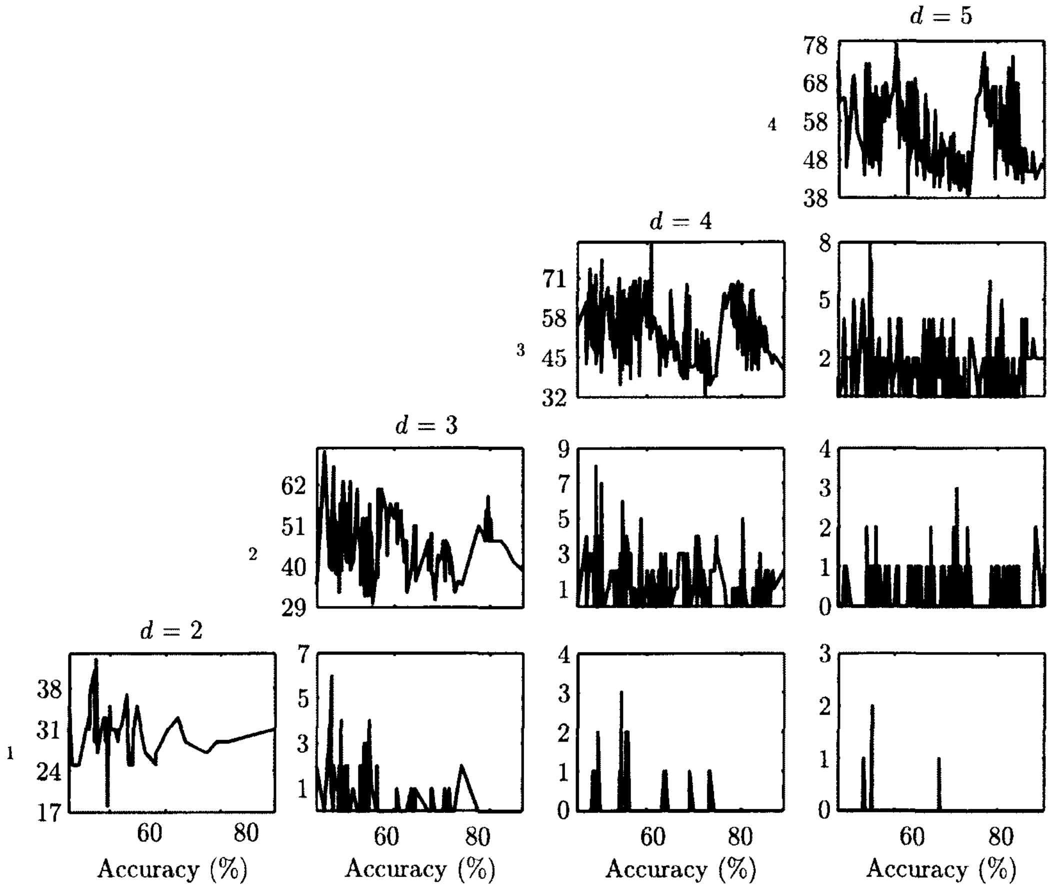


Figure 6.4: Betti number results for the 3NN classifier. Each row corresponds to dimension d where $d = 2, 3, 4, 5$ from top to bottom, while the columns represent the Betti number β_n where $n = 0, \ldots, d-1$ from left to right. Each plot shows two lines: the blue line is the Betti number of dimension n (y-axis) as a function of classifier accuracy (%) (x-axis), while the black dotted line is a linear best-fit line for the Betti numbers provided for easy visualization of any relationship present between the classifier accuracy and the Betti numbers. There is a general trend of an increasing slope for the best-fit line, indicating a relationship between lower Betti number and higher classification accuracy.

order polynomial may fit the data better. However, closer inspection reveals that this is not exactly the case. In both the 3NN and the QDC results, the classification accuracies are split into a group of higher accuracies with a slight gap to a second group of lower accuracies. This splits the Betti numbers into two disjoint sections. These sections, when isolated, display the general increase in Betti number relationship we were expecting to find. An example of this is shown in Figure 6.5, where the subplot of β_{d-1} for $d = 5$ from 6.3 is enlarged. The two disjoint sections of data are highlighted, with bounds of each individual cluster of feature spaces indicated by vertical lines. The gap in classification accuracy from just above 70% to roughly 80% can be seen easily here. Individual linear best-fit lines are included to highlight the general trend of the data. The linear fit for the feature spaces in the lower range of accuracies, bounded by the dotted lines, has a best-fit slope of -0.84. Similarly, the linear fit for the feature spaces in the higher range of accuracies, bounded by the dashed lines, has a best-fit slope of -1.61. These both indicate the negative correlation between Betti number β_{d-1} and classification accuracy we hoped to find.

The cause of this gap in classification accuracies is unknown, but it may be related to the concept of nested classes in the feature space being topologically identical to disjoint classes. In other words, only classes which overlap each other contribute to the increase in n-holes and therefore Betti number β_{n-1}. Disjoint *and* nested classes do not. If a class is nested within the bounds of a separate class, it follows that it would be difficult to distinguish between the two classes using formal classification techniques. Nested classes therefore result in a lower classification accuracy while maintaining an identical topological makeup when compared to similar disjoint classes.

If we could identify the specific feature spaces that correspond to the higher of the two sections only, we could then sort these feature spaces by β_{d-1} and have generally higher-to-lower classification accuracies from these results. To further explore this

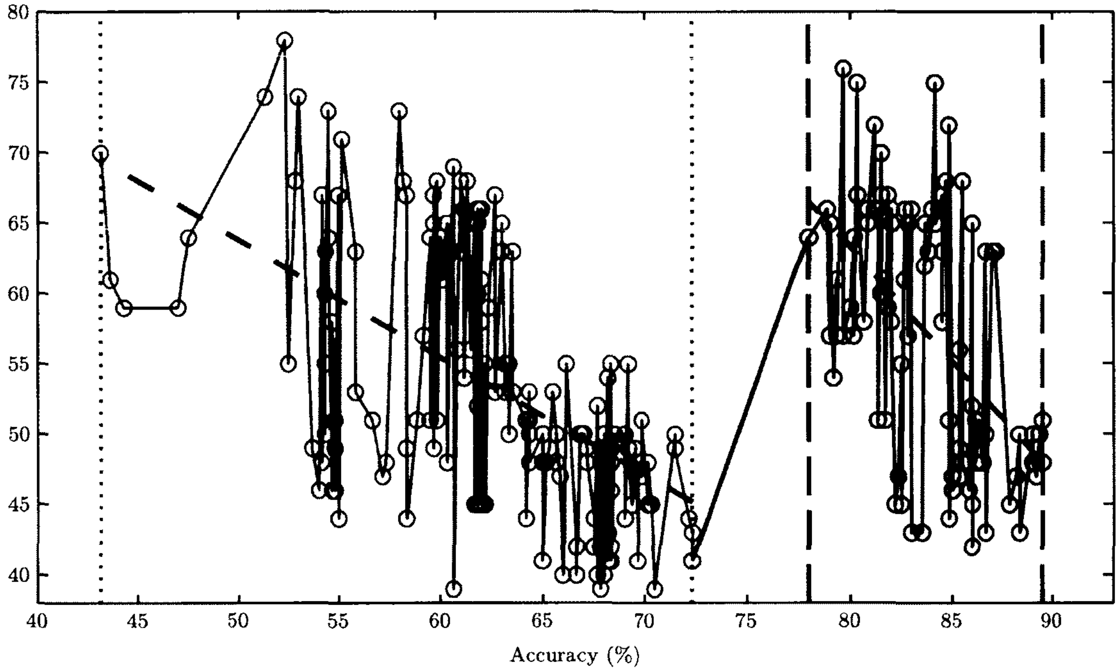


Figure 6.5: A plot of β_{d-1} for $d = 5$ as a function of QDC classification accuracy for all feature space subsets of dimension $d = 5$. There is a gap in the classification accuracies dividing the feature space subsets into a group of lower accuracy feature spaces and a second distinct group of higher accuracy feature spaces. These two sections are bounded here by dotted and dash-dot vertical lines, receptively. Individual linear best-fits lines are applied to each group, shown here as dashed lines. The linear fit for the lower range of accuracies has a best-fit slope of -0.84. Similarly, the linear fit for the higher range of accuracies has a best-fit slope of -1.61. These both indicate the negative correlation between Betti number and classification accuracy we expected to find.

concept, we will investigate utilizing both computational homology to explore topological features of each feature space as well as geometric measurements of the class distribution from within each feature space, as was explored in Chapter 5. That is, we will combine the tortuosity, average angle, and average centroid distance feature selection metrics with a fourth, Betti number-based metric. By combining methods, we will hopefully be able to account for variations within each individual metric method that result in lowered classification accuracies.

To begin exploring this concept, we first calculate the geometric measurements (tortuosity between class centroids, average angle between class centroids, and average distance between class centroids) for this reduced data set for all possible feature spaces of dimension $d = 2, \ldots, 5$ from the top 10 SFBS features previously discussed. We then match these feature spaces up with their corresponding Betti numbers and classification accuracy.

It should be noted here that we can perform this analysis because we have considered *all* possible subsets of dimension $d = 2, \ldots, 5$ in our results. Previously, with the larger 14-class problem, we were only considering the top feature spaces as determined by each metric for comparison of classification accuracy due to the large size of the problem at hand. Attempting to compare those limited results in this manner would not be complete, and would not be reliable in determining general trends in the data as we are here.

We restrict ourselves to using the QDC classifier for this analysis because of how similar its Betti number results were compared to the 3NN classifier, as well as how consistent the QDC classifier performance was throughout the previous chapter. For each feature space dimension $d = 2, \ldots, 5$ we first plot the three sets of geometric measurement values against their QDC classification accuracy. A threshold is applied for each metric that allows inclusion of the highest feature space classification accuracies but removes as many of the lower classification accuracy feature spaces as possible. A formal threshold analysis to determine ideal threshold levels is not performed in this study. Instead, thresholds are selected based only on the data at hand in an attempt to provide a proof-of-concept for this technique. The Betti numbers β_n for $n = 0, \ldots, d-1$ are also plotted against classification accuracy, and a similar threshold is applied for β_n of $n < d-1$. Finally, β_{d-1} is plotted against classification accuracy, and feature spaces that have met all threshold values are highlighted. Ideally, only those corresponding to the higher-accuracy 'section' described above remain, which can then be sorted by β_{d-1} with higher accuracies corresponding to lower β_{d-1}.

These results are presented in Figures 6.6 - 6.8 for $d = 2$, Figures 6.9 - 6.11 for $d = 3$, Figures 6.12 - 6.14 for $d = 4$, and Figures 6.15 - 6.17 for $d = 5$. After an initial analysis of the data, threshold values were selected for each metric. For the three geometric measures (tortuosity, avg. angle, and avg. centroid distance), a threshold level was chosen that was consistent across all dimensions. For the Betti number results, since the Betti numbers are dimensionally dependent, individual thresholds were selected. For each of these sets of figures, the first figure shows the three geometric measures plotted against their QDC accuracy, with applied thresholds represented by black dotted lines and any feature spaces that meet each threshold highlighted by a black star. Any feature space without a black star is removed from consideration. A plot of β_{d-1} is provided, where any feature spaces which meet *all three* of the geometric thresholds are highlighted by a red star. Next, a similar plot is given for Betti numbers β_n where $n = 0, \ldots, d-1$. For each β_n where $n < d-1$, a threshold is applied and all feature spaces that meet each threshold are highlighted by black stars. A plot of β_{d-1} is provided, where any feature spaces which meet *all* of the lower dimensional Betti number thresholds are highlighted by a red star. Finally, a plot is given of β_{d-1} versus classification accuracy, where any feature space that has not been removed by either the combined geometric thresholding or the Betti number thresholding is highlighted by a red star.

For the geometric results, we have chosen threshold values that are consistent across all dimensions. For the tortuosity metric, the values are limited to $\tau < 4.7$. Average angle between centroid values are limited to $[77^\circ < \bar{\theta} < 107^\circ]$. Finally, average distance between class centroid values are limited to $\overline{\|\mathbf{v}\|} > 0.12$. There are several trends in these results that are consistent across all dimensions. For the tortuosity metric, the group of feature spaces that have the highest accuracy, above any gap in accuracy that may be present, are all clustered together below the $\tau = 4.7$ limit. Those corresponding to lower classification accuracies tend to produce significantly higher values of τ, which are mostly filtered out with the applied threshold. For each dimension, there still exist several lower accuracy feature space points that also meet the threshold limit. These results indicate that the tortuosity metric is a promising technique for identifying feature spaces that result in high classification accuracy, however alone it is not perfect.

The average angle between centroid results also have a trend that is consistent across all dimensions. Most of the feature spaces corresponding to high classification accuracy are clustered within the $[77^\circ < \bar{\theta} < 107^\circ]$ range we have identified. Higher-dimensional feature spaces show a second cluster of average angle values up around 130°, however we have chosen not to include this range of angles because it tends to be the lower-end of the highest accuracy cluster in addition to many of the lower accuracy points as well. Because we can still identify the highest accuracy values in all dimensions, we believe this metric also shows potential for use in parallel with other metrics for removing the lower accuracy feature spaces from consideration.

The average distance between class centroids ($\overline{\|\mathbf{v}\|}$) have less of a general trend in the data that allows us to separate the higher accuracies from the lower. We

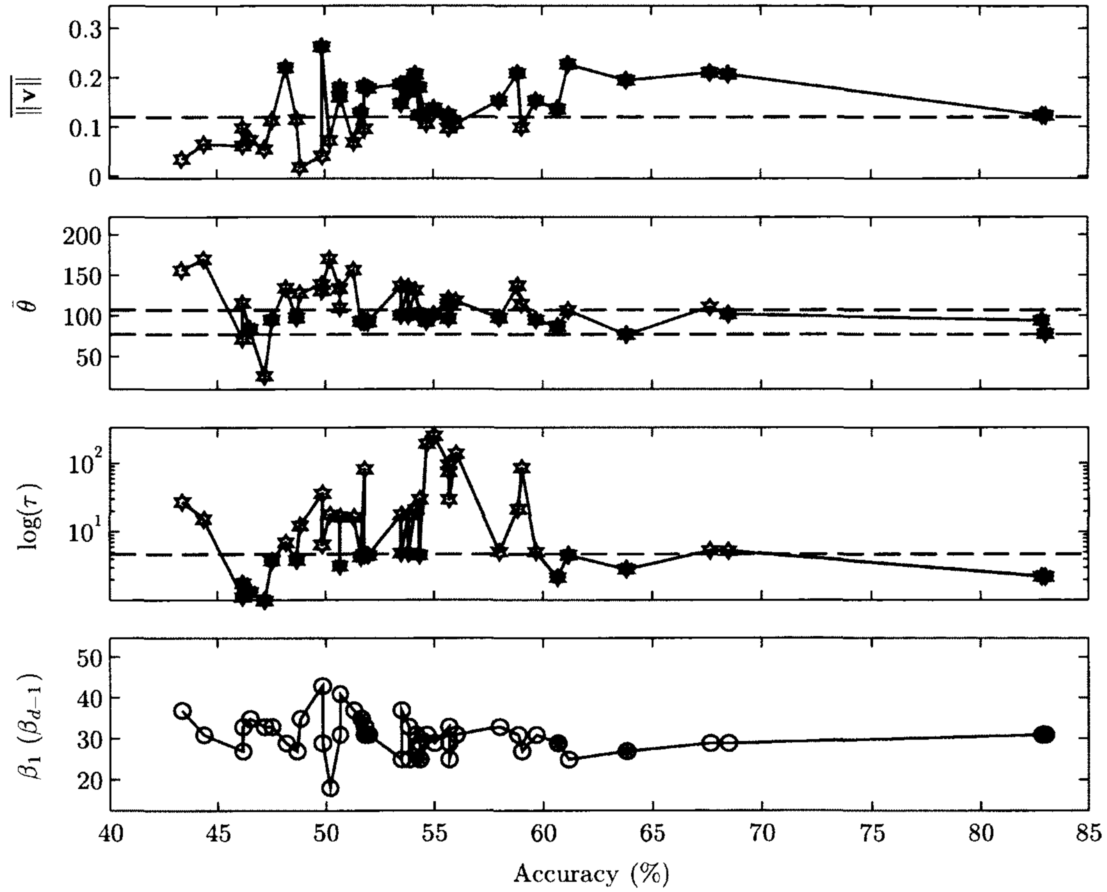


Figure 6.6: Values of the three geometric feature space values versus QDC classification accuracy (% ±1 flaw depth) for all feature spaces subsets of dimensions $d = 2$. Average distance between class centroids ($\overline{\|\mathbf{v}\|}$, top), average angle between class centroids ($\bar{\theta}$, upper-middle), and tortuosity relative to class centroids (τ, lower-middle) are shown. Thresholds represented by the black dotted lines have been applied at $\overline{\|\mathbf{v}\|} > 0.12$, $[77° < \bar{\theta} < 107°]$, and $\tau < 4.7$, respectively, to isolate data corresponding to the highest accuracies. Feature spaces which meet each of thresholds are highlighted by black stars (∗), while those which meet *all three* are highlighted in the plot of β_{d-1} (bottom) as red stars (∗). It can be seen that the majority of feature spaces identified by this method correspond to higher classification accuracies, however several of the feature spaces corresponding to lower classification accuracies are also included.

