ECR Bridge Decks: Damage Detection and Assessment of Remaining Service Life for Various Overlay Repair Options — Part I

Report to the Michigan Department of Transportation

by

Ronald S. Harichandran
Professor and Chairperson

and

Gang Zhang
Graduate Student

Department of Civil and Environmental Engineering
Michigan State University
East Lansing, MI 48824-1226

Tel: (517) 355-5107
Fax: (517) 432-1827
E-Mail: harichan@egr.msu.edu
Web: www.egr.msu.edu/cee/~harichan

January 2011

Technical Report Documentation Page

1. Report No. Research Report RC-1502	2. Government Accession No.	3. MDOT Project Manager : Steve Kahl
4. Title and Subtitle: ECR Bridge Decks: Damage Detection and Assessment of Remaining Service Life for Various Overlay Repair Options - Part I		5. Report Date: 11/10/2010
7. Author(s): Ronald Harichandran & Gang Zhang		6. Performing Org. Code
9. Performing Organization Name and Address Department of Civil and Environmental Engineering Michigan State University East Lansing, MI 4824-1226		8. Performing Org. Report No.
12. Sponsoring Agency Name and Address Michigan Department of Transportation Construction and Technology Division P.O. Box 30049 Lansing, MI 48909		10. Work Unit No. (TRAIS)
		11. Contract Number: 2002-0532
		11(a). Authorization Number: Z19
15. Supplementary Notes		13. Type of Report & Period Covered Research Report, 9/29/06-6/30/10
		14. Sponsoring Agency Code

16. Abstract:

Delamination of the concrete cover above upper reinforcing bars is a common problem in concrete bridge decks. The delamination is typically initiated by corrosion of the upper reinforcing bars and promoted by freeze-thaw cycling and traffic loading. The detection of delamination is important for bridge maintenance and acoustic non-destructive evaluation (NDE) is widely used due to its low cost, speed, and easy implementation. In traditional acoustic approaches, the inspector sounds the surface of the deck by impacting it with a bar or by dragging a chain, and assesses delamination by the "hollowness" of the sound. The acoustic signals are often contaminated by traffic and ambient noise at the site and the detection is highly subjective. The operator also needs to be well trained.

The performance of acoustic NDE methods can be improved by employing a suitable noise-cancelling algorithm and a reliable detection algorithm that eliminates subjectivity. Since the noise is non-stationary and unpredictable, the algorithms should be adaptive. After evaluating different noise cancelling algorithms based on a numerical performance criterion and through visual inspection, a noise cancelling algorithm using a modified independent component analysis (ICA) is used to separate the sounding signals from recordings in a noisy environment. Different feature extraction algorithms were used to extract features of the filtered signals and their performance was evaluated using repeatability, separability and mutual information measures. Mel-frequency cepstral coefficients (MFCC) were identified as the best features for detection. The extracted features were further reduced based on the mutual information value to reduce the negative effect of features with poor separability. The selected features were used to train classifiers and the trained classifiers were used to classify new signals. The error rate was used to evaluate the performance of different classifiers. Radial basis function neural network had the lowest error rate and was selected as the classifier for field applications.

The proposed noise-cancelling and delamination detection algorithms were implemented into a seamless software containing MATLAB, LabVIEW and C/C++ modules. The performance of the system was verified using both experimental and field data. The proposed system showed good noise robustness. The performance of the system was satisfactory if there is sufficient available data for training and the selection of the training data is representative.

17. Key Words:delamination, noise cancelling, feature extraction, pattern recognition.	18. Distribution Statement No restrictions. This document is available to the public through the Michigan Department of Transportation.

19. Security Classification (report) Unclassified	20. Security Classification (Page) Unclassified	21. No of Pages 160	22. Price

TABLE OF CONTENTS

TABLE OF CONTENTS i

LIST OF FIGURES iv

LIST OF TABLES vi

EXECUTIVE SUMMARY vii

CHAPTER 1: INTRODUCTION 1

1.1 MOTIVATION 1
1.2 PROBLEM STATEMENT 2
1.3 RESEARCH OBJECTIVES 3
1.4 ORGANIZATION OF THE REPORT 4

CHAPTER 2: LITERATURE REVIEW 7

2.1 DAMAGE IN CONCRETE 7
2.1.1 Crack *7*
2.1.2 Honeycombing *8*
2.1.3 Delamination *9*
2.2 NON-DESTRUCTIVE EVALUATION (NDE) METHODS FOR CONCRETE 10
2.2.1 Impact Echo and Impulse Response *10*
2.2.2 Ultrasonic Methods *14*
2.2.3 Ground Penetrating Radar *17*
2.2.4 Infrared Thermography *20*
2.2.5 X-ray Imaging *22*
2.2.6 Sounding Methods *23*
2.3 SUMMARY 26

CHAPTER 3: NOISE CANCELLING ALGORITHMS 29

3.1 EVALUATION CRITERIA FOR NOISE CANCELLING ALGORITHMS 29
3.2 SPECTRAL SUBTRACTION 31
3.2.1 Theoretical Background *31*
3.2.2 Performance Evaluation *33*
3.3 ADAPTIVE FILTERS 33
3.3.1 Theoretical Background *33*
3.3.2 Performance Evaluation *37*
3.4 INDEPENDENT COMPONENT ANALYSIS 43
3.4.1 Theoretical Background *43*
3.4.2 Performance Evaluation *48*
3.5 MODIFIED ICA 51
3.5.1 Theory Background and Procedures *51*
3.5.2 Performance Evaluation *55*
3.6 SELECTION OF NOISE CANCELLING ALGORITHMS 58
3.7 SUMMARY 59

CHAPTER 4: FEATURE EXTRACTION 63

4.1 FEATURE EXTRACTION OF ACOUSTIC SIGNALS 63
4.1.1 Sub-band Energy 64
4.1.2 Energy of Wavelet Packet Tree 65
4.1.3 Psycho-Acoustic Features 68
4.1.4 Principal Component Analysis 71
4.1.5 Independent Component Analysis 73
4.2 PERFORMANCE OF DIFFERENT FEATURES 74
4.2.1 Criteria for Evaluation 74
4.2.2 Performance of Sub-band Energy 77
4.2.3 Performance of the Wavelet Packet Tree 81
4.2.4 Performance of MFCC 84
4.2.5 Performance of Features Extracted by PCA 86
4.2.6 Performance of Features Extracted by ICA 89
4.2.7 Summary of the Section 92
4.3 SELECTION OF THE FEATURE EXTRACTION ALGORITHM 92
4.3.1 Algorithm Selection Based on Weighted Rank 93
4.3.2 Algorithm Selection Based on Error Rates 96
4.4 SUMMARY 98

CHAPTER 5: PATTERN RECOGNITION AND DELAMINATION DETECTION 101

5.1 DETECTION ALGORITHMS 102
5.1.1 Bayesian-Based Classifier 102
5.1.2 Support Vector Machine 105
5.1.3 Multi-Layer Perceptron 108
5.1.4 Radial Basis Function 113
5.2 PERFORMANCE EVALUATION 115
5.2.2 Performance of Bayesian Classifier 117
5.2.3 Performance of Support Vector Machine 119
5.2.4 Performance of Multi-Layer Perceptron 121
5.2.5 Performance of Radial Basis Function 124
5.2.6 Selection of Detection Algorithm 126
5.2.7 Error Rate for Multiple Impacts 127
5.3 SUMMARY 131

CHAPTER 6: DELAMINATION DETECTION SYSTEM DEVELOPMENT AND ALGORITHM VERIFICATION 135

6.1 HARDWARE DEVELOPMENT 135
6.2 SOFTWARE DEVELOPMENT 136
6.2.1 Training Process 137
6.2.2 Inspection Process 138
6.2.3 Crank Noise Removal 140
6.2.4 Implementation of the Algorithms 142
6.3 VERIFICATION OF ALGORITHMS 144
6.3.1 Lab Experiments 146