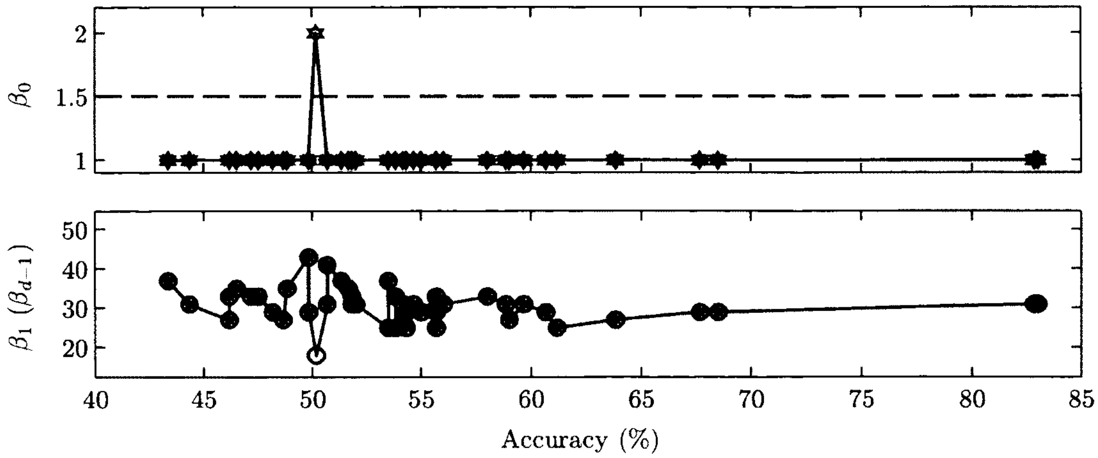


Figure 6.7: Betti numbers β_n for $n = 0, 1$ plotted against QDC classifier accuracy (% ±1 flaw depth) for all possible feature spaces of dimensions $d = 2$. A threshold has been applied for β_n for $n < d - 1$, shown as a black dotted line in each respective subplot. Feature space combinations that meet this threshold are indicated by black stars (*), while those which meet all thresholds are highlighted in the plot of β_{d-1}. It can be seen that only a single feature space was removed using this technique.

had expected the higher values of $\overline{||\mathbf{v}||}$ to correlate to better class separation, so we apply a lower-bound threshold here. The spread of high-accuracy $\overline{||\mathbf{v}||}$ values for the higher-dimensional cases is similar to that of the low-accuracy values, restricting the usefulness of this data set. Lower-dimensional spaces, however, do exhibit a lower $\overline{||\mathbf{v}||}$ lower accuracy relationship that we can exploit the application of a threshold to identify these lower accuracy points, so we include this metric in our analysis.

The Betti numbers values are unique to each dimensional space, as the number of nonzero β_n is related to d by $\beta_n \neq 0$ for $n \leq d - 1$. Therefore, we determine the threshold values for each dimension on an individual basis. The same principle applies, however, in that we are looking for a straightforward amplitude threshold to apply which separates as many of the higher accuracy feature spaces from the lower accuracy ones. We apply these thresholds only to β_n for $n < d - 1$, since we have already seen that β_{d-1} displays behavior that prevents us from using a simple selection criterion such as a threshold. For $d = 2$, we see that $\beta_0 = 1$ for all but a single feature

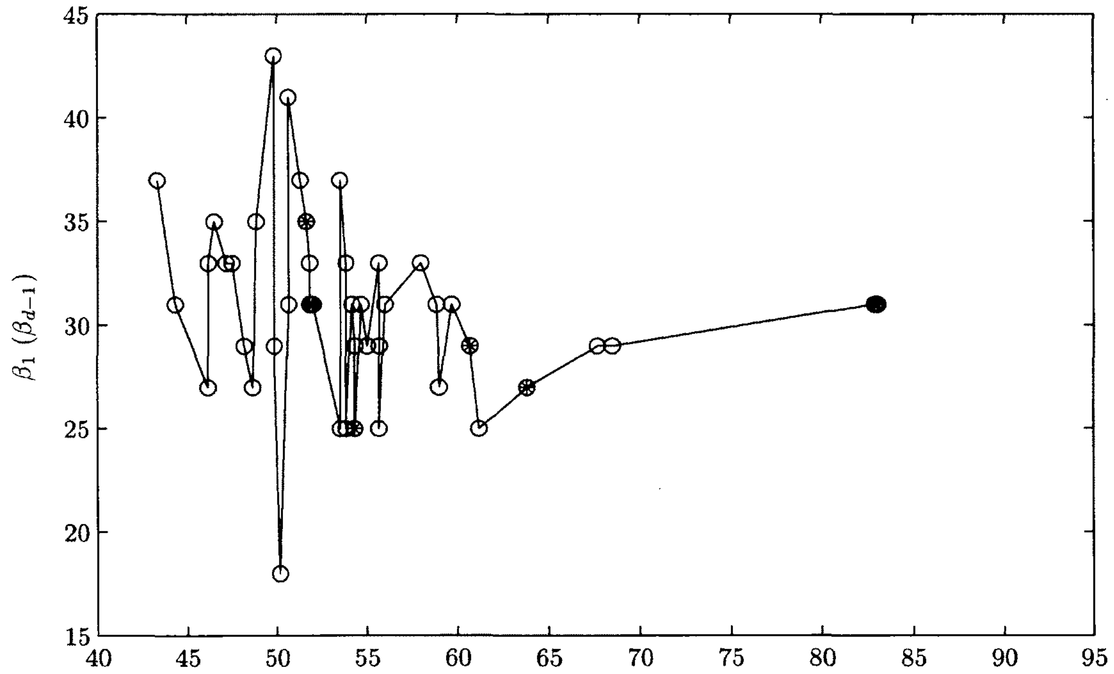


Figure 6.8: A plot of β_{d-1} for $d = 2$ sorted by classification accuracy (% ±1 flaw depth), where each possible feature subspace of dimension $d = 2$ represented by a blue circle. The average centroid distance, average angle, and tortuosity. thresholding techniques seen in Figure 6.6 as well as the Betti number threshold technique seen in Figure 6.7 are both applied here, with only those feature space subsets which meet all of these threshold values highlighted by red starts. It can be seen that using these thresholds allows us to identify the feature space combinations that produce the highest classification accuracies, as well as several corresponding to lower accuracies.

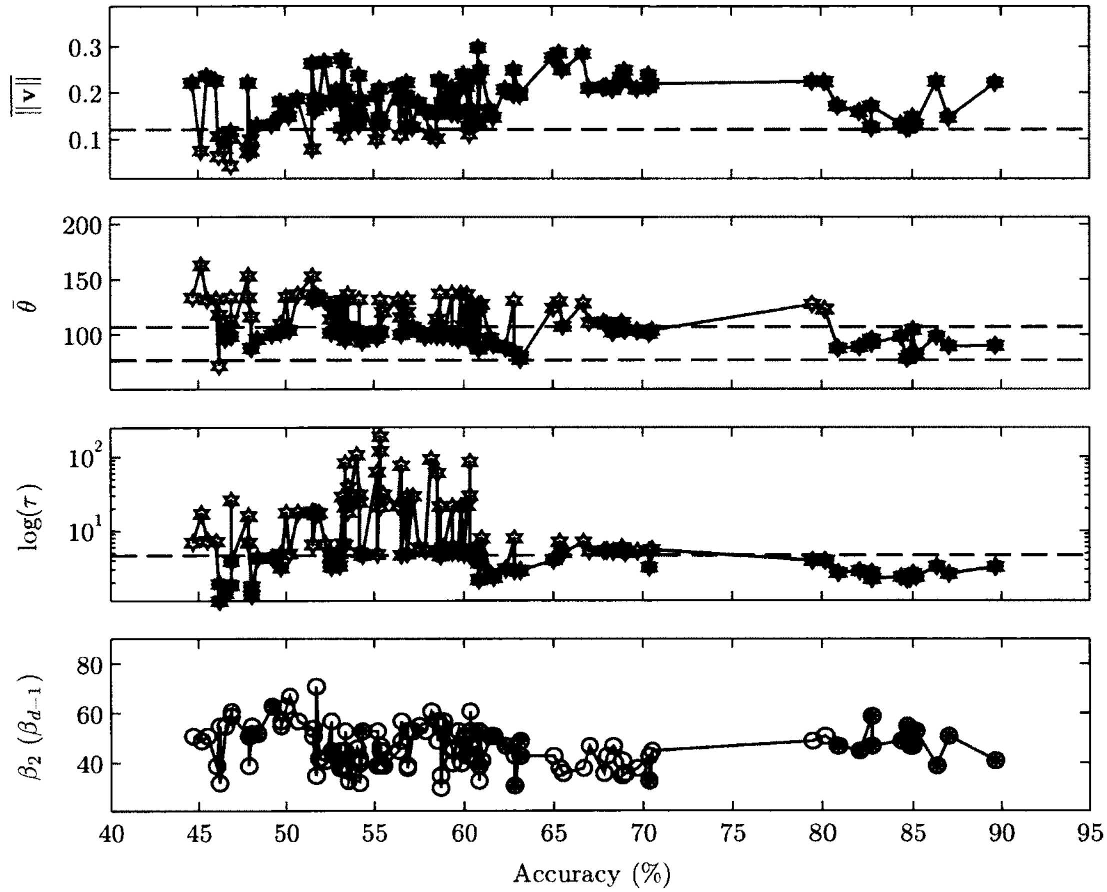


Figure 6.9: Values of the three geometric feature space values versus QDC classification accuracy (% ±1 flaw depth) for all feature spaces subsets of dimensions $d = 3$. Average distance between class centroids ($\overline{\|\mathbf{v}\|}$, top), average angle between class centroids ($\bar{\theta}$, upper-middle), and tortuosity relative to class centroids (τ, lower-middle) are shown. Thresholds represented by the black dotted lines have been applied at $\overline{\|\mathbf{v}\|} > 0.12$, $[77° < \bar{\theta} < 107°]$, and $\tau < 4.7$, respectively, to isolate data corresponding to the highest accuracies. Feature spaces which meet each of thresholds are highlighted by black stars (∗), while those which meet *all three* are highlighted in the plot of β_{d-1} (bottom) as red stars (∗). It can be seen that the majority of feature spaces identified by this method correspond to higher classification accuracies, however several of the feature spaces corresponding to lower classification accuracies are also included.

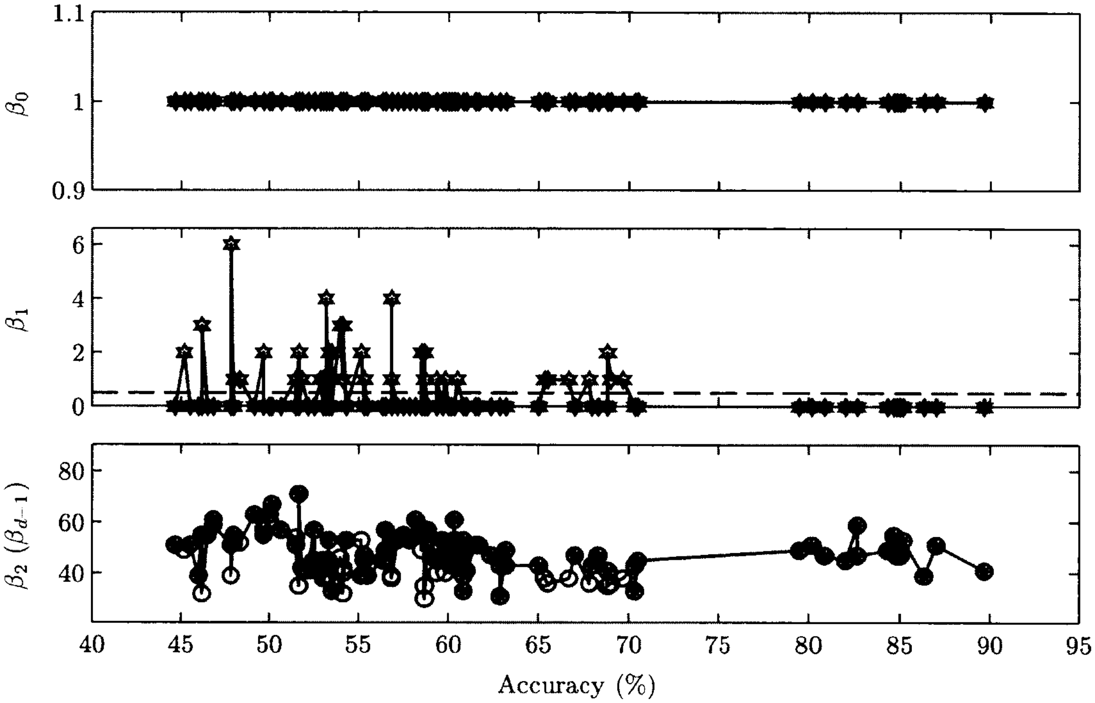


Figure 6.10: Betti numbers β_n for $n = 0, \ldots, 2$ plotted against QDC classifier accuracy (% ±1 flaw depth) for all possible feature spaces of dimensions $d = 3$. A threshold has been applied for β_n for $n < d-1$, shown as a black dotted line in each respective subplot. Feature space combinations that meet this threshold are indicated by black stars ($*$), while those which meet all thresholds are highlighted in the plot of β_{d-1}. It can be seen that only a single feature space was removed using this technique.

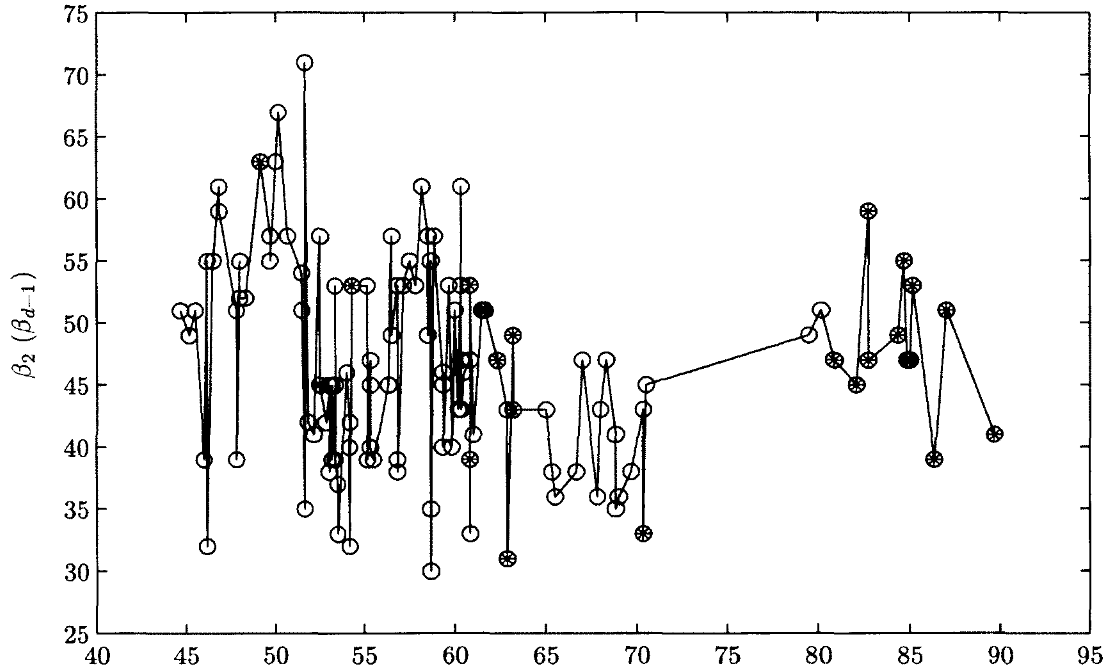


Figure 6.11: A plot of β_{d-1} for $d = 3$ sorted by classification accuracy (% ±1 flaw depth), where each possible feature subspace of dimension $d = 3$ represented by a blue circle. The average centroid distance, average angle, and tortuosity thresholding techniques seen in Figure 6.9 as well as the Betti number threshold technique seen in Figure 6.10 are both applied here, with only those feature space subsets which meet all of these threshold values highlighted by red starts. It can be seen that using these thresholds allows us to identify the feature space combinations that produce the highest classification accuracies, as well as several corresponding to lower accuracies.

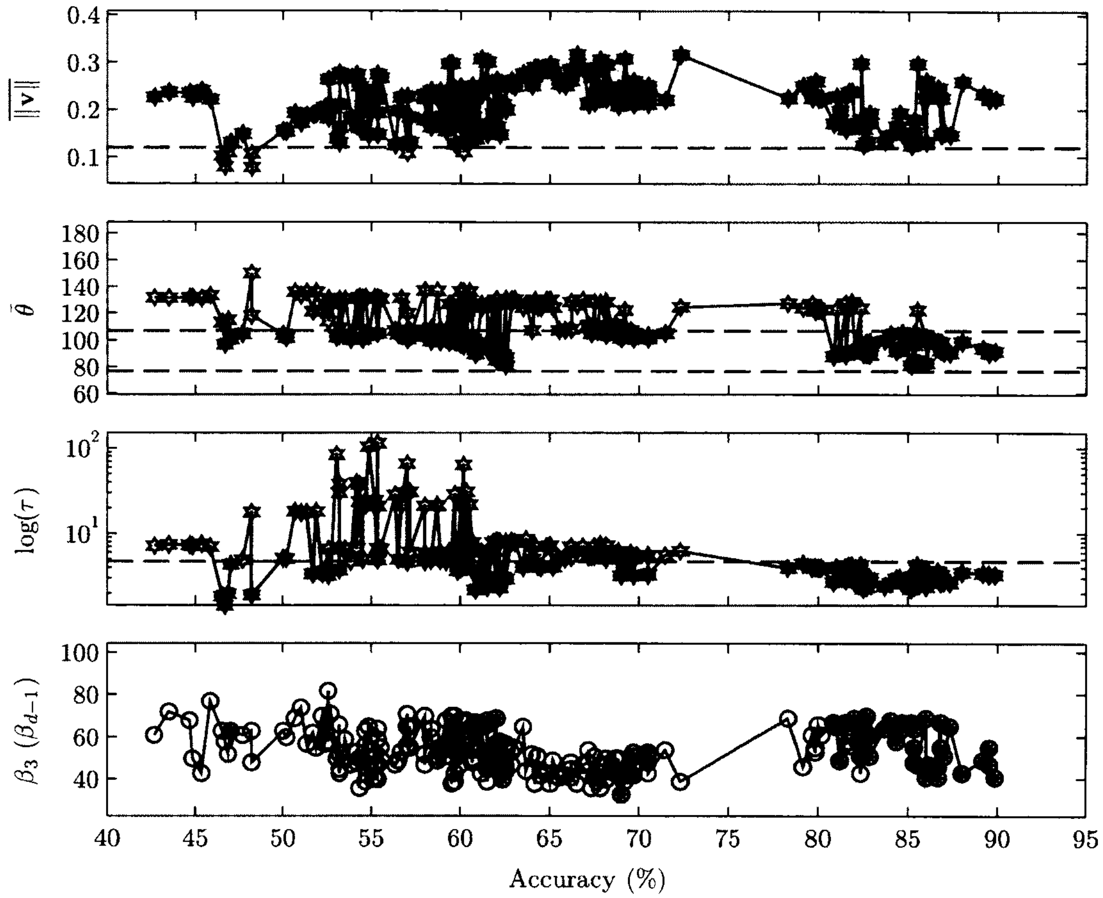


Figure 6.12: Values of the three geometric feature space values versus QDC classification accuracy (% ±1 flaw depth) for all feature spaces subsets of dimensions $d = 4$. Average distance between class centroids ($\overline{\|\mathbf{v}\|}$, top), average angle between class centroids ($\bar{\theta}$, upper-middle), and tortuosity relative to class centroids (τ, lower-middle) are shown. Thresholds represented by the black dotted lines have been applied at $\overline{\|\mathbf{v}\|} > 0.12$, $[77° < \bar{\theta} < 107°]$, and $\tau < 4.7$, respectively, to isolate data corresponding to the highest accuracies. Feature spaces which meet each of thresholds are highlighted by black stars ($*$), while those which meet *all three* are highlighted in the plot of β_{d-1} (bottom) as red stars ($*$). It can be seen that the majority of feature spaces identified by this method correspond to higher classification accuracies, however several of the feature spaces corresponding to lower classification accuracies are also included.

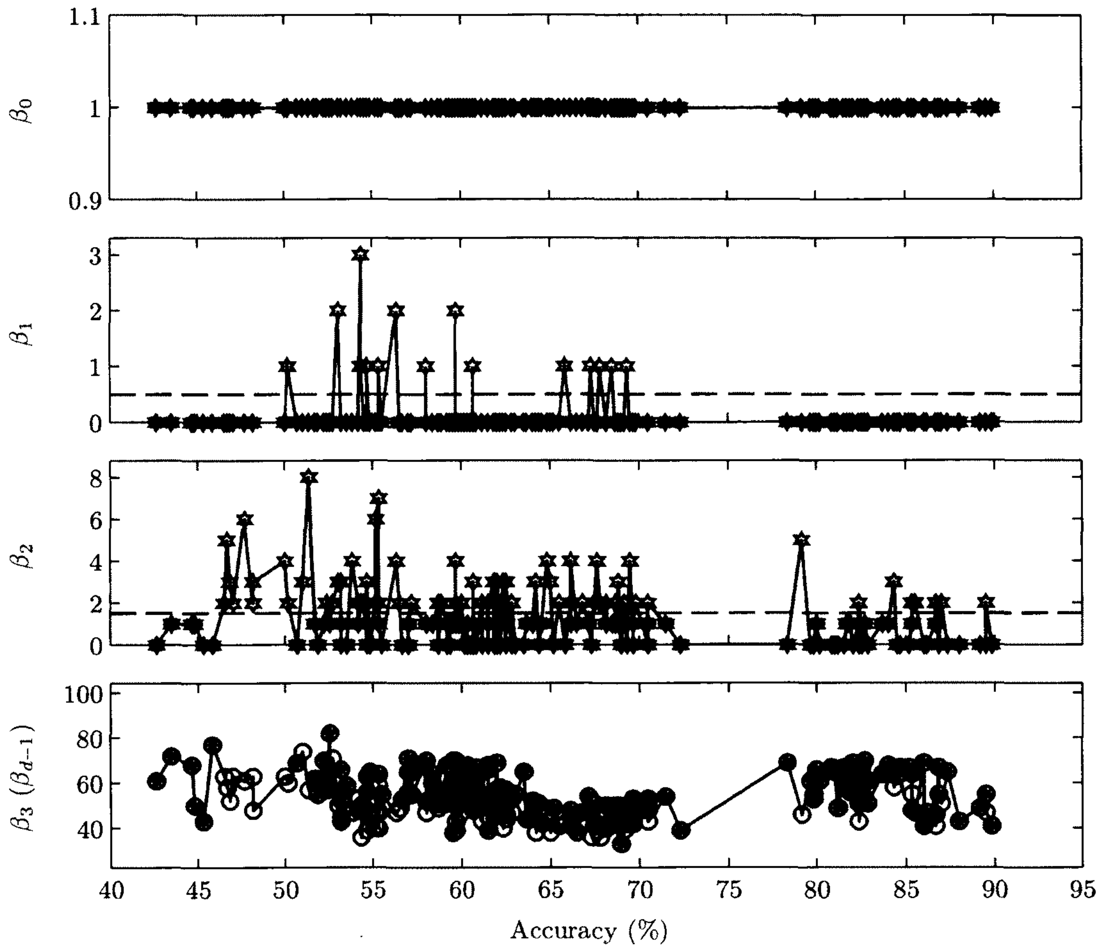


Figure 6.13: Betti numbers β_n for $n = 0, \ldots, 3$ plotted against QDC classifier accuracy (% ±1 flaw depth) for all possible feature spaces of dimensions $d = 4$. A threshold has been applied for β_n for $n < d-1$, shown as a black dotted line in each respective subplot. Feature space combinations that meet this threshold are indicated by black stars (*), while those which meet all thresholds are highlighted in the plot of β_{d-1}. It can be seen that only a single feature space was removed using this technique.

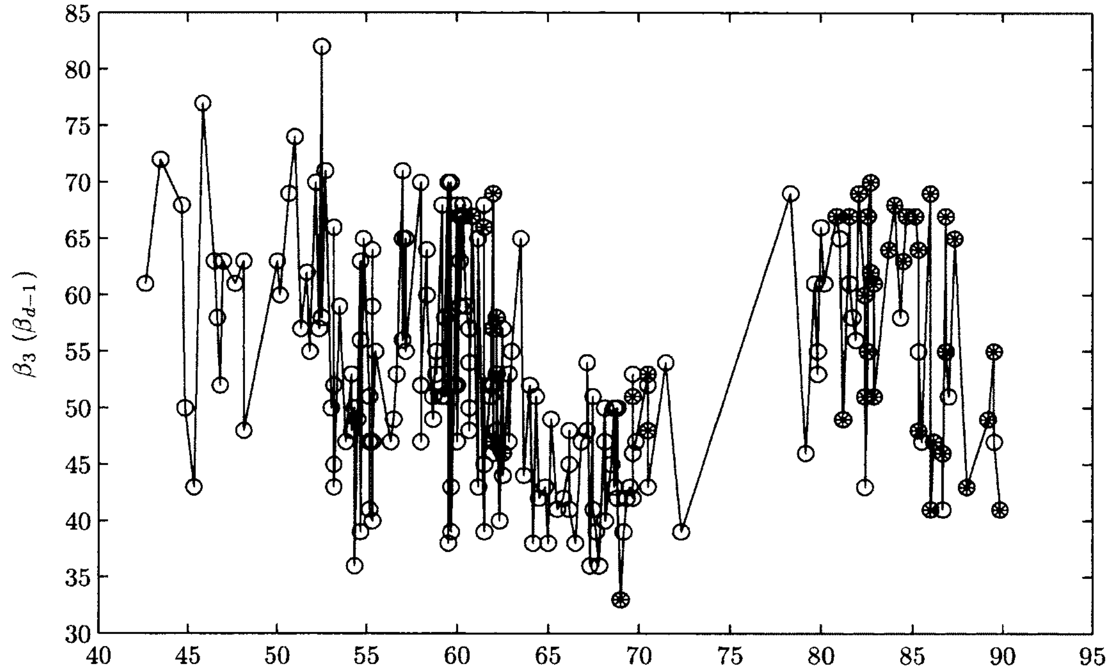


Figure 6.14: A plot of β_{d-1} for $d = 4$ sorted by classification accuracy (% ±1 flaw depth), where each possible feature subspace of dimension $d = 4$ represented by a blue circle. The average centroid distance, average angle, and tortuosity thresholding techniques seen in Figure 6.12 as well as the Betti number threshold technique seen in Figure 6.13 are both applied here, with only those feature space subsets which meet all of these threshold values highlighted by red starts. It can be seen that using these thresholds allows us to identify the feature space combinations that produce the highest classification accuracies, as well as several corresponding to lower accuracies.

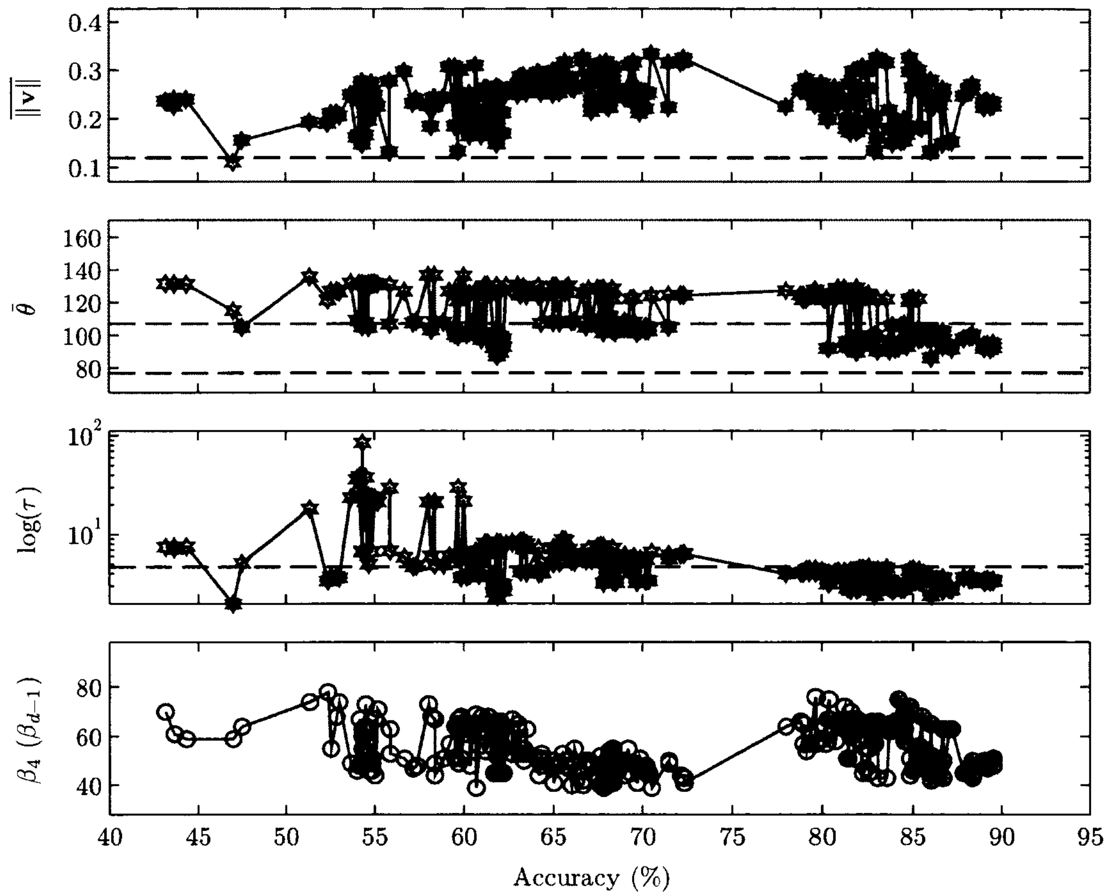


Figure 6.15: Values of the three geometric feature space values versus QDC classification accuracy (% ±1 flaw depth) for all feature spaces subsets of dimensions $d = 5$. Average distance between class centroids ($\|\mathbf{v}\|$, top), average angle between class centroids ($\bar{\theta}$, upper-middle), and tortuosity relative to class centroids (τ, lower-middle) are shown. Thresholds represented by the black dotted lines have been applied at $\|\mathbf{v}\| > 0.12$, $[77° < \bar{\theta} < 107°]$, and $\tau < 4.7$, respectively, to isolate data corresponding to the highest accuracies. Feature spaces which meet each of thresholds are highlighted by black stars ($*$), while those which meet *all three* are highlighted in the plot of β_{d-1} (bottom) as red stars ($*$). It can be seen that the majority of feature spaces identified by this method correspond to higher classification accuracies, however several of the feature spaces corresponding to lower classification accuracies are also included.

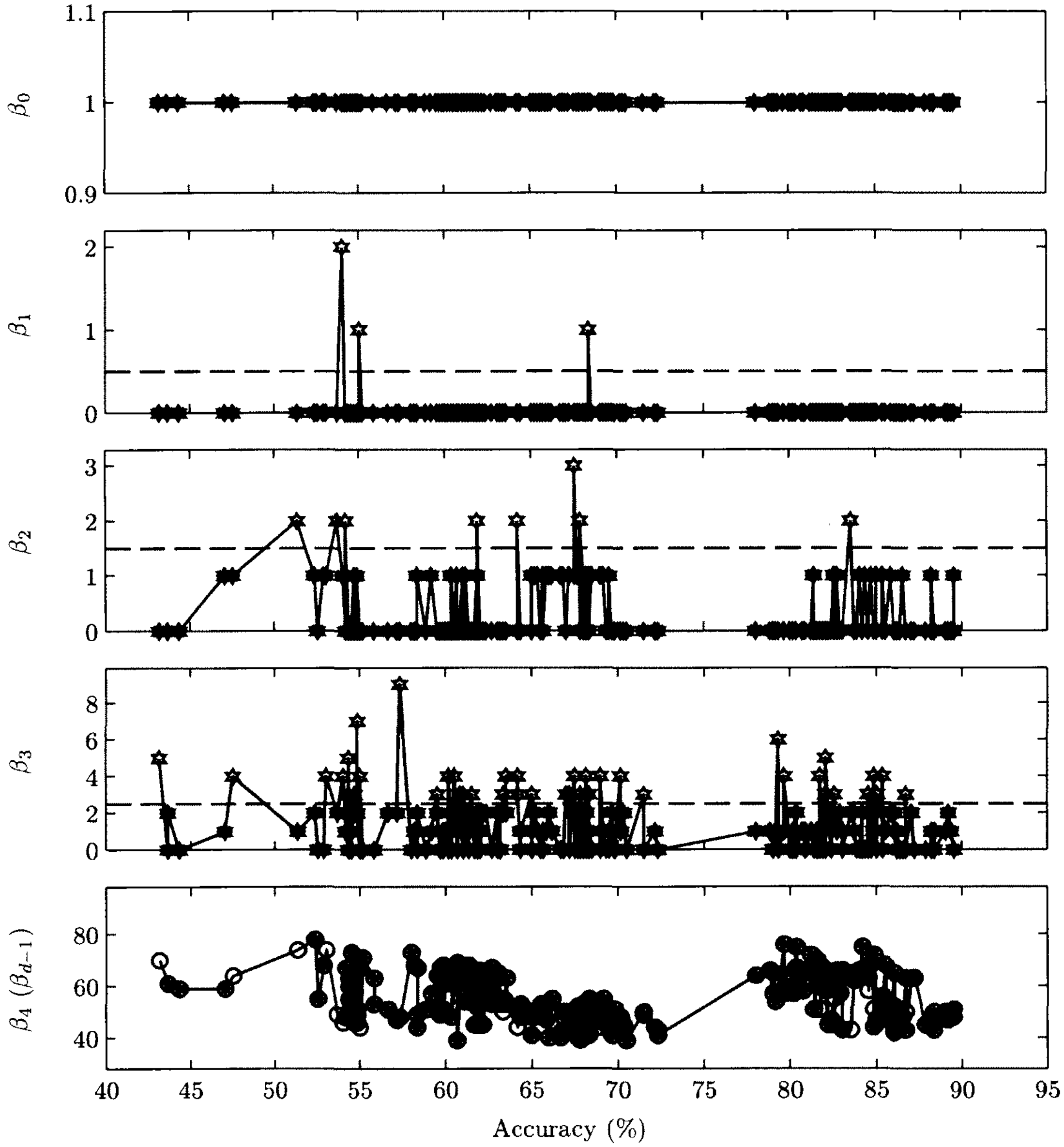


Figure 6.16: Betti numbers β_n for $n = 0, \ldots, 4$ plotted against QDC classifier accuracy (% ±1 flaw depth) for all possible feature spaces of dimensions $d = 5$. A threshold has been applied for β_n for $n < d-1$, shown as a black dotted line in each respective subplot. Feature space combinations that meet this threshold are indicated by black stars ($*$), while those which meet all thresholds are highlighted in the plot of β_{d-1}. It can be seen that only a single feature space was removed using this technique.

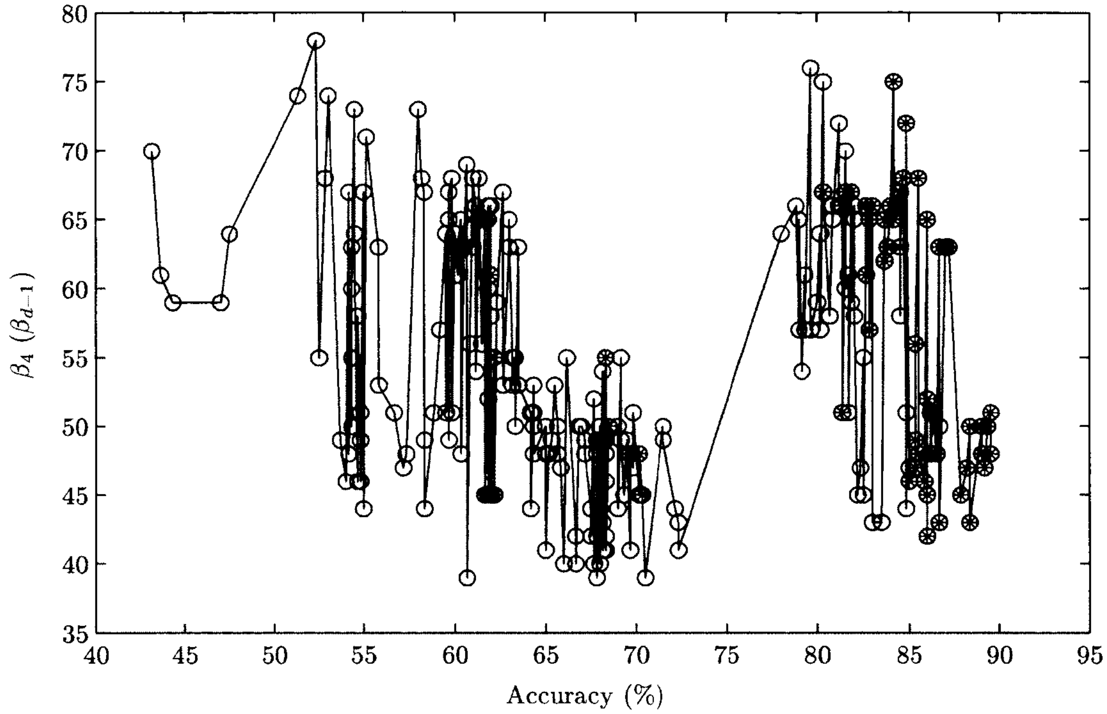


Figure 6.17: A plot of β_{d-1} for $d = 5$ sorted by classification accuracy (% ±1 flaw depth), where each possible feature subspace of dimension $d = 5$ represented by a blue circle. The average centroid distance, average angle, and tortuosity thresholding techniques seen in Figure 6.15 as well as the Betti number threshold technique seen in Figure 6.16 are both applied here, with only those feature space subsets which meet all of these threshold values highlighted by red starts. It can be seen that using these thresholds allows us to identify the feature space combinations that produce the highest classification accuracies, as well as several corresponding to lower accuracies.

space. We apply a threshold of $\beta_0 \leq 1$ anyway, which removes only a single feature space as seen in Figure 6.7. For $d = 3$, we again have that $\beta_0 = 1$ and therefore no room to apply a threshold. The values of β_1, however, allow us to identify several of the low-accuracy feature spaces by using a threshold limit of $\beta_1 < 0.5$. Since $\beta_n \in \mathbb{Z}$ for all n, this selects only values of $\beta_1 = 0$. This is shown to remove a variety of the low-accuracy feature spaces from consideration, as seen in Figure 6.10.