6.3.2 Field Inspection *149*
6.4 SUMMARY 152

CHAPTER 7: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK 155

7.1 SUMMARY OF THE STUDY 155
7.2 MAJOR CONCLUSIONS 157
7.3 RECOMMENDATIONS FOR FUTURE WORK 159

BIBLIOGRAPHY 161

LIST OF FIGURES

Figure 2.1 Cracks in Concrete 8
Figure 2.2 Honeycombing in Concrete 9
Figure 2.3 Delamination in Concrete 10
Figure 2.4 Principle of the Impact Echo Method 12
Figure 2.5 Example spectrum from Impact Echo Tests [15] 12
Figure 2.6 Ultrasonic Pulse Velocity Method 16
Figure 2.7 Ultrasonic Tomography [20] 17
Figure 2.8 Ground Penetrating Radar 19
Figure 2.9 Results from Ground Penetrating Radar [9] 19
Figure 2.10 Example Image from an Infrared Camera 21
Figure 2.11 X-ray Imaging of Concrete Samples [21] 23
Figure 2.12 Chain Drag Test 24
Figure 2.13 Spectrogram of Chain Drag Tests 25
Figure 3.1 Performance of Spectral Subtraction 34
Figure 3.2 Noise Cancelling using Adaptive Filter [33] 35
Figure 3.3 Performance of RLS Adaptive Filter (Case 1) 39
Figure 3.4 Performance of RLS Adaptive Filter (Case 2) 40
Figure 3.5 Performance of RLS Adaptive Filter (Case 3) 41
Figure 3.6 Performance of RLS Adaptive Filter (Case 4) 42
Figure 3.7 Performance of RLS Adaptive Filter (Case 5) 43
Figure 3.8 Independent Component Analysis 44
Figure 3.9 Performance of EFICA for Instantaneous Mixture 50
Figure 3.10 Performance of EFICA for Convolutive Mixture 51
Figure 3.11 Modified ICA [39] 55
Figure 3.12 Performance of Modified ICA for Instantaneous Mixture 57
Figure 3.13 Performance of Modified ICA for Convolutive Mixture 57
Figure 4.1 Rectangular Filter Bank 65
Figure 4.2 Wavelet Decomposition [44] 67
Figure 4.3 Wavelet Packet Decomposition [45] 68
Figure 4.4 Hamming Window 70
Figure 4.5 Mel-Frequency Filter Banks 70
Figure 4.6 Repeatability of the Sub-band Energy 78
Figure 4.7 Separability of the Sub-band Energy 79
Figure 4.8 REP of the Sub-band Energy 80
Figure 4.9 SEP of the Sub-band Energy 80
Figure 4.10 Mutual Information of the Sub-band Energy 81
Figure 4.11 AAR Wavelet 82
Figure 4.12 Repeatability of the WP Tree 82
Figure 4.13 Separability of the WP Tree 82
Figure 4.14 REP of the Wavelet Packet Tree 83
Figure 4.15 SEP of the Wavelet Packet Tree 83
Figure 4.16 Mutual Information of the Wavelet Packet Tree 84
Figure 4.17 Repeatability of the MFCC 84

Figure 4.18 Separability of the MFCC 85
Figure 4.19 REP of the MFCC 85
Figure 4.20 SEP of the MFCC 86
Figure 4.21 Mutual Information of the MFCC 86
Figure 4.22 Repeatability of the PCA 87
Figure 4.23 Separability of the PCA 87
Figure 4.24 REP of the PCA 88
Figure 4.25 SEP of the PCA 89
Figure 4.26 Mutual Information of the PCA 89
Figure 4.27 Repeatability of the ICA 90
Figure 4.28 Separability of the ICA 90
Figure 4.29 REP of the ICA 91
Figure 4.30 SEP of the ICA 91
Figure 4.31 Mutual Information of the ICA 92
Figure 4.32 REP of different algorithms 94
Figure 4.33 SEP of different algorithms 94
Figure 4.34 Mutual Information of different algorithms 95
Figure 5.1 Threshold of Bayesian Classifiers 103
Figure 5.2 Support Vector Machine 106
Figure 5.3 Multi-Layer Perceptron 109
Figure 5.4 Signal-Flow Graph of the Perceptron 110
Figure 5.5 Architecture of Radial Basis Function Network 113
Figure 5.6 Variation of Error Rate due to Random Selection 117
Figure 5.7 Performance of Linear Bayesian Classifier 118
Figure 5.8 Performance of Quadratic Bayesian Classifier 119
Figure 5.9 Performance of Linear Kernel SVM Classifier 120
Figure 5.10 Performance of Quadratic Kernel SVM Classifier 121
Figure 5.11 Log-Sigmoid Activation Function 122
Figure 5.12 Performance of MLP with Different Structures 123
Figure 5.13 Performance of MLP44 123
Figure 5.14 Effect of Number of Neurons on RBF 124
Figure 5.15 Effect of the Variance of RBF 125
Figure 5.16 Performance of RBF Classifier 126
Figure 5.17 Comparison of Different Classifiers 127
Figure 5.18 Error Rate of Multiple Impacts 128
Figure 6.1 Proto-type of the Impacting Cart 136
Figure 6.2 Flow Chart of the Training Process 139
Figure 6.3 Flow Chart of the Inspection Process 141
Figure 6.4 Crank Removal 143
Figure 6.5 Data Communication 144
Figure 6.6 GUI forTraining Module 145
Figure 6.7 GUI for Inspection Module 146
Figure 6.8 Slab for Lab Experiment 147
Figure 6.9 Side View of the Slab 147
Figure 6.10 Barnes over US127 (Bridge 1) 150
Figure 6.11 Sitts over US127 (Bridge 2) 150