For $d = 4$, a similar trend is seen. No threshold can be applied to the constant β_0, however thresholds are applied to β_1 and β_2. It can be seen that the threshold applied for β_2 removes a small number of the higher accuracy feature spaces. At this dimension, there is a trade-off between how many lower accuracy feature spaces we remove and how many higher accuracy spaces we keep. It can be seen that if we were to lower the threshold limit to include all the higher accuracy spaces, we would include a much larger number of lower accuracy spaces as well. We have chosen a threshold value that only removes a small number of higher accuracy spaces here, as seen in Figure 6.13.

We again see similar trends for $d = 5$. We again have that $\beta_0 = 1$ for all feature spaces, and thus no threshold can be applied. The values of β_1 are approaching a similar spread, as only a couple have values above 0. We are also again faced with a trade-off between the number of high accuracy spaces to include versus the number of low accuracy spaces to remove for β_2 and β_3. It can be seen that we have chosen threshold values that again remove a small number of the higher accuracy feature spaces in an attempt to remove a larger number of lower accuracy spaces, seen in Figure 6.16.

These threshold results are summed up in the final figure for each dimension, Figure 6.8 for $d = 2$, Figure 6.11 for $d = 3$, Figure 6.14 for $d = 4$, and Figure 6.17 for $d = 5$. Each figure shows the plot of β_{d-1}, with all feature spaces that met all the previous thresholds highlighted by red stars. It can be seen that for each dimension, the feature spaces corresponding to the highest classification accuracies are identified by red stars. In addition, a smaller number of the lower-dimensional spaces make the cut as well.

6.4 Discussion

We have presented a feature selection technique using computational homology for multi-mode Lamb wave damage characterization. We utilize Lamb wave tomography to generate reconstruction images of each plate sample, automatically identifying and sizing the flaws at each depth. Waveforms which pass through the flaw region are automatically identified and extracted. Features are generated from these waveforms using a variety of signal analysis techniques including the DWFP technique. We then explore a feature selection technique that uses topological measures of the class distribution within the feature space to identify those most appropriate for Lamb wave flaw severity analysis.

We have created a filter-type feature selection routine that is able to identify subsets of a larger feature space that correlate with the highest possible classification accuracies for a several subset dimensions. We have approached the problem of the sequential ordering associated with a physical process that is a function of time, such as progressive damage in materials. To do this, we have combined measurements of the class distribution within each potential feature space with the topological connectedness of that distribution. We apply thresholds to these measures, identifying a subset of feature spaces that correlate with high classification accuracy. From these remaining feature spaces, we can use the negative correlation between Betti numbers of dimension $d-1$ and classification accuracy to identify those which are most appropriate for use with pattern classifiers.

Our work provides insight into the potential use of computational homology algorithms for studying the topological structure of class distributions within various feature spaces. For applications where the sequential ordering of classes is important, as it is in Lamb wave structural health monitoring, comparing the Betti number β_{d-1} of each feature space under consideration shows potential for identifying feature subspaces which return the highest classification accuracies.

Chapter 7

Conclusions & Future Work

7.1 Conclusions

In this work we have shown how intelligent feature selection techniques are needed for real-world applications of signal analysis using pattern classification. The motivation for performing this work was to study the manifestation of physical changes in signal characteristics as they translate to a representative feature space. All time-domain signals have an element of deviation from their baseline form under real-world conditions. When the underlying physics behind the signal of choice is very complicated to begin with, these changes can be difficult to accurately monitor if an optimal set of identifying features is not chosen.

We have demonstrated the ability of the DWFP technique for generating image representations of raw time-domain signals. These images are created using a continuous wavelet transform and projecting the three-dimensional coefficients onto the time-scale plane. Properties are measured from the resulting two-dimensional binary images using a variety of image processing techniques. These measurements are used to create feature vectors for each individual signal, which can be monitored to identify changes in the signal. We found that we were able to identify specific features in DWFP-transformed Rayleigh-Lamb wave signals that identify the interaction with both dents and surface cracks in aluminum plates. This processing technique was developed to monitor the damage generated from minor runway incidents.

This analysis technique was then applied to radio-frequency signals, where the DWFP technique was again used to generate signal features in an attempt to uniquely identify individual RFID tags from cloned copies. The difficult problem of identifying subtle tag signatures within the noisy recordings required the use of advanced pattern classification routines. A multi-dimensional feature space was generated using a series of DWFP features in addition to several statistical measures that represented the original data set. We introduced tag-to-reader variability commonly associated with RFID applications, including changes in environmental conditions and tag orientation/proximity to the RF antenna. We established the framework to study how these intra-class physical changes result in changes in the class structure within the feature space.

This framework was then applied to ultrasonic flaw characterization. Instead of simply identifying a possible flaw, we developed a pattern classification routine with the aim of characterizing the severity of a flaw. The concept of feature space reduction was explored. Once a feature space had been generated from the data set, an intelligent reduction in dimension was required for sufficient classification performance. The curse of dimensionality provides a relationship between the size of a data set and the dimensionality of the space representing it, suggesting it's necessary to have at least 10 times as many samples as there are features for reliable performance. Formal feature selection methods rely heavily on a representative training data set, but when an endless number of physical changes can occur that manifest themselves in a data set's feature space, such as intermediate flaw severities due to corrosion in metals, the concept of a representative training set becomes unrealistic. Rather than creating a fully representative set, we explore an alternate approach involving the intelligent selection of a feature space based on available training data where only a handful of intermediate classes exist. Features are then selected based on their ability to not only separate these classes, but to do so in a way that preserves the underlying class structure. We found that we were able to train a classifier using this intelligent feature selection that was then able to correctly identify the severity of new data that was not represented in the training set, but in fact was from an "intermediate" class.

This concept of intelligent feature selection was expanded upon with the introduction of computational homology theory. By including a measure of inter-class connectedness which relates directly to class separability in multi-dimensional feature spaces, we provide a new approach to the problem of reducing a feature space into a more optimal subspace. Betti numbers were extracted from each class distribution, relating the multi-dimensional connectedness of the space to the formal classification accuracy. We found a general trend of decreasing Betti numbers with increasing classifier accuracy, however artifacts in the higher-dimensional Betti number distributions indicate this method is not as straightforward as originally expected. The technique of combining three geometric measures along with the Betti number metric proved a useful approach for identifying a subset of the possible feature spaces of various dimensions.

7.2 Future Work

The RFID results presented in Chapter 4 provide an upper bound on the classification potential to correctly identify EPCs extracted from RFID tags. We observed how a variety of physical changes in the way RFID tags are used effect the RF signal and the resulting classification accuracy. Future work could focus on exploring how these changes manifest themselves in the data. Rather than introducing more variability in how data is collected, we suggest an analysis on the class distribution within the feature space for the physical variables described in this work. For example, an understanding of how the RFID tag position relative to the antenna changes the class distribution in the feature space could provide insight into how feasible this classification routine is for most real-world applications.

An intelligent feature selection routine should be explored for this application. We have already applied an intermediate feature selection step in the DWFP feature extraction process, however the inclusion of statistical, Mellin transform, and wavelet packet decomposition features results in a feature space of relatively high dimension. It has been shown that the size of the feature space relative to the number of objects per class can have an effect on a classifier's ability to distinguish between classes [9]. If the feature space is too large, the data within the feature space can become sparse as dimension increases, effectively over-complicating the decision boundary construction. One important step in the classification process that could benefit from refinement as the feature set grows is the intermediate DWFP feature selection. Alternate methods to identify and extract features of interest should be explored and tested against the Euclidean distance metrics in use currently.

Improvements to the classifier are also suggested. Further refinement of the EPC extraction routines could be performed, as our technique is specific to the encoding algorithm used by our RFID reader. Alternate readers types will potentially use alternate encoding techniques, so the EPC extraction should be generalized to accommodate these as well as alternate transmission protocols. New classifiers could be introduced to test against the group already in use.

The characterization of flaw severity analysis provided in Chapter 5 was able to classify the severity of several testing samples that were not represented by the training set. Testing additional validation plates is an obvious next step for this analysis. Determining how the classifier is able to identify gradual thickness losses like those found in corrosion would be a useful exercise. The training set may prove to be inadequate for these flaw types, and if so it should be determined what additional level of training is required to expand the potential flaw base able to be characterized by this algorithm.

Suggested future work also includes further development of the feature set used to represent the ultrasonic waveforms. Including additional time-frequency representations like the spectrogram could prove useful. We discuss a damage threshold analysis in Section 5.8.1, which reduces the multi-class problem at hand to a binary one of over/under a specific damage level. Therefore, classifiers that are inherently binary like support vector machines (SVM) could be explored and may prove useful for this application.

With regard to the geometric measures used for feature selection that were discussed in Section 5.6.1, other measures could be explored. We used intuition to select the three measures explored here, including the distance between class centroids, the average angle between sequential class centroids, and the tortuosity of the sequential class 'curve' within the feature space. Each is known to have its drawbacks, but more sophisticated measure could be applied which account for all three of these measures simultaneously. For example, a vector-like value could be assigned that accounts for both centroid distance as well as the direction of the next sequential class location relative to the previous classes. Additionally, for each of these measures, we apply a threshold to identify those feature spaces which are favorable for classification. A formal analysis of metric threshold values could be considered as this technique is further explored. Additionally, a weighing scheme rather than a threshold cutoff could be included, where the different geometric measures are assigned a weight based on their value for a given feature space

The computational homology feature selection technique introduced in Chapter 6 provides the framework to include further non-traditional measures in the selection of optimal feature space subsets. We use the standard deviation of each class to define the boundaries for the cubical set form of the class distribution, however additional methods to define each class boundary could be explored. These could include the skewness and kurtosis (or any of the higher-order moments) to provide usable bounds. Along these lines, different shapes may be used to define the classes as well. We used hypercubes because they are the simplest to implement using cubical data sets (which serves as input to the CHomP software), however the fact that the hypercubes larger than unit length are simply composed of multiple unit length elementary cubes means that alternate shapes can be constructed using these elementary cubes as well. Hyperspheres that are 'pixelated' in boundary could be defined around the class centroids, or even hyper-ellipsoids that vary in eccentricity based on intra-class dimensional variance.

In order to avoid some of the computational overhead associated with the current implementation of this algorithm, the CHomP algorithms could be implemented into MATLAB (or the MATLAB code could be implemented into C++, which would result in a less user-friendly interface but would reduce overhead associated with MATLAB). This would remove the intermediate processing step of creating a temporary input file and switching between computing languages. This would cut down on the computation time and memory requirements for the larger-dimensional feature spaces.

We have developed a robust signal analysis technique that is suitable for a wide variety of time-domain signal analysis applications. We have applied statistical pattern classification routines to problems of interest involving a physical change in the domain of the problem that translated into the signal characteristics. We have explored intelligent feature selection techniques in an attempt to identify optimal representations of these data sets.

Appendix A

Rayleigh-Lamb Frequency Equation Derivation

In Cartesian coordinates, the three governing equations of linear elasticity for an isotropic homogeneous material are [164]

$$\rho\partial_t^2 u_i - \partial_j\sigma_{ij} = 0 \tag{A.1}$$

$$\epsilon_{kl} = \frac{1}{2}(\partial_k u_l + \partial_l u_k) \tag{A.2}$$

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \tag{A.3}$$

with

$$C_{ijkl} = \delta_{ij}\delta_{kl}\lambda + (\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})\mu$$

where (A.1) is the equation of motion, (A.2) is a strain-displacement relation for linear-elastic solids assuming infinitesimal strains, and (A.3) is the generalized Hooke's Law. In these equations, u_i is the displacement vector in a three dimensional elastic material, σ_{ij} is the stress tensor, ρ is the material density, and λ and μ are the Lamé parameters. Equations (A.2) and (A.3) can be substituted in (A.1) to derive Navier's equation

$$\rho\partial_t^2 u_i - \mu\partial^2 u_i - (\lambda+\mu)\partial_i(\partial_j u_j) = 0, \tag{A.4}$$

or in vector notation,

$$\rho\partial_t^2\vec{u} - \mu\nabla^2\vec{u} - (\lambda+\mu)\nabla(\nabla\cdot\vec{u}) = 0. \tag{A.5}$$

Harmonic time variations are suppressed by setting

$$u(\vec{x},t) = u(\vec{x})e^{-i\omega t}, \quad \partial_t \Rightarrow -i\omega, \quad \partial_t^2 \Rightarrow -\omega^2$$

and it follows that (A.5) becomes

$$-\omega^2\rho\vec{u} - \mu\nabla^2\vec{u} - (\lambda+\mu)\nabla(\nabla\cdot\vec{u}) = 0. \tag{A.6}$$

A Helmholtz decomposition can be applied,

$$\vec{u} = \vec{u_L} + \vec{u_T} = \nabla\Phi + \nabla\times\vec{H} \tag{A.7}$$

since any vector field can be written as the sum of two parts: $\vec{u_L}$ having no curl, and $\vec{u_T}$ being a vector having no divergence. Using the vector identities

$$\nabla^2\vec{X} = \nabla(\nabla\cdot\vec{X}) - \nabla\times(\nabla\times\vec{X})$$

$$\nabla\cdot(\nabla\times\vec{H}) = 0$$

$$\nabla\times(\nabla\Phi) = 0$$

$$\nabla\cdot(\nabla\Phi) = \nabla^2\Phi$$

and substituting (A.7) into (A.6) results in

$$\nabla\left[\left[-\omega^2\rho-(\lambda+2\mu)\nabla^2\right]\Phi\right]=\nabla\times\left[\left[\omega^2\rho+\mu\nabla^2\right]\vec{H}\right] \tag{A.8}$$

which is only satisfied when

$$\nabla^2\Phi+\frac{\omega^2\rho}{\lambda+2\mu}\Phi=0 \tag{A.9}$$

$$\nabla^2\vec{H}+\frac{\omega^2\rho}{\mu}\vec{H}=0. \tag{A.10}$$

These two equations are longitudinal and transverse wave equations with velocities

$$c_L=\sqrt{\frac{\lambda+2\mu}{\rho}} \tag{A.11}$$

$$c_T=\sqrt{\frac{\mu}{\rho}}. \tag{A.12}$$

The resulting linearly independent scalar wave equations

$$\nabla^2\Phi=\frac{1}{c_L^2}\frac{\partial^2\Phi}{\partial t^2} \tag{A.13}$$

$$\nabla^2H_n=\frac{1}{c_T^2}\frac{\partial^2H_n}{\partial t^2}\qquad n=x,y,z \tag{A.14}$$

show that longitudinal and transverse waves propagate independently in an infinite homogeneous elastic medium. However, mode coupling does occur at material boundaries, where a propagating longitudinal wave will generate transverse waves and vice versa. This mode coupling is included implicitly in the boundary conditions. In plate-like or pipe-like structures, with two parallel traction-free surface boundaries, this mode coupling gives rise to the development of guided wave modes; when the material thickness is on the same order as the wavelength, the superposition of the longitudinal and transverse waves as well as their coupled reflections mix to produce wave packets that propagate throughout the material, better known as Lamb waves. The following analysis examines the derivation of the Lamb wave solutions. For a more complete discussion, see Graff [165].

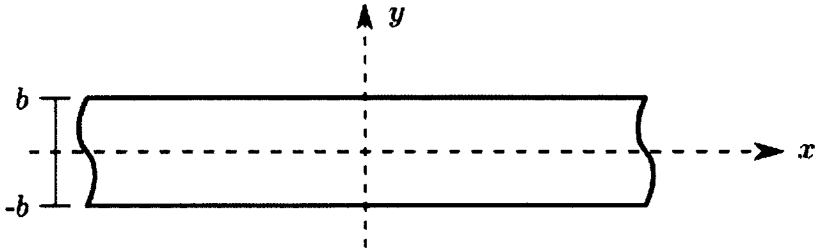


Figure A.1: Coordinate system for a plate in the $x - z$ plane with thickness $y = 2b$.

Consider a coordinate system with a plate in the $x - z$ plane with upper and lower traction-free boundaries at $y = \pm b$, as seen in Figure A.1. Boundary conditions are given by

$$\sigma_{yy} = \sigma_{xy} = \sigma_{zy} = 0, \qquad y = \pm b. \tag{A.15}$$

If variations with respect to z are excluded, displacements can be written

$$\begin{aligned} u_x &= \frac{\partial \Phi}{\partial x} + \frac{\partial H_z}{\partial y} \\ u_y &= \frac{\partial \Phi}{\partial y} + \frac{\partial H_z}{\partial x} \\ u_z &= \frac{\partial H_x}{\partial y} + \frac{\partial H_y}{\partial x}. \end{aligned} \tag{A.16}$$

For plane waves in the $x - z$ plane, consider solutions of the general form

$$\begin{aligned} \Phi &= f(y)e^{i(\xi x - \omega t)} \\ II_x &= h_x(y)e^{i(\xi x - \omega t)} \\ H_y &= h_y(y)e^{i(\xi x - \omega t)} \\ H_z &= h_z(y)e^{i(\xi x - \omega t)}. \end{aligned} \tag{A.17}$$

Substitution of these in (A.13) and (A.14) gives

$$\begin{aligned}\Phi &= (A\cos\alpha y + B\sin\alpha y)e^{i(\xi x-\omega t)}\\ H_x &= (C\cos\beta y + D\sin\beta y)e^{i(\xi x-\omega t)}\\ H_y &= (E\cos\beta y + F\sin\beta y)e^{i(\xi x-\omega t)}\\ H_z &= (G\cos\beta y + H\sin\beta y)e^{i(\xi x-\omega t)}\end{aligned} \tag{A.18}$$

where

$$\alpha^2 = \frac{\omega^2}{c_L^2} - \xi^2 \qquad \text{and} \qquad \beta^2 = \frac{\omega^2}{c_T^2} - \xi^2. \tag{A.19}$$

The displacements then become

$$\begin{aligned}u_x =&[i\xi(A\cos\alpha y + B\sin\alpha y) + \beta(-G\sin\beta y + H\cos\beta y)]e^{i(\xi x-\omega t)}\\ u_y =&[\alpha(-A\sin\alpha y + B\cos\alpha y) - i\xi(G\cos\beta y + H\sin\beta y)]e^{i(\xi x-\omega t)}\\ u_z =&[-\beta(-C\sin\beta y + D\cos\beta y) + i\xi(E\cos\beta y + F\sin\beta y)]e^{i(\xi x-\omega t)}.\end{aligned} \tag{A.20}$$

From (A.3), the stresses are

$$\begin{aligned}\sigma_{yy} =&(\lambda+2\mu)\frac{\partial u_y}{\partial y} + \lambda\frac{\partial u_x}{\partial x}\\ \sigma_{xy} =&\mu(\frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y})\\ \sigma_{yz} =&\mu\frac{\partial u_z}{\partial y}.\end{aligned} \tag{A.21}$$

The boundary conditions provide six equations to solve for the eight unknown constants, $A, B, \ldots, H$. The final two equations come from the divergence condition on $\vec{H}$, given by

$$\frac{\partial H_z}{\partial x} + \frac{\partial H_y}{\partial y} = 0 \quad \text{at} \quad y = \pm b. \tag{A.22}$$

The resulting system of eight homogeneous equations is

$$\begin{aligned}
\{(\lambda+2\mu)\alpha^2+\lambda\xi^2\}(A\cos\alpha b+B\sin\alpha b)+2i\mu\beta\xi(-G\sin\beta b+H\cos\beta b)&=0\\
\{(\lambda+2\mu)\alpha^2+\lambda\xi^2\}(A\cos\alpha b-B\sin\alpha b)+2i\mu\beta\xi(G\sin\beta b+H\cos\beta b)&=0\\
2i\alpha\xi(-A\sin\alpha b+B\cos\alpha b)+(\xi^2-\beta^2)(G\cos\beta b+H\sin\beta b)&=0\\
2i\alpha\xi(A\sin\alpha b+B\cos\alpha b)+(\xi^2-\beta^2)(G\cos\beta b-H\sin\beta b)&=0\\
\beta^2(C\cos\beta b+D\sin\beta b)+i\beta\xi(-E\sin\beta b+F\cos\beta b)&=0\\
\beta^2(C\cos\beta b-D\sin\beta b)+i\beta\xi(E\sin\beta b+F\cos\beta b)&=0\\
i\xi(C\cos\beta b+D\sin\beta b)+\beta(-E\sin\beta b+F\cos\beta b)&=0\\
i\xi(C\cos\beta b-D\sin\beta b)+\beta(E\sin\beta b+F\cos\beta b)&=0.
\end{aligned}\tag{A.23}$$

The determinant of the coefficient matrix must vanish in order to ensure a solution to this system other than the trivial one:

$$\begin{vmatrix}
c\cos\alpha b & c\sin\alpha b & 0 & 0 & -f\sin\beta b & f\cos\beta b & 0 & 0\\
c\cos\alpha b & -c\sin\alpha b & 0 & 0 & f\sin\beta b & f\cos\beta b & 0 & 0\\
0 & 0 & -h\sin\beta b & h\cos\beta b & 0 & 0 & \beta^2\cos\beta b & \beta^2\sin\beta b\\
0 & 0 & -h\sin\beta b & h\cos\beta b & 0 & 0 & \beta^2\cos\beta b & -\beta^2\sin\beta b\\
-d\sin\alpha b & d\cos\alpha b & 0 & 0 & g\cos\beta b & g\sin\beta b & 0 & 0\\
d\sin\alpha b & d\cos\alpha b & 0 & 0 & g\cos\beta b & -g\sin\beta b & 0 & 0\\
0 & 0 & -\beta\sin\beta b & \beta\cos\beta b & 0 & 0 & i\xi\cos\beta b & i\xi\sin\beta b\\
0 & 0 & \beta\sin\beta b & \beta\cos\beta b & 0 & 0 & i\xi\cos\beta b & -i\xi\sin\beta b
\end{vmatrix}=0 \tag{A.24}$$

where

$$c=\{(\lambda+2\mu)\alpha^2+\lambda\xi^2\},\qquad d=2i\alpha\xi,\qquad f=2i\mu\beta\xi,\qquad g=\xi^2-\beta^2,\qquad h=i\beta\xi.$$

Each of the columns of the determinant are associated with constants A, B, E, F, G, H, C, D respectively. By applying standard matrix manipulations, the determinant can be written as the product of four sub-determinants,

$$\begin{vmatrix} i\xi\cos\beta b & \beta\cos\beta b \\ \beta^2\cos\beta b & h\cos\beta b \end{vmatrix} \times \begin{vmatrix} -\beta\sin\beta b & i\xi\sin\beta b \\ h\sin\beta b & \beta^2\sin\beta b \end{vmatrix} \times \begin{vmatrix} c\cos\alpha b & f\cos\beta b \\ -d\sin\alpha b & g\sin\beta b \end{vmatrix} \times \begin{vmatrix} g\cos\beta b & d\cos\alpha b \\ f\sin\beta b & c\sin\alpha b \end{vmatrix} = 0 \tag{A.25}$$

where the coefficients associated with the subdeterminants are

$$|C, F| \times |E, D| \times |A, H| \times |B, G| = 0 \tag{A.26}$$

It follows that four families of solutions exist for various non-zero combinations of the constants:

$$\text{Solution I} : A, B, D, E, G, H = 0 \quad C, F \neq 0 \tag{A.27}$$

$$\text{Solution II} : A, B, C, F, G, H = 0 \quad D, E \neq 0 \tag{A.28}$$

$$\text{Solution III} : B, C, D, E, F, G = 0 \quad A, H \neq 0 \tag{A.29}$$

$$\text{Solution IV} : A, C, D, E, F, H = 0 \quad B, G \neq 0 \tag{A.30}$$

Solutions I and II are pure shear waves with displacement in the z direction corresponding to antisymmetric and symmetric shear horizontal (SH) modes, respectively, further details of which can be found in texts by Auld [74], Graff [165], and Rose [72]. Solutions III and IV are coupled longitudinal and shear vertical waves, usually referred to as Lamb waves. Solution III presents a case where the motion occurs symmetrically with respect to the $z = 0$ plane, with displacements defined as

$$\begin{aligned} u_x &= (i\xi A\cos\alpha y + \beta H\cos\beta y)e^{i(\xi x-\omega t)} \\ u_y &= -(\alpha A\sin\alpha y + \xi H\sin\beta y)e^{i(\xi x-\omega t)} \\ u_z &= 0 \end{aligned} \tag{A.31}$$

while solution IV describes waves in which the motion is antisymmetric with respect to $z = 0$, with displacements defined as

$$\begin{aligned} u_x =& (i\xi B \sin \alpha y - \beta G \sin \beta y) e^{i(\xi x - \omega t)} \\ u_y =& (\alpha B \cos \alpha y - i\xi G \cos \beta y) e^{i(\xi x - \omega t)} \\ u_z =& 0. \end{aligned} \tag{A.32}$$

Each of these solutions results in a subsystems which can be presented as a system of linear equations itself, both of which have a solution when the determinant of their coefficient matrices vanish, as given by

$$\begin{bmatrix} c \cos \alpha b & f \cos \beta b \\ -d \sin \alpha b & g \sin \beta b \end{bmatrix} = 0 \qquad \text{Solution III} : \textit{Symmetric} \tag{A.33}$$

$$\begin{bmatrix} g \cos \beta b & d \cos \alpha b \\ f \sin \beta b & c \sin \alpha b \end{bmatrix} = 0. \qquad \text{Solution IV} : \textit{Anti-symmetric} \tag{A.34}$$

Equations (A.33) and (A.34) can be re-written in the following dimensionless form

$$\frac{\tan \beta b}{\tan \alpha b} = \frac{-4\alpha\beta\xi^2}{(\xi^2 - \beta^2)} \qquad \textit{Symmetric} \tag{A.35}$$

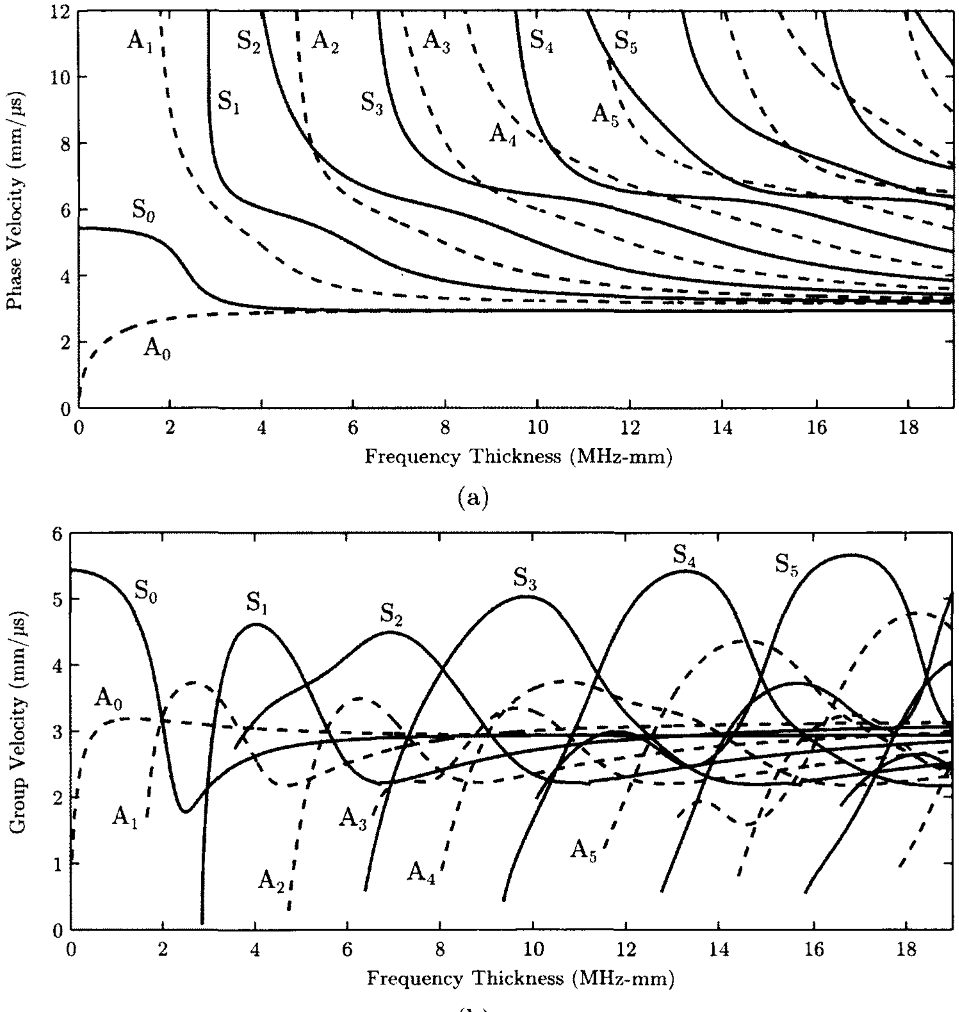
$$\frac{\tan \beta b}{\tan \alpha b} = \frac{(\xi^2 - \beta^2)}{-4\alpha\beta\xi^2} \qquad \textit{Antisymmetric.} \tag{A.36}$$

These equations are known as the Rayleigh-Lamb frequency equations. Since the wavenumber, ξ, is numerically equal to the frequency (ω) divided by the phase velocity (c_p), equations (A.35) and (A.36) provide a way to relate phase velocity to frequency-thickness, resulting in dispersion curves. The real valued solutions to these dispersion relations represent the undamped propagating modes of the structure, and can be found by numerically applying a root-finding algorithm to determine the existing phase velocity values for a given frequency, applied over a range of frequency values [72, 166]. The group velocity can then be found from the phase velocity by

$$c_g = \frac{d\omega}{dk} = c_p^2 \left[c_p - (fd) \frac{dc_p}{d(fd)} \right]^{-1} . \tag{A.37}$$

Since λ and μ are material properties, dispersion curves are unique to the material of interest. As an example, phase velocity and group velocity dispersion curves are shown for Al 2024 ($\rho = 2.78$ g/cm^3, Young's modulus $E = 73.1$ GPa, Poisson's ratio $\nu = 0.33$)[1] in Figure A.2. It can be seen that as the frequency-thickness product increases, so does the number of existing Lamb wave modes. All of these higher-order modes, aside from A_0 and S_0, have cutoff frequency-thickness values where the phase velocity approaches infinity and the group velocity approaches zero, i.e. whenever standing longitudinal or shear waves are present. If the plate thickness is known, an inspection frequency can be chosen that generates a highly dispersive mode where any changes in thickness or material properties will result in a change of propagating group velocity. For example, a 2.5 mm thick aluminum plate sample excited with a 1 MHz toneburst would result in an fd value of about 2.5 MHz-mm, which from the group velocity dispersion curve (Figure A.2(b)) can be seen to correlate with three existing modes, A_1, A_0, and S_0. Any changes in the thickness in the plate would result in a proportional change in the fd value. If a slight thinning was present, the fd value would decrease slightly, resulting in a slower A_1 mode velocity, no change in the A_0 mode velocity, and an increase in the S_0 mode velocity. This dispersive property is one of the reasons Lamb waves are so fitting for use in NDE applications. By propagating Lamb waves across a region of interest, changes in the various mode arrival times can help identify the location of flaws or corrosion, and often directly

[1] http://asm.matweb.com/search/SpecificMaterial.asp?bassnum=MA2024T4



(a)

(b)

Figure A.2: Phase (a) and group (b) velocity dispersion curves for aluminum 2024. Antisymmetric modes are shown in in blue, symmetric modes are shown in red.

correlate to changes in material thickness. This allows for the detection of structural weaknesses before serious failure occurs.

While modes will often overlap each other in time, if a frequency-thickness regime is being used where modes are expected to be separated in time, a modal analysis can be performed to help identify which modes will be most sensitive to the flaws at hand. The constants A, B, G, and H can be solved using the boundary conditions (A.23), resulting in

$$A = \cos \beta b \tag{A.38}$$

$$B = \sin \beta b \tag{A.39}$$

$$G = -\frac{(\xi^2 - \beta^2)}{2i\xi\beta} \sin \alpha b \tag{A.40}$$

$$H = \frac{(\xi^2 - \beta^2)}{2i\xi\beta} \cos \alpha b. \tag{A.41}$$

It follows that for both the symmetric (Solution III) and anti-symmetric (Solution IV) families of solutions, displacement components (A.20) as well as stress components (A.21) can be determined for a given mode and frequency-thickness value. Using these, the time-averaged projection of the Poynting vector onto the x coordinate as a function of the depth can then be written as [167]

$$< P_x >= -\frac{1}{2}\left(u_y^* \sigma_{xy} + u_x^* \sigma_{xx}\right) \tag{A.42}$$

where * denotes complex conjugation. This value represents the rate at which energy is transported by a given mode per unit cross-sectional area. Integrating across the thickness of the plate yields the time averaged power flow in the x direction

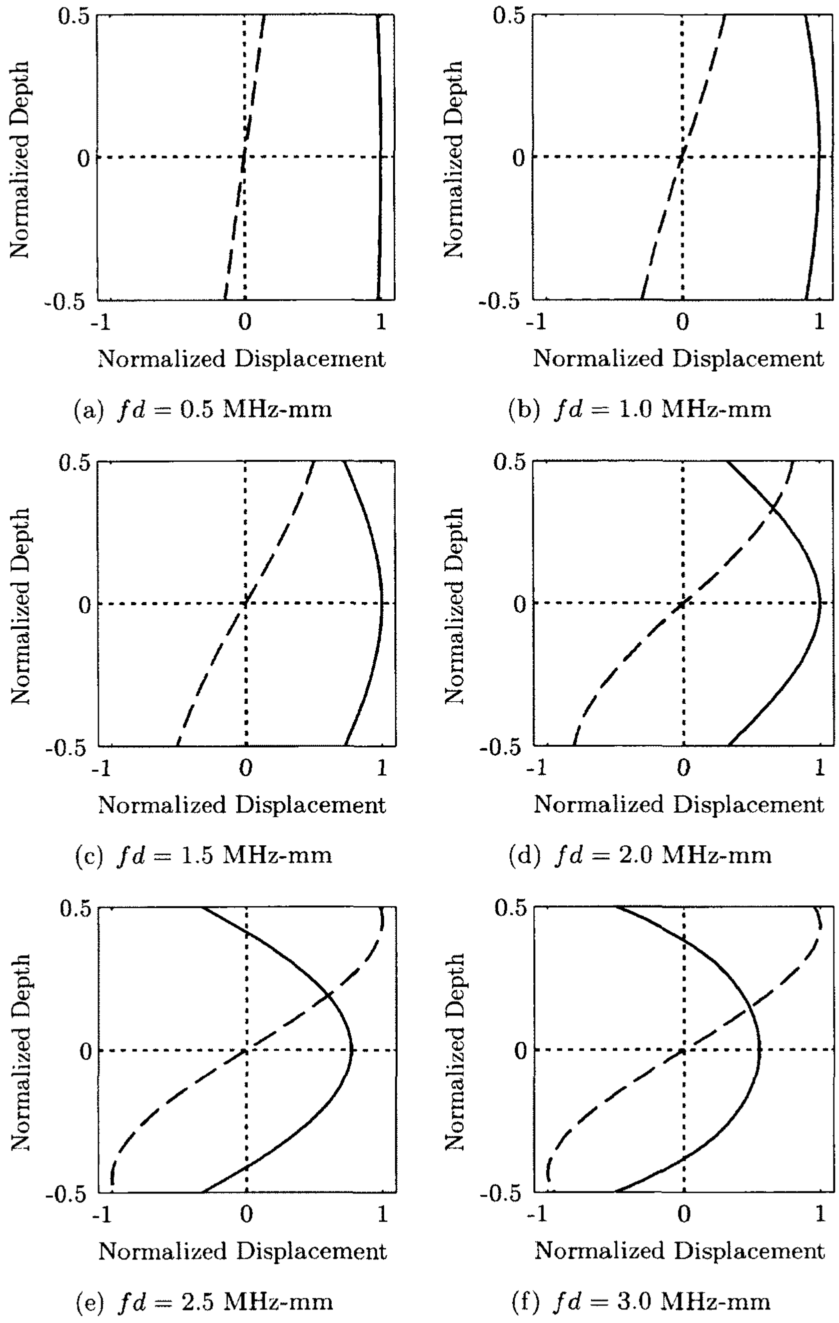
$$< P_{nn} >= \int_{-\frac{b}{2}}^{\frac{b}{2}} < P_x > \mathrm{d}y \tag{A.43}$$

which is then used as a "percent" energy function

$$\%P_{nn}(\zeta) = \frac{\int_0^{\zeta} < P_x > \mathrm{d}y}{< P_{nn} >} \times 100 \tag{A.44}$$

which gives the percent of the total energy flow rate included in the plate from the top surface to a depth ζ. In other words, this function provides a measure of the localization of the energy of a mode.