LIST OF TABLES

Table 2.1 Summary of Different Non-destructive Evaluation Methods 27
Table 3.1 Comparison of Noise Cancelling Algorithms..... 58
Table 3.2 Performance for Convolutive Mixtures. 59
Table 4.1 Rank of Different Feature Extraction Algorithms 96
Table 4.2 Error Rate of Different Feature Extraction Algorithms..... 97
Table 6.1 Error Rate under Different Noise Levels 149
Table 6.2 Error Rates of Original Signals and Filtered Signals..... 151
Table 6.3 Error Rates under Different Training Sets 152

Executive Summary

Delamination of the concrete cover above the upper reinforcing bars is a common problem in concrete bridge decks. The delamination is typically initiated by corrosion of the upper reinforcing bars and promoted by freeze-thaw cycling and traffic loading. The detection of delamination is important for bridge maintenance and acoustic non-destructive evaluation (NDE) is widely used due to its low cost, speed, and easy implementation. In traditional acoustic approaches, the inspector sounds the surface of the deck by impacting it with a hammer or bar, or by dragging a chain, and assesses delamination by the "hollowness" of the sound. The acoustic signals are often contaminated by traffic and ambient noise at the site and the detection is highly subjective.

The performance of acoustic NDE methods can be improved by employing a suitable noise-cancelling algorithm and a reliable detection algorithm that eliminates subjectivity. Since the noise is non-stationary and unpredictable, the algorithms should be adaptive. After evaluating different noise cancelling algorithms based on a numerical performance criterion and through visual inspection, a noise cancelling algorithm using a modified independent component analysis (ICA) was used to separate the sounding signals from recordings in a noisy environment. After the noise signals and the impact signals were successfully separated, the features of filtered signal were extracted. Different feature extraction algorithms were used to extract features of the filtered signals. The performance of different feature extraction algorithms were evaluated against repeatability, separability and mutual information which measures the information about

the condition of the concrete bridge deck. Mel-frequency cepstral coefficients (MFCC) were used as features for detection. The extracted features were further selected based on the value of the mutual information to reduce the negative effect of features with poor separability. The features selected were used to train the classifiers and the trained classifiers were used to classify new signals. The error rate was used to evaluate the performance of different classifiers. The radial basis function neural network had the lowest error rate and was selected as the classifier for field application.

The proposed noise-cancelling and delamination detection algorithms were then implemented using mixed-language programming in MATLAB, LabVIEW and C/C++. The performance of the system was verified using both experimental and field data. The proposed system showed good noise robustness. The performance of the system was satisfactory when there was sufficient available data for training and the selection of the training data was representative.

CHAPTER 1

INTRODUCTION

1.1 Motivation

After years of usage, aging of structures is inevitable and has now become one of the most severe problems facing the infrastructure in the United States. According to the American Society of Civil Engineers (ASCE), more than 26% of the America's bridges are either structurally deficient or functionally obsolete and an annually investment of $17 billion is needed to improve the current bridge conditions [2]. The priority has shifted from building new structures to inspection, assessment and maintenance of existing structures [3]. As the designers of these structures, civil engineers are required not only know how to design structures with sufficient strength at least cost, but also to understand that maintenance and rehabilitation of the structures is as important as the design of the structure because repair of bridges while in service can result in expensive replacement costs and user delays. Therefore, it is of vital importance to detect damages and defects and have them repaired before they progress and lead to structural or functional failures.

Reinforced concrete bridge decks are continuously degraded due to normal traffic and environmental exposure. This degradation is exacerbated in climatic regions where de-icing chlorides are used and in coastal regions where bridges are exposed to high salt air concentrations. Delamination is the major form of deck distress. This type of damage usually initiates underneath the surface due to corrosion of the steel reinforcement and

freeze-thaw and cannot be easily detected by visual inspection. With time, the delamination propagates and leads to spalling of the bridge deck. Small delaminated areas can be repaired by patching the affected area. A very large area of delamination will usually result in the replacement of the entire deck, which is expensive and causes significant user delay. It is therefore necessary to detect delamination at an early stage to reduce the cost of repair.

In order to detect delamination and evaluate the condition of a bridge deck such that appropriate repair or rehabilitation measures can be taken, effective non-destructive techniques that can provide information about the damage location and damage type is needed.

1.2 Problem Statement

Many methods have been considered for the inspection of bridge deck systems including impact echo, ultrasonic pulse velocity, ground penetrating radar, infrared thermography, X-ray imaging and sounding methods. Although many of these methods have been successfully used in detecting delamination and other defects in bridge decks, the different methods have their own advantages and limitations. A detailed comparison between these techniques is described in Chapter 2.

Sounding methods have the advantages of being fast, simple and inexpensive when compared with other more sophisticated techniques. However traditional sounding methods have several problems. First, the detection is subjective and operator dependent. Second, the effectiveness of the method is affected by the level of ambient noise. Although several attempts have been made to improve the performance of traditional

sounding methods, only modest improvements have been made and the results are still not satisfactory. It is therefore desirable to develop an improved acoustic method that can be used directly by bridge inspectors for bridge deck inspection. An effective method should be able to overcome ambient, not require the subjective judgment of a well-trained operator and be fast, automatic and robust.

1.3 Research Objectives

The major task of this research is to develop an automated inspection system to accurately detect delamination in concrete bridge decks. This can be achieved by accomplishing the following objectives:

1. Develop a noise cancelling algorithm that can cancel or separate the ambient noise from soundings. There are different noise cancelling algorithms and each algorithm has its own range of application. A criterion is needed to evaluate the performance of different algorithms so that the optimal algorithm can be selected.

2. Develop algorithms that can differentiate between soundings on the solid concrete and those on a delaminated concrete. These algorithms must be robust and fast.

3. Develop an automatic delamination detection system so that the operator can perform the detection on-site. The system must be easy to use, not require extensive set-up, and be fast so that lane closure durations are minimized.

1.4 Organization of the Report

Part 1 of this report is organized into seven chapters.

Chapter 1 provides an introduction of why the research on delamination detection is needed, describes what problem is solved, and presents the research objectives.

Chapter 2 provides a literatures review on the type of damage in concrete bride decks and on the techniques that are available to detect these defects and damages. A comparison of different non-destructive evaluation (NDE) based on previous research is also included.

Chapter 3 compares and evaluates the performance of different noise cancelling algorithms. Numerical criteria are first developed for the comparisons. Detailed theoretical background and derivation about each algorithm are included and their performance is evaluated. The most efficient noise cancelling algorithm based on a modified independent component analysis (ICA) is selected to separate the ambient noise from the recordings.

Chapter 4 deals with the problem of dimension reduction to facilitate delamination detection. Different models are used to extract features of the signals and the number of extracted features is further reduced based on mutual information so that those features that are most useful for delamination detection are retained. Mel-frequency cepstral coefficients (MFCCs) are selected as the best features for delamination detection.

In Chapter 5, an algorithm is developed for delamination detection. This task is formulated as a classification problem. First, the theoretical background of different

classifiers is described. The performance of different classifiers is compared and evaluated and the best classifier is selected based on weighted rank and the error rate of test samples.

Chapter 6 describes the development and verification of the hardware and software components of the automatic delamination detection system. Various components in the software for collecting and processing the data and detecting delamination are described. The inspection and training process used is also described. Field data from two bridges and tests performed on a full-scale slab constructed in the lab were used to verify the performance of the system. A brief discussion on how to select the training set to improve the performance is also included.

Chapter 7 provides a summary of the findings of part 1 of this research and gives recommendation for further research directions.

(This page intentionally left blank.)

CHAPTER 2

LITERATURE REVIEW

Various types of defects and damage may be caused in concrete structures due to environmental factors. The presence of damage in concrete may significantly reduce the strength, service life and the integrity of structures. Detecting concrete damage at an early stage can reduce maintenance costs. In order to have a better understanding of the types of damage in concrete and the methods to detect them, this chapter briefly describes common defects in concrete and then provides a review of existing non-destructive evaluation methods.