An example of how these parameters are useful is provided here, using the dispersion curves for aluminum, seen in Figure A.2. The structure of each wave mode changes as the frequency thickness value changes, seen by the normalized mode displacement components calculated for several fd values for both the S_0 and A_0 modes in Figures A.3 and A.4. The in-plane (x) component of the S_0 mode is essentially constant across the plate thickness at lower fd values, however it becomes much more concentrated at the center of the plate as the fd value increases. Meanwhile, the out-of-plane (y) component is close to zero on the outside surfaces for small fd values, but becomes dominant on the outside surface as the fd value increases. A similar analysis can be made for the A_0 mode. This type of mode structure analysis can be important in determining an inspection frequency. An increase in sensitivity at various depths of a plate can be gained by adjusting the modal in-plane or out-of-plane displacement at that depth. In situations where a plate is surrounded by water or insulation, it is advantageous to use modes that have dominant in-plane displacement with minimal out-of-plane displacement on the outside surface since shear waves cannot propagate in fluids, thereby reducing the leakage of wave energy into the fluid. If the detection of surface defects was a goal, increasing the concentration of energy on the outside surface of a plate is advantageous. In this case, percent energy flow comparisons could be made to determine which of the possible modes had the highest concentration of energy at the surface. An example of this can be seen in Figure A.5 where the percent energy flow versus depth is presented for the S_0 and A_0 modes at $fd = 2.5$. The A_0 mode has more of its energy located near the upper and lower surfaces of the plate, making it appropriate to use to use for surface defect detection, while the less ideal S_0 mode has its energy concentrated near the center of the plate. These types of applications of mode-related features can prove to be very useful in NDE testing to interrogate structures for flaws.



(a) $fd = 0.5$ MHz-mm

(b) $fd = 1.0$ MHz-mm

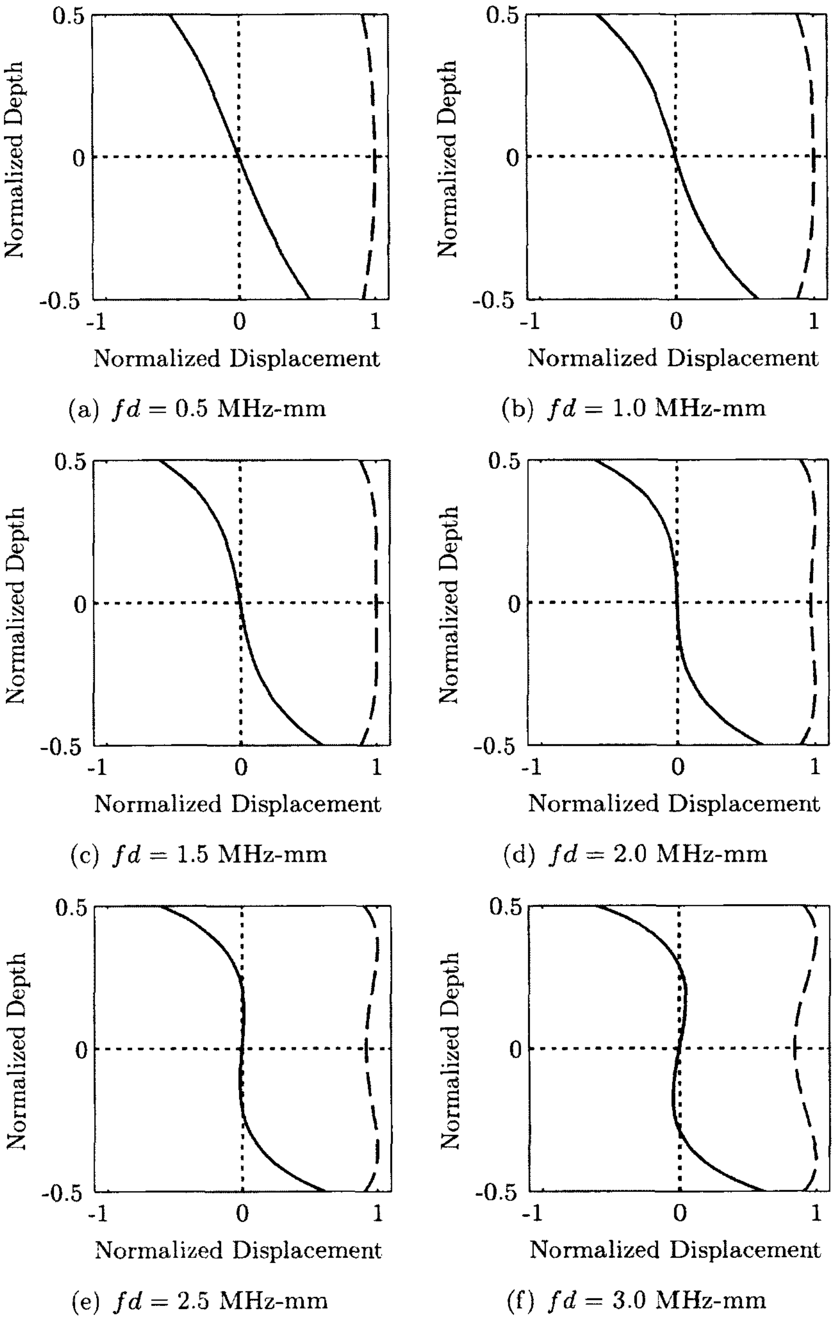
(c) $fd = 1.5$ MHz-mm

(d) $fd = 2.0$ MHz-mm

(e) $fd = 2.5$ MHz-mm

(f) $fd = 3.0$ MHz-mm

Figure A.3: Wave structure for points on the S_0 mode of an aluminum plate. Both the in-plane u_x (solid line) and out-of-plane u_y (dashed line) normalized displacement components are shown versus normalized material depth.



(a) $fd = 0.5$ MHz-mm

(b) $fd = 1.0$ MHz-mm

(c) $fd = 1.5$ MHz-mm

(d) $fd = 2.0$ MHz-mm

(e) $fd = 2.5$ MHz-mm

(f) $fd = 3.0$ MHz-mm

Figure A.4: Wave structure for points on the A_0 mode of an aluminum plate. Both the in-plane u_x (solid line) and out-of-plane u_y (dashed line) normalized displacement components are shown versus normalized material depth.

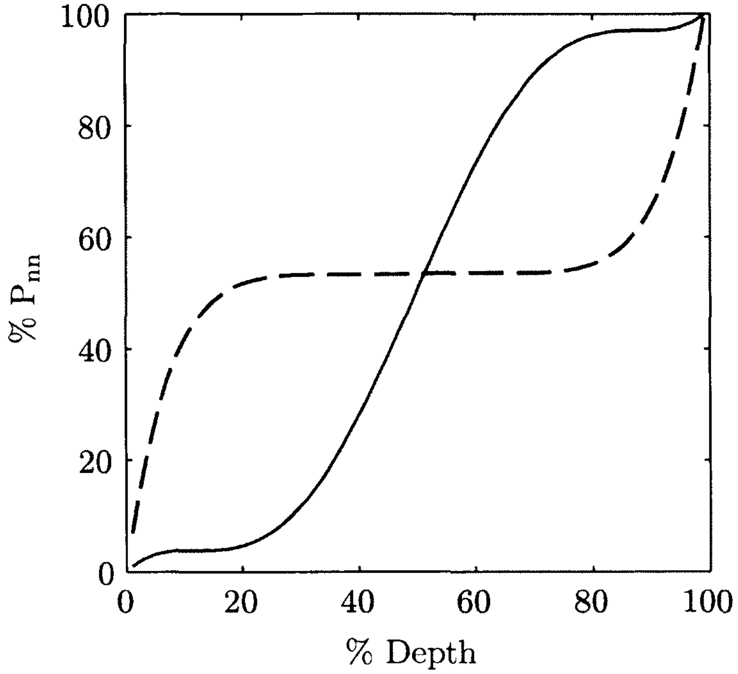


Figure A.5: Percent energy flow versus percent depth for S_0 (dashed line) and A_0 (solid line) modes at $fd = 2.5$ MHz-mm. The A_0 mode has more of its energy flow located near the upper and lower surfaces of a plate, while the S_0 mode has its energy flow concentrated near the center of a plate.

Appendix B

Computational Homology Theory

We define homology and it's elementary properties here. Specifically, we explore topological spaces that are built out of d-dimensional unit cubes with vertices on the integer lattice. We begin by establishing the notion of cubes, the building blocks for cubical homology. This introduction follows [158].

Definition 1. An *elementary interval* is a closed interval $I \subset \mathbf{R}$ of the form

$$I = [l, l+1] \quad \text{or} \quad I = [l, l]$$

for some $l \in \mathbf{Z}$.

Definition 2. An *elementary cube* Q is a finite product of elementary intervals, that is

$$Q = I_1 \times I_2 \times \cdots \times I_d \subset \mathbf{R}^d,$$

where each I_i is an elementary interval. The set of all elementary cubes in $\mathbf{R}^d$ is denoted by $\mathcal{K}^d$. Also, let $\mathcal{K}_k = \{Q \in \mathcal{K} \mid \dim Q = k\}$ and $\mathcal{K}_k^d = \mathcal{K}_k \cap \mathcal{K}^d$.

Proposition 1. Let $Q \in \mathcal{K}_k^d$ and $P \in \mathcal{K}_{k'}^{d'}$. Then

$$Q \times P \in \mathcal{K}_{k+k'}^{d+d'}.$$

Definition 3. A set $X \in \mathbf{R}^d$ is *cubical* if X can be written as a finite union of elementary cubes.

Now we need definitions that can be used to pass from the topology of a cubical set to the algebra of homology. With each elementary k-cube $Q \in \mathcal{K}_k^d$ we associate an algebraic object $\widehat{Q}$ called an *elementary k-chain* of $\mathbf{R}^d$. The set of all elementary k-chains of $\mathbf{R}^d$ is denoted by

$$\widehat{\mathcal{K}}_k^d = \left\{ \widehat{Q} \mid Q \in \mathcal{K}_k^d \right\}.$$

Given any *finite* collection of $\left\{ \widehat{Q}_1, \widehat{Q}_2, \ldots, \widehat{Q}_m \right\} \subset \mathcal{K}_k^d$ of k-dimensional elementary chains, we can consider the sum

$$c = \alpha_1 \widehat{Q_1} + \alpha_2 \widehat{Q_2} + \ldots + \alpha_m \widehat{Q_m},$$

where α_i are arbitrary integers. These can be thought of as k-chains, the set of which is denoted by C_k^d. It should be noted that C_k^d is a free abelian group with basis $\widehat{\mathcal{K}}_k^d$. Each elementary cube is used to generate a basis element called an elementary chain, and therefore a chain is just defined as a finite sum of elementary chains.

Definition 4. Consider $c_1, c_2 \in C_k^d$, where $c_1 = \sum\limits_{i=1}^{m} \alpha_i \widehat{Q}_i$ and $c_2 = \sum\limits_{i=1}^{m} \gamma_i \widehat{Q}_i$. The *scalar product* of the chains c_1 and c_2 is defined as

$$\langle c_1, c_2 \rangle = \sum_{i=1}^{m} \alpha_i \gamma_i.$$

Definition 5. Given two elementary cubes $P \in \mathcal{K}_k^d$ and $Q \in \mathcal{K}_{k'}^{d'}$ set $\widehat{P} \diamond \widehat{Q} = \widehat{P \times Q}$. This definition extends to arbitrary chains $c_1 \in C_k^d$ and $c_2 \in C_{k'}^{d'}$ by

$$c_1 \diamond c_2 = \sum_{P \in \mathcal{K}_k, Q \in \mathcal{K}_{k'}} \left\langle c_1, \widehat{P} \right\rangle \left\langle c_2, \widehat{Q} \right\rangle \widehat{P \times Q}$$

The chain $c_1 \diamond c_2 \in C_{k+k'}^{d+d'}$ is called the *cubical product* of c_1 and c_2.

Definition 6. Given $k \in \mathbf{Z}$, the *cubical boundary operator*

$$\partial_k : C_k^d \to C_{k-1}^d$$

is a homomorphism of free abelian groups. This is defined for an elementary chain $\widehat{Q} \in \mathcal{K}_k^d$ by induction on d as follows.

Let $d = 1$. Then Q is an elementary interval and therefore $Q = [l] \in \mathcal{K}_0^1$ or $Q = [l, l+1] \in \mathcal{K}_1^1$ for some $l \in \mathbf{Z}$. Define

$$\partial_k \widehat{Q} = \begin{cases} 0 & \text{if} Q = [l], \\ \left[\widehat{l+1}\right] - [\widehat{l}] & \text{if} Q = [l, l+1]. \end{cases}$$

Now assume that $d > 1$. Let $I = I_1(Q)$ and $P = I_2(Q) \times \cdots \times I_d(Q)$. Then we have that

$$\widehat{Q} = \widehat{I} \diamond \widehat{P}.$$

Now define

$$\partial_k \widehat{Q} = \partial_{k_1} \widehat{I} \diamond \widehat{P} + (-1)^{\dim I} \widehat{I} \diamond \partial_{k_2} \widehat{P},$$

where $k_1 = \dim I$ and $k_1 = \dim P$. Finally, we extend the definition to all chains by linearity; if $c = \alpha_1 \widehat{Q}_1 + \alpha_2 \widehat{Q}_2 + \cdots + \alpha_m \widehat{Q}_m$, then

$$\partial_k c = \alpha_1 \partial_k \widehat{Q}_1 + \alpha_2 \partial_k \widehat{Q}_2 + \cdots + \alpha_m \partial_k \widehat{Q}_m.$$

Proposition 2. Let c and c' be cubical chains. We then have

$$\partial(c \diamond c') = \partial c \diamond c' + (-1)^{\dim c} c \diamond \partial c'.$$

Proposition 3.

$$\partial \diamond \partial = 0$$

This is a crucial property of the boundary operator and simply says that the boundary of a boundary is always zero.

Definition 7. The boundary operator for the cubical set X is defined to be

$$\partial_k^X : C_k(X) \to C_{k-1}(X)$$

by restricting $\partial_k : C_k^d \to C_{k-1}$ to $C_k(X)$. The lack of superscript X in ∂_k^X was intentionally left out due to the clear context here.

We now introduce the concept of *homology groups*. Let $X \subset \mathbf{R}^d$ be a cubical set. A k-chain $z \in C_k(X)$ is called a *cycle* in X if $\partial z = 0$. The kernel of a linear map is the set of elements that are sent to zero and is a subgroup of the domain. Therefore, the set of all k-cycles in X, which is denoted by $Z_k(X)$, is $\ker \partial_k^X$ and forms a subgroup of $C_k(X)$. This can be summarized by the follow relations:

$$Z_k(X) = \ker \partial_k^X = C_k(X) \cap \ker \partial_k \subset C_k(X).$$

A k-chain $z \in C_k(X)$ is called a *boundary* in X if there exists $c \in C_{k+1}(X)$ such that $\partial c = z$. The set of boundary elements in $C_k(X)$, which is denoted by $B_k(X)$, therefore consists of the image of ∂_{k+1}^X. Since ∂_{k+1}^X is a homomorphism, $B_k(X)$ is a subgroup of $C_k(X)$. Again, this can be summarized by

$$B_k(X) = \mathrm{im}\partial_{k+1}^X = \partial_{k+1}\left(C_{k+1}(X)\right) \subset C_k(X).$$

By Proposition 3, $\partial c = z$ implies that $\partial z = \partial^2 c = 0$, meaning that every boundary is a cycle and $B_k(X)$ is a subgroup of $Z_k(X)$. Since every boundary is a cycle, we treat all cycles that are boundaries as trivial. We are therefore interested in cycles that are not boundaries. To give nontrivial cycles an algebraic structure, we introduce an equivalence relation by saying that two cycles $z_1, z_2 \in Z_k(X)$ are homologous and we write z_1 z_2 if $z_1 - z_2$ is a boundary of X, that is, $z_1 - z_2 \in B_k(X)$. The equivalence classes are elements of the quotient group $Z_k(X)/B_k(X)$.

Definition 8. The kth *cubical homology group* is the quotient group

$$H_k(X) = Z_k(X)/B_k(X).$$

We will use the homology groups of the cubical set X to gain information about the topological structure of X by introducing *Betti numbers.* Betti numbers are generated by the homology groups of a topological space, where the kth Betti number measures the dimensions of the kth homology group

$$\beta_k = \dim H_k.$$

We now present an example to illustrate how homology groups, and therefore Betti numbers, are computed.

B.1 Example Calculation of Homology Groups

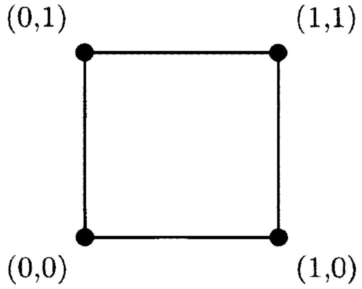


Figure B.1: Cubical set Γ.

Consider the cubical set $\Gamma = [0] \times [0,1] \cup [1] \times [0,1] \cup [0,1] \times [0] \cup [0,1] \times [1]$ associated with Figure B.1. The elementary set of cubes are

$$\begin{aligned}\mathcal{K}_0(\Gamma) &= \{[0] \times [0], [0] \times [1], [1] \times [0], [1] \times [1]\}\\ \mathcal{K}_1(\Gamma) &= \{[0] \times [0,1], [1] \times [0,1], [0,1] \times [0], [0,1] \times [1]\}\end{aligned}$$

from which the bases for the sets of chains are

$$\begin{aligned}\widehat{\mathcal{K}}_0(\Gamma) &= \left\{\widehat{[0] \times [0]}, \widehat{[0] \times [1]}, \widehat{[1] \times [0]}, \widehat{[1] \times [0]}\right\}\\ &= \left\{\widehat{[0]} \diamond \widehat{[0]}, \widehat{[0]} \diamond \widehat{[1]}, \widehat{[1]} \diamond \widehat{[0]}, \widehat{[1]} \diamond \widehat{[1]}\right\}\\ \widehat{\mathcal{K}}_1(\Gamma) &= \left\{\widehat{[0] \times [0,1]}, \widehat{[1] \times [0,1]}, \widehat{[0,1] \times [0]}, \widehat{[0,1] \times [1]}\right\}\\ &= \left\{\widehat{[0]} \diamond \widehat{[0,1]}, \widehat{[1]} \diamond \widehat{[0,1]}, \widehat{[0,1]} \diamond \widehat{[0]}, \widehat{[0,1]} \diamond \widehat{[1]}\right\}\end{aligned}$$

We next compute the boundary of the basis elements

$$\begin{aligned}\partial(\widehat{[0]} \diamond \widehat{[0,1]}) &= -\widehat{[0]} \diamond \widehat{[0]} + \widehat{[0]} \diamond \widehat{[1]}\\ \partial(\widehat{[1]} \diamond \widehat{[0,1]}) &= -\widehat{[1]} \diamond \widehat{[0]} + \widehat{[1]} \diamond \widehat{[1]}\\ \partial(\widehat{[0,1]} \diamond \widehat{[0]}) &= -\widehat{[0]} \diamond \widehat{[0]} + \widehat{[1]} \diamond \widehat{[0]}\\ \partial(\widehat{[0,1]} \diamond \widehat{[1]}) &= -\widehat{[0]} \diamond \widehat{[1]} + \widehat{[1]} \diamond \widehat{[1]}\end{aligned}$$

Which we can put into matrix form:

$$\partial_1 = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

We need to know $\ker \partial_1$ in order to understand $Z_1(\Gamma)$. We therefore solve

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which gives

$$\alpha_1 = -\alpha_2 = -\alpha_3 = \alpha_4.$$

and therefore

$$Z_1(\Gamma) = \{\alpha[1, -1, -1, 1]^T \mid \alpha \in \mathbf{Z}\}.$$

Since $C_2(\Gamma) = 0$, $B_1(\Gamma) = 0$ and hence

$$H_1(\Gamma) = Z_1(\Gamma) \cong \mathbf{Z}.$$

It follows that $\beta_1 = 1$.

Next, we compute $H_0(\Gamma)$. It should be first noted that there is no solution to the equation

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

which implies that $\widehat{[0]} \diamond \widehat{[0]} \notin B_0(\Gamma)$. Meanwhile,

$$\begin{aligned}
\partial(\widehat{[0]} \diamond \widehat{[0,1]}) &= -\widehat{[0]} \diamond \widehat{[0]} + \widehat{[0]} \diamond \widehat{[1]}, \\
\partial(\widehat{[0]} \diamond \widehat{[0,1]} + \widehat{[0,1]} \diamond \widehat{[1]}) &= -\widehat{[0]} \diamond \widehat{[0]} + \widehat{[1]} \diamond \widehat{[1]}, \\
\partial(\widehat{[0]} \diamond \widehat{[0,1]} + \widehat{[0,1]} \diamond \widehat{[1]} - \widehat{[1]} \diamond \widehat{[0,1]}) &= -\widehat{[0]} \diamond \widehat{[0]} + \widehat{[1]} \diamond \widehat{[0]},
\end{aligned}$$

and therefore

$$\left\{\widehat{[0]} \diamond \widehat{[0]} - \widehat{[0]} \diamond \widehat{[1]}, \widehat{[0]} \diamond \widehat{[0]} - \widehat{[1]} \diamond \widehat{[0]}, \widehat{[0]} \diamond \widehat{[0]} - \widehat{[1]} \diamond \widehat{[1]}\right\} \subset B_0(\Gamma).$$

In particular, all the elementary chains are homologous: $\widehat{[0]} \diamond \widehat{[0]} \sim \widehat{[0]} \diamond \widehat{[1]} \sim \widehat{[1]} \diamond \widehat{[0]} \sim \widehat{[1]} \diamond \widehat{[1]}$. Consider an arbitrary chain $z \in C_0(\Gamma)$. Then we have that

$$z = \alpha_1 \widehat{[0]} \diamond \widehat{[0]} + \alpha_2 \widehat{[0]} \diamond \widehat{[1]} + \alpha_3 \widehat{[1]} \diamond \widehat{[0]} + \alpha_4 \widehat{[1]} \diamond \widehat{[1]}.$$

From the homologies identified above, it follows that

$$\begin{aligned}
[z]_\Gamma &= \left[\alpha_1 \widehat{[0]} \diamond \widehat{[0]} + \alpha_2 \widehat{[0]} \diamond \widehat{[1]} + \alpha_3 \widehat{[1]} \diamond \widehat{[0]} + \alpha_4 \widehat{[1]} \diamond \widehat{[1]}\right]_\Gamma \\
&= \alpha_1 \left[\widehat{[0]} \diamond \widehat{[0]}\right]_\Gamma + \alpha_2 \left[\widehat{[0]} \diamond \widehat{[1]}\right]_\Gamma + \alpha_3 \left[\widehat{[1]} \diamond \widehat{[0]}\right]_\Gamma + \alpha_4 \left[\widehat{[1]} \diamond \widehat{[1]}\right]_\Gamma \\
&= (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4) \left[\widehat{[0]} \diamond \widehat{[0]}\right]_\Gamma
\end{aligned}$$

We can therefore think of every element of $H_0(\Gamma) = Z_0(\Gamma)/B_0(\Gamma)$ as being generated by $\widehat{[0]} \diamond \widehat{[0]}$ and therefore $\dim H_0(\Gamma) = 1$. It follows that $\beta_0 = 1$.

This example shows that homology groups essentially measure the number of k -dimensional holes within a set. $\beta_0$corresponds to the number of connected components, β_1 corresponds to the number of enclosed loop, β_2 corresponds to the number of enclosed cavities, etc. Here, $\beta_0 = 1$ agrees with the one connected component in Figure B.1, while $\beta_1 = 1$ agrees with the one enclosed loop in Figure B.1.

Bibliography

[1] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification.* Wiley, New York, 2nd edition, 2001.

[2] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Computer Science and Scientific Computing. Academic Press, Boston, 2nd edition, 1990.

[3] L.I. Kuncheva. *Combining Pattern Classifiers.* Wiley, New York, 2004.

[4] C.M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[5] A.R. Webb. *Statistical Pattern Recognition.* Wiley, 2012.

[6] B.D. Ripley. *Pattern Recognition and Neural Networks.* Cambridge University Press, 1996.

[7] G. Nagy. State of the art in pattern recognition. *Proceedings of the IEEE*, 56(5):836 863, 1968.

[8] L. Kanal. Patterns in pattern recognition: 1968-1974. *Information Theory, IEEE Transactions on*, 20(6):697 722, 1974.

[9] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: a review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4 37, Jan 2000.

[10] S. Watanabe. *Pattern recognition: human and mechanical.* Wiley-Interscience publication. Wiley, 1985.

[11] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1):245 271, 1997.

[12] A.K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. In P.R. Krishnaiah and L.N. Kanal, editors, *Classification Pattern Recognition and Reduction of Dimensionality*, volume 2 of *Handbook of Statistics*, pages 835 855. Elsevier, 1982.

[13] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507 2517, 2007.

[14] M. Dash and H. Liu. Feature selection for classification. *Intellectual Data Analysis*, 1(1-4):131 156, 1997.

[15] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, and A.K. Jain. Dimensionality reduction using genetic algorithms. *Evolutionary Computation, IEEE Transactions on*, 4(2):164 171, 2000.

[16] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

[17] J. Fan and J. Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101–148, 2010.

[18] E. Romero, J.M. Sopena, G. Navarrete, and R. Alquézar. Feature selection forcing overtraining may help to improve performance. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 2181 2186. IEEE, 2003.

[19] T. Lambrou, P. Kudumakis, R. Speller, M. Sandler, and A. Linney. Classification of audio signals using statistical features on time and wavelet transform domains. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 6, pages 3621 3624. IEEE, 1998.

[20] S.W. Smith. *Digital signal processing: a practical guide for engineers and scientists.* Newnes, 2003.

[21] D.R. Cox and D.V. Hinkley. *Theoretical Statistics.* Chapman & Hall, 1979.

[22] J.B. Harley and J.M.F. Moura. Guided wave temperature compensation with the scale-invariant correlation coefficient. In *Ultrasonics Symposium (IUS), 2011 IEEE International*, pages 1068 1071, Orlando, FL, October 2011. IEEE International Ultrasonics Symposium.

[23] J.B. Harley, Y. Ying, J.M.F. Moura, I.J. Oppenheim, and L. Sobelman. Application of Mellin transform features for robust ultrasonic guided wave structural health monitoring. In *AIP Conference Proceedings*, volume 1430, pages 1551 1558, 2012.

[24] I. Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, 1992.

[25] R.E. Learned and A.S. Willsky. A wavelet packet approach to transient signal classification. *Applied and Computational Harmonic Analysis*, 2(3):265 278, 1995.

[26] G.G. Yen and K.C. Lin. Wavelet packet feature extraction for vibration monitoring. *Industrial Electronics, IEEE Transactions on*, 47(3):650 667, 2000.

[27] X. Jin, S. Gupta, K. Mukherjee, and A. Ray. Wavelet-based feature extraction using probabilistic finite state automata for pattern classification. *Pattern Recognition*, 44(7):1343 1356, 2011.

[28] L. Gaul and S. Hurlebaus. Wavelet-transform to identify the location and force-time-history of transient load in a plate. In F.-K. Chang, editor, *Structural Health Monitoring*, pages 851–860, Lancaster, 1999. Technomic Publishing Co.

[29] A. Abbate, J. Koay, J. Frankel, S.C. Schroeder, and P. Das. Application of wavelet transform signal processor to ultrasound. In *Ultrasonics Symposium, 1994. Proceedings., 1994 IEEE*, volume 2, pages 1147–1152. IEEE, 1994.

[30] H. Sohn, C.R. Farrar, F.M. Hemez, D.D. Shunk, D.W. Stinemates, B.R. Nadler, and J.J. Czarnecki. A review of structural health monitoring literature: 1996-2001. Technical report, Los Alamos National Laboratory, Los Alamos, NM, 2004. Report Number LA-13976-MS.

[31] C. Lee and S. Park. Damage classification of pipelines under water flow operation using multi-mode actuated sensing technology. *Smart Materials and Structures*, 20(11):115002 115010, 2011.

[32] J. Min, S. Park, C.B. Yun, C.G. Lee, and C. Lee. Impedance-based structural health monitoring incorporating neural network technique for identification of damage type and severity. *Engineering Structures*, 39(0):210 220, 2012.

[33] H. Sohn, C.R. Farrar, N.F. Hunter, and K. Worden. Structural health monitoring using statistical pattern recognition techniques. *Journal of Dynamic Systems, Measurement, and Control*, 123(4):706–711, 2001.

[34] C. Biemans, W.J. Staszewski, C. Boller, and G.R. Tomlinson. Crack detection in metallic structures using piezoceramic sensors. *Key Engineering Materials*, 167:112 121, 1999.

[35] S. Legendre, D. Massicotte, J. Goyette, and T.K. Bose. Wavelet-transform-based method of analysis for Lamb-wave ultrasonic NDE signals. *Instrumentation and Measurement, IEEE Transactions on*, 49(3):524–530, 2000.

[36] J. Zou and J. Chen. A comparative study on time–frequency feature of cracked rotor by Wigner-Ville distribution and wavelet transform. *Journal of sound and vibration*, 276(1):1 11, 2004.

[37] J.D. Hou and M.K. Hinders. Dynamic wavelet fingerprint identification of ultrasound signals. *Materials Evaluation*, 60(9):1089 1093, 2002.

[38] M.K. Hinders, J. Hou, and J.C.P. McKeon. Ultrasonic inspection of thin multilayers. In *AIP Conference Proceedings*, volume 760, pages 1137–1144, 2005.

[39] M.K. Hinders and J. Hou. Ultrasonic periodontal probing based on the dynamic wavelet fingerprint. In *Review of Progress in Quantitative Nondestructive Evaluation: 24*, volume 760, pages 1549–1556, 2005.

[40] J.R. Hou, S.T. Rose, and M.K. Hinders. Ultrasonic periodontal probing based on the dynamic wavelet fingerprint. *EURASIP Journal on Applied Signal Processing*, 7:1137–1146, 2005.

[41] J.E. Lynch, M.K. Hinders, and G.B. McCombs. Clinical comparison of an ultrasonographic periodontal probe to manual and controlled-force probing. *Measurement*, 39(5):429–439, 2006.

[42] M. Hinders, R. Jones, K. Leonard, and K. Rudd. Wavelet thumbprint analysis of time domain reflectometry signals for wiring flaw detection. *Engineering intelligent systems for electrical engineering and communications*, 15(4):225–240, 2007.

[43] M.K. Hinders and J.P. Bingham. Lamb wave pipe coating disbond detection using the dynamic wavelet fingerprinting technique. In *AIP Conference Proceedings*, volume 1211, pages 615–622, 2010.

[44] J.P. Bingham and M.K. Hinders. Automatic multimode guided wave feature extraction using wavelet fingerprints. In *AIP Conference Proceedings*, volume 1211, pages 623–630, 2010.

[45] K.R. Leonard, E.V. Malyarenko, and M.K. Hinders. Ultrasonic Lamb wave tomography. *Inverse Problems*, 18(6):1795–1808, 2002.

[46] J. Hou, K.R. Leonard, and M.K. Hinders. Automatic multi-mode Lamb wave arrival time extraction for improved tomographic reconstruction. *Inverse Problems*, 20(6):1873–1888, 2004.

[47] J.M. Hyde, M.K. Miller, M.G. Hetherington, A. Cerezo, G.D.W. Smith, and C.M. Elliott. Spinodal decomposition in Fe-Cr alloys: Experimental study at the atomic level and comparison with computer models-III: Development of morphology. *Acta metallurgica et materialia*, 43(9):3415–3426, 1995.

[48] M. Allili, K. Mischaikow, and A. Tannenbaum. Cubical homology and the topological classification of 2d and 3d imagery. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 173–176. IEEE, 2001.

[49] M. Niethammer, A.N. Stein, W.D. Kalies, P. Pilarczyk, K. Mischaikow, and A. Tannenbaum. Analysis of blood vessel topology by cubical homology. In *Image Processing, 2002 International Conference on*, volume 2, pages II–969 II–972. IEEE, 2002.

[50] M. Zelawski. Pattern recognition based on homology theory. *Machine Graphics & Vision International Journal*, 14(3):309–324, 2005.

[51] D. Ziou and M. Allili. Generating cubical complexes from image data and computation of the euler number. *Pattern Recognition*, 35(12):2833–2839, 2002.

[52] V. de Silva and R. Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *The International Journal of Robotics Research*, 25(12):1205–1222, 2006.

[53] T. Teramoto and Y. Nishiura. Morphological characterization of the diblock copolymer problem with topological computation. *Japan journal of industrial and applied mathematics*, 27(2):175–190, 2010.

[54] M Mrozek, M Żelawski, A Gryglewski, S Han, and A Krajniak. Homological methods for extraction and analysis of linear features in multidimensional images. *Pattern Recognition*, 45(1):285–298, 2012.

[55] M. Gameiro and P. Pilarczyk. Automatic homology computation with application to pattern classification. *RIMS Kokyuroku Bessatsu*, B3:1–10, 2007.

[56] M. Gameiro, K. Mischaikow, and T. Wanner. Evolution of pattern complexity in the Cahn–Hilliard theory of phase separation. *Acta Materialia*, 53(3):693–704, 2005.

[57] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[58] S. Watanabe and S.K. Donovan. *Knowing and guessing: A quantitative study of inference and information.* Wiley New York, 1969.

[59] G. Kaiser. *A Frendly Guide to Wavelets.* Birkhäuser, 1994.

[60] The MathWorks Inc. *MATLAB.* Natick, Massachusetts, version 7.10.0 (R2010b) edition, 2010.

[61] K.R. Leonard and M.K. Hinders. Lamb wave tomography of pipe-like structures. *Ultrasonics*, 43(7):574–583, 2005.

[62] J. Bingham and M. Hinders. Lamb wave characterization of corrosion-thinning in aircraft stringers: Experiment and three-dimensional simulation. *The Journal of the Acoustical Society of America*, 126:103–113, 2009.