2.1 Damage in Concrete

Cracks, voids and delaminations are considered the most serious types of damage in concrete, especially in concrete bridges [4]. This section mainly describes the cause of damage as well as the effect of the damage on structures.

2.1.1 Crack

Cracks are the most commonly observed type of damage in concrete because of the low tensile strength of concrete. Figure 2.1 shows a typical crack in concrete. Cracks can be caused by shrinkage of the concrete, freeze-thaw cycling of the moisture, chemical reactions inside the concrete (such as alkali-silica reaction (ASR)) as well as loading. The presence of cracks may affect the performance of concrete structures. Cracks open a path for water to penetrate and accelerate damage due to freeze-thaw cycling in cold regions.

When de-icing salts are used or in marine environments, cracks enable the rapid ingression of chloride ions that accelerates corrosion of the steel reinforcement which leads to expansion and further opening of cracks. Continuously developing cracks may even affect the integrity of the entire structural system. The simplest way to detect the cracks is through visual inspection. Other sophisticated methods such as impact echo [6] and ultrasonic pulse velocity [7] may be used.

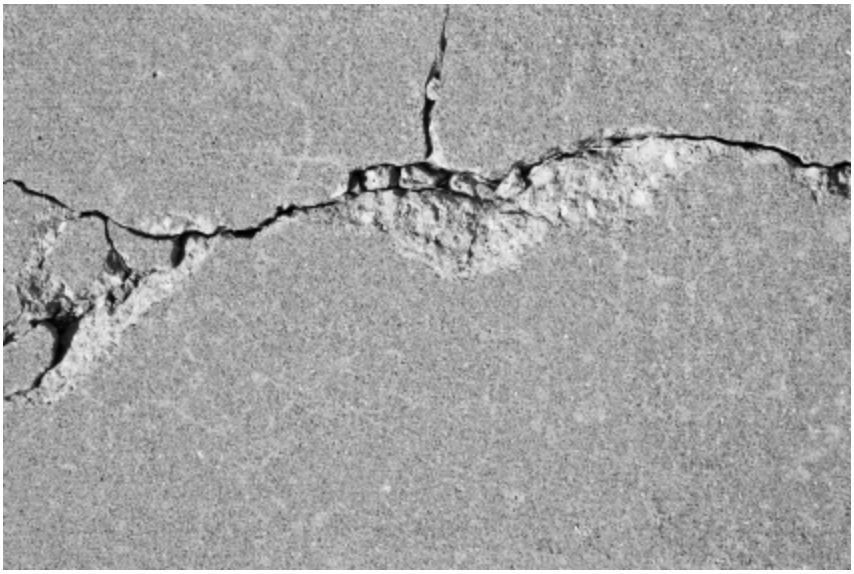


Figure 2.1 Cracks in Concrete
(http://www.silverspringsconcrete.com/concrete-question/)

2.1.2 Honeycombing

Honeycombing refers to the small holes inside concrete caused by poorly graded concrete mixes or by insufficient consolidation during construction [5]. Concrete with honeycombing will often not have enough strength. The presence of the honeycombing increases the permeability of concrete and makes it susceptible to freeze-thaw damage and other environmental attacks. The reinforcement in concrete with honeycombing are also more exposed to corrosive agents from the outside thereby leading to greater

corrosion. All these effects will greatly reduce the durability of concrete structures with honeycombing. Commonly used NDE techniques for detecting honeycombing includes the impact echo [8] and ground penetrating radar (GPR) [9].

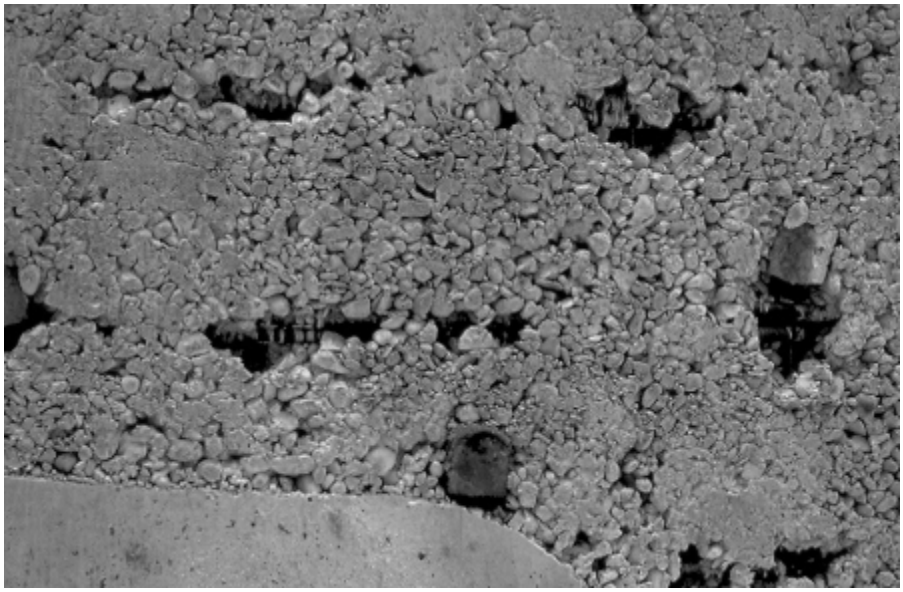


Figure 2.2 Honeycombing in Concrete
(http://www.concrete.org/Troubleshooting/afmviewfaq.asp?faqid=63)

2.1.3 Delamination

Delamination is a layered separation of concrete from the main body. The separation usually occurs at or just above the level of reinforcement as shown in Figure 2.3. This type of damage is usually caused by corrosion of the steel reinforcement [10], a high amount of moisture, and the presence of cracks in the concrete. The progress of the delamination leads to open spalling of the concrete and eventually affects the functional performance of the structure. Concrete delamination impairs not only the appearance of the structure but also its serviceability, and incurs costly repairs if it is not detected in the early stages so that repairs can be made before large-scale deterioration occurs.

Techniques used for detecting delamination include sounding method [11], impact echo [12], and GPR [13].

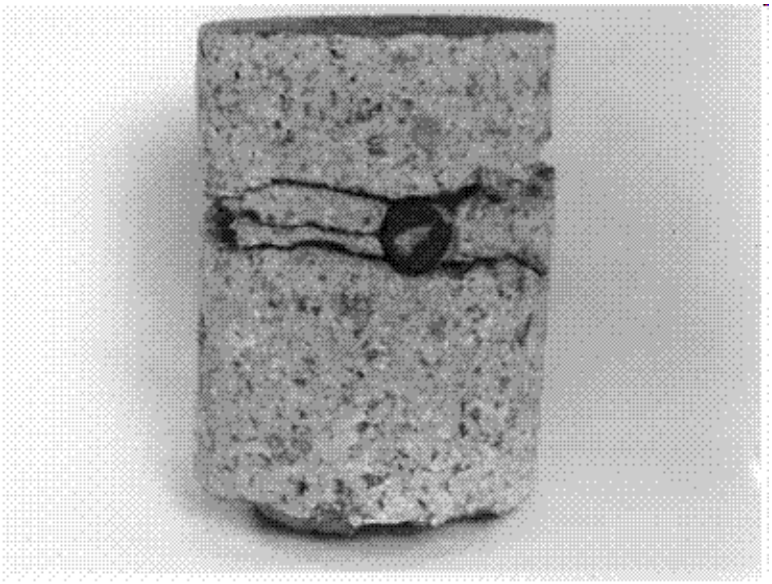


Figure 2.3 Delamination in Concrete
(http://www.fhwa.dot.gov/pavement/pccp/pubs/04150/chapt3.cfm)

2.2 Non-Destructive Evaluation (NDE) Methods for Concrete

The last section described three major types of damage in concrete and their effects on the safety and serviceability of structures. Researchers have developed many different non-destructive methods to detecting concrete damage. This section summarizes the more commonly used NDE methods for concrete damage detection.

2.2.1 Impact Echo and Impulse Response

When the concrete is excited by an impact, a stress wave is generated and propagates inside the body of the concrete. The presence of damages or defects changes the propagation path of the stress wave reflections. The damage can then be identified by

measuring and analyzing the stress waves. There are two dominant NDE methods in this category: impact echo and impulse response.

The impact echo method was first developed at the National Institute of Standards and Technology (NIST) in the 1980s and then further refined by Mary Sansalone at Cornell University [14]. In this method, the