[63] C.A. Bertoncini. *Applications of Pattern Classification to Time-Domain Signals.* PhD dissertation, The College of William and Mary, Williamsburg, VA, 2010.

[64] C. Bertoncini. K. Rudd, B. Nousain, and M. Hinders. Wavelet fingerprinting of radio-frequency identification (RFID) tags. *Industrial Electronics, IEEE Transactions on*, 59(12):4843–4850, 2012.

[65] E.A. Dieckman and M.K. Hinders. Use of pattern classification algorithms to interpret acoustic echolocation data from a walking-speed robotic sensor platform. *The Journal of the Acoustical Society of America*, 132(3):1971, 2012.

[66] E.A. Dieckman and M.K. Hinders. Acoustic echolocation for mobile robots with parametric arrays. [Online]. Available: `http://www.acoustics.org/press/164th/Dieckman_2pSPa5.html`, October 2012.

[67] L. Rayleigh. On waves propagated along the plane surface of an elastic solid. *Proceedings of the London Mathematical Society*, 1(1):4–11, 1885.

[68] H. Lamb. On waves in an elastic plate. *Proceedings of the Royal Society of London. Series A*, 93(648):114–128, 1917.

[69] R.D. Mindlin. Influence of rotatory inertia and shear on flexural motions of isotropic, elastic plates. *Journal of Applied Mechanics*, 18:31–38, 1951.

[70] D.C. Worlton. Ultrasonic testing with Lamb waves. *Nondestructive Testing*, 158(4):218 222, August 1957.

[71] D.C. Worlton. Experimental confirmation of Lamb waves at megacycle frequencies. *Journal of Applied Physics*, 32(6):967–971, Jun 1961.

[72] J.L. Rose. *Ultrasonic Waves in Solid Media.* Cambridge University Press, 1999.

[73] I.A. Viktorov. *Rayleigh and Lamb Waves - Physical Theory and Applications.* Plenum Press, 1967.

[74] B.A. Auld. *Acoustic fields and waves in solids*, volume 2. RE Kreiger, 1990.

[75] S.K. Advani, L.J. Breon, and J.L. Rose. Guided wave thickness measurement tool development for estimation of thinning in platelike structures. In *American Institute of Physics Conference Series*, volume 1211, pages 207 214, 2010.

[76] H. Kannajosyula, P. Puthillath, C.J. Lissenden, and J.L. Rose. Interface waves for SHM of adhcsively bonded joints. *Structural Health Monitoring 2009: From System Integration to Autonomous Systems, Proceedings of 7th International Workshop on Structural Health Monitoring*, pages 247 274, 2009.

[77] W. Qu, L. Xiao, and Y. Zhou. Finite element simulation of Lamb wave with piezoelectric transducers for plasticity-driven damage detection. *Structural Health Monitoring 2009: From System Integration to Autonomous Systems, Proceedings of 7th International Workshop on Structural Health Monitoring*, pages 737–744, 2009.

[78] Y. Lu, L. Ye, D. Wang, and Z. Su. Lamb wave signals for damage detection using a probability-based imaging algorithm. *Structural Health Monitoring 2009: From System Integration to Autonomous Systems, Proceedings of 7th International Workshop on Structural Health Monitoring*, pages 1272–1279, 2009.

[79] C. Jayaraman, C.V. Krishnamurthy, and K. Balasubramaniam. Axial higher order modes cluster (A-HOMC) guided wave for pipe inspection. *AIP Conference Proceedings*, 1211(1):161–168, 2010.

[80] M. Ratassepp and M.J.S. Lowe. SH_0 guided wave interaction with a crack aligned in the propagation direction in a plate. *AIP Conference Proceedings*, 1096(1):161–168, 2009.

[81] K.F. Graff. *Wave Motion in Elastic Solids*. Dover Books on Engineering. Dover Publications, 1975.

[82] J.L. Rose. A baseline and vision of ultrasonic guided wave inspection potential. *Journal of Pressure Vessel Technology*, 124(3):273–282, August 2002.

[83] A. Raghavan and C.E.S. Cesnik. Review of guided-wave structural health monitoring. *The Shock and Vibration Digest*, 39(2):91–114, 2007.

[84] S.B. Kim and H. Sohn. Instantaneous reference-free crack detection based on polarization characteristics of piezoelectric materials. *Smart Materials and Structures*, 16(6):2375–2387, 2007.

[85] J. Ayers, N. Apetre, M. Ruzzene, and V. Sharma. Phase gradient-based damage characterization of structures. *Structural Health Monitoring 2009: From System Integration to Autonomous Systems, Proc. of 7th Int'l Workshop on Structural Health Monitoring*, pages 1113–1120, 2009.

[86] S.R. Anton. Baseline-free and self-powered structural health monitoring. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA, 2008.

[87] S. Park, S.R. Anton, D.J. Inman, J.K. Kim, and D.S. Ha. Instantaneous baseline damage detection using a low power guided waves system. *Structural Health Monitoring 2009: From System Integration to Autonomous Systems, Proceedings of 7th International Workshop on Structural Health Monitoring*, pages 505–512, 2009.

[88] D. Harris. The influence of human factors on operational efficiency. *Aircraft Engineering and Aerospace Technology*, 78(1):20-25, 2006.

[89] G. Marsh. Duelling with composites. *Reinforced Plastics*, 50(6):18-23, 2006.

[90] D. Bowermaster. Alaska isn't the only airline with ground-safety troubles. [Online]. Available: `http://seattletimes.com/html/businesstechnology/2002750657_alaska20.html`, January 2006.

[91] R. Bracewell. *The Hilbert Transform.* McGraw-Hill,, New York, 3rd edition, 1999.

[92] E.W.T. Ngai, K.K.L. Moon, F.J. Riggins, and C.Y. Yi. RFID research: An academic literature review (1995-2005) and future research directions. *International J. of Production Economics*, 112(2):510 520, 2008.

[93] A.E. Abdulhadi and R. Abhari. Design and experimental evaluation of miniaturized monopole UHF RFID tag antennas. *Antennas and Wireless Propagation Letters, IEEE*, 11:248-251, 2012.

[94] M.A. Khan, U.S. Bhansali, and H.N. Alshareef. High-performance non-volatile organic ferroelectric memory on banknotes. *Advanced Materials*, 24(16):2165-2170, 2012.

[95] A. Juels. RFID security and privacy: A research survey. *Selected Areas in Communications, IEEE Journal on*, 24(2):381 394, 2006.

[96] J. Halamka, A. Juels, A. Stubblefield, and J. Westhues. The security implications of verichip cloning. *Journal of the American Medical Informatics Association*, 13(6):601 607, 2006.

[97] T. Heydt-Benjamin, D. Bailey, K. Fu, A. Juels, and T. O'Hare. Vulnerabilities in first-generation RFID-enabled credit cards. In S. Dietrich and R. Dhamija, editors, *Financial Cryptography and Data Security*, volume 4886 of *Lecture Notes in Computer Science*, pages 2 14. Springer Berlin / Heidelberg, 2007.

[98] H. Richter, W. Mostowski, and E. Poll. Fingerprinting passports. In *NLUUG Spring Conference on Security*, pages 21-30, 2008.

[99] D. White. NCR: RFID in retail. *RFID: Applications, Security, and Privacy*, pages 381 395, 2005.

[100] J. Westhues. Hacking the prox card. *RFID: Applications, Security, and Privacy*, pages 291 300, 2005.

[101] Smart Card Alliance. Proximity mobile payments: Leveraging NFC and the contactless financial payments infrastructure. *Whitepaper*, 2007.

[102] E. Léopold. The future of mobile check-in. *Journal of Airport Management*, 3(3):215 222, 2009.

[103] M.H. Wang, J.F. Liu, J. Shen, Y.Z. Tang, and N. Zhou. Security issues of RFID technology in supply chain management. *Advanced Materials Research*, 490:2470 2474, 2012.

[104] A. Juels. Yoking-proofs for RFID tags. In *"Proc. Second Annu. IEEE Pervasive Comput. and Commun. Workshops (PERCOMW 2004)*, PERCOMW '04, pages 138 143, March 2004.

[105] A. Juels. Strengthening epc tags against cloning. In *Proceedings of the 4th ACM workshop on Wireless security*, WiSe '05, pages 67- 76. ACM, 2005.

[106] M.J. Riezenman. Cellular security: better, but foes still lurk. *Spectrum, IEEE*, 37(6):39 - 42, 2000.

[107] W.C. Suski, M.A. Temple, M.J. Mendenhall, and R.F. Mills. Using spectral fingerprints to improve wireless network security. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1-5. IEEE, 2008.

[108] R. M. Gerdes, M. Mina, S. F. Russell, and T. E. Daniels. Physical-layer identification of wired ethernet devices. *Information Forensics and Security, IEEE Transactions on*, 7(4):1339 1353, aug. 2012.

[109] I. O. Kennedy, P. Scanlon, F. J. Mullany, M. M. Buddhikot, K. E. Nolan, and T. W. Rondeau. Radio transmitter fingerprinting: A steady state frequency domain approach. In *Proc. IEEE 68th Vehicular Technology Conf. (VTC 2008)*, pages 1-5, 2008.

[110] B. Danev, T.S. Heydt-Benjamin, and S. Čapkun. Physical-layer identification of RFID devices. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 199 214, Berkeley, CA, USA, 2009. USENIX Association.

[111] D. Zanetti, B. Danevs, and S. Čapkun. Physical-layer identification of UHF RFID tags. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 353-364, New York, NY, USA, 2010. ACM.

[112] H. P. Romero, K. A. Remley, D. F. Williams, and C.-M. Wang. Electromagnetic measurements for counterfeit detection of radio frequency identification cards. *Microwave Theory and Techniques, IEEE Transactions on*, 57(5):1383 1387, May 2009.

[113] EPCglobal Inc. *EPC Radio-Frequency Identity Protocols: Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz Version 1.2.0*, October 2008.

[114] GNU Radio Website. Software available at [`http://www.gnuradio.org`], accessed January 2011.

[115] K. J. Ellis and N. Serinken. Characteristics of radio transmitter fingerprints. *Radio Science*, 36(4):585–597, 2001.

[116] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, Boston, MA, 1992.

[117] Y. Feng and F.S. Schlindwein. Normalized wavelet packets quantifiers for condition monitoring. *Mechanical Systems and Signal Processing*, 23(3):712–723, April 2009.

[118] M.R. Spiegel and L.J. Stephens. *Schaum's Outline of Statistics: Theory and Problems of Statistics*. Schaum's Outline Series. McGraW-Hill, 1999.

[119] A. de Sena and D. Rocchesso. A fast Mellin and scale transform. *EURASIP Journal on Advances in Signal Processing*, 2007(1):089170, April 2007.

[120] H. Sundaram, S.D. Joshi, and R.K.P. Bhatt. Scale periodicity and its sampling theorem. *IEEE Transactions on Signal Processing*, 45(7):1862–1865, July 1997.

[121] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[122] G. M. Weiss and F. Provost. The effect of class distribution on classifier learning:. Technical Report ML-TR-44, Rutgers University, Department of Computer Science, 2001.

[123] Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the* 14^{th} *International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann Publishers, Inc, 1997.

[124] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Applications of Mathematics. Springer, 1996.

[125] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Academic Press, 1999.

[126] R.P.W. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D.M.J. Tax, and S. Verzakov. *PRTools4.1, A Matlab Toolbox for Pattern Recognition*, 2007.

[127] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[128] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861 874, 2006.

[129] J.A. Hanley and B.J. McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839 843, 1983.

[130] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[131] J.M. Lobo, A. Jiménez-Valverde, and R. Real. Auc: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, 17(2):145 151, 2007.

[132] B. Hanczar, J. Hua, C. Sima, J. Weinstein, M. Bittner, and E.R. Dougherty. Small-sample precision of ROC-related estimates. *Bioinformatics*, 26(6):822 830, 2010.

[133] D.J. Hand. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine learning*, 77(1):103–123, 2009.

[134] K.V.S. Rao, P.V. Nikitin, and S.F. Lam. Antenna design for UHF RFID tags: A review and a practical application. *Antennas and Propagation, IEEE Transactions on*, 53(12):3870 3876, 2005.

[135] V. Giurgiutiu. Damage assessment of structures-an air force office of scientific research structural mechanics perspective. *Key Engineering Materials*, 347:69 74, 2007.

[136] R. Carandente and P. Cawley. A method to estimate the size of corrosion patches with guided waves in pipes. *AIP Conf Proc*, 1430(1):1929–1936, 2012.

[137] M. Hinders, E. Malyarenko, and J. McKeon. Contact scanning Lamb wave tomography. *The Journal of the Acoustical Society of America*, 104(3):1790–1791, 1998.

[138] J.C.P. McKeon and M.K. Hinders. Parallel projection and crosshole Lamb wave contact scanning tomography. *The Journal of the Acoustical Society of America*, 106(5):2568–2577, 1999.

[139] J.C.P. McKeon. *Tomography Applied to Lamb Wave Contact Scanning.* PhD dissertation, The College of William and Mary, Williamsburg, VA, 1998.

[140] J.C.P. McKeon and M.K. Hinders. Lamb wave scattering from a through hole. *Journal of Sound and Vibration*, 224(5):843 862, 1999.

[141] J.C.P. McKeon and M.K. Hinders. Lamb wave contact scanning tomography. In *Review of Progress in Quantitative Nondestructive Evaluation*, volume 18, pages 951–958, 1999.

[142] E.V. Malyarenko. *Lamb Wave Diffraction Tomography.* PhD dissertation, The College of William and Mary, Williamsburg, VA, 2000.

[143] E.V. Malyarenko and M.K. Hinders. Fan beam and double crosshole Lamb wave tomography for mapping flaws in aging aircraft structures. *Journal of the Acoustical Society of America*, 108(4):1631 1639, 2000.

[144] E.V. Malyarenko and M.K. Hinders. Ultrasonic Lamb wave tomography. *Ultrasonics*, 39(4):269 281, 2001.

[145] K.R. Leonard and M.K. Hinders. Guided wave helical ultrasonic tomography of pipes. *Journal of the Acoustical Society of America*, 114(2):767 774, 2003.

[146] K.R. Leonard. *Ultrasonic Guided Wave Tomography of Pipes.* PhD dissertation, The College of William and Mary, 2004.

[147] K.R. Leonard and M.K. Hinders. Lamb wave helical ultrasonic tomography of pipes. In *Review of Progress in Quantitative Nondestructive Evaluation*, volume 23, pages 173 179, 2004.

[148] M.K. Hinders, K.R. Leonard, and E.V. Malyarenko. Blind test of Lamb wave diffraction tomography. *Review of Progress in Quantitative Nondestructive Evaluation*, 21:278 283, 2002.

[149] K.R. Leonard. Milti-mode Lamb wave tomography with arrival time sorting. *Journal of the Acoustical Society of America*, 117(4):2028–2038, 2005.

[150] K.R. Leonard and M.K. Hinders. Lamb wave tomography of pipes and tanks using frequency compounding. In *Review of Progress in Quantitative Nondestructive Evaluation*, volume 24, pages 867 874, 2005.

[151] C.A.C. Leckey, M.D. Rogge, C.A. Miller, and M.K. Hinders. Multiple-mode Lamb wave scattering simulations using 3D elastodynamic finite integration technique. *Ultrasonics*, 52(2):193 207, 2012.

[152] Z. Sun and C.C. Chang. Structural damage assessment based on wavelet packet transform. *J. Struct. Eng.*, 128(10):1354–1361, 2002.

[153] S.G. Mattson and S.M. Pandit. Statistical moments of autoregressive model residuals for damage localisation. *Mechanical Systems and Signal Processing*, 20(3):627 645, 2006.

[154] S.S. Kessler and P. Agrawal. Application of pattern recognition for damage classification in composite laminates. In *Proceedings of the 6th International Workshop on SHM*, pages 11 14, Stanford University, September 2007.

[155] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119 – 1125, 1994.

[156] P. Allison, M. Altman, J. Gill, and M.P. McDonald. Convergence problems in logistic regression. *Numerical Issues in Statistical Computing for the Social Scientist*, pages 238–252, 2004.

[157] G.E. Muller and C.J. Schmid. Factor of safety-USAF design practice. Technical report, DTIC Document, 1978.

[158] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology*. Applied Mathematical Sciences. Springer, 2004.

[159] Computational Homology Project (CHomP) Website. Software available at `http://chomp.rutgers.edu`, accessed September 2012.

[160] J. Dumas, F. Heckenbach, B.D. Saunders, and V. Welker. GAP homology package. Software available at `http://www.linalg.org/gap.html`, 2008.

[161] A. Tausz, M. Vejdemo-Johansson, and H. Adams. Javaplex: A research software package for persistent (co)homology. Software available at `http://code.google.com/javaplex`, 2011.

[162] D. Morozov. Dionysus library for computing persistent homology. Software available at `http://www.mrzv.org/software/dionysus`, 2012.

[163] CAPD:RedHom Library : Reduction homology algorithms. Software available at `http://redhom.ii.uj.edu.pl`.

[164] W. S. Slaughter. *The Linearized Theory of Elasticity*. Birkhäuser, Boston, 2001.

[165] K E Graff. *Wave Motion in Elastic Solids*. Dover, New York, 1991.

[166] W. Ostachowicz, P. Kudela, M. Krawczuk, and A. Zak. *Guided Waves in Structures for SHM: The Time-Domain Spectral Element Method*. Wiley, New Jersey, 2012.

[167] J.J. Ditri, J.L. Rose, and G. Chen. Mode selection criteria for defect detection optimization using Lamb waves. In *Review of Progress in Quantitative Nondestructive Evaluation. Vol. 11B*, volume 11, pages 2109–2115, 1992.

Vita

Corey Miller was born in Bethlehem, Pennsylvania on May 27, 1986. He attended Essex High School in Essex Junction, Vermont where he graduated in June, 2004. Corey received his Bachelor of Science degree in both Physics and Applied Mathematics from The College of William and Mary located in Williamsburg, Virginia in May of 2008. He was also a member of the mathematics honor society Pi Mu Epsilon. Corey continued his graduate studies in the Applied Science Department at The College of William and Mary, where he worked as a Research Assistant in the Nondestructive Evaluation laboratory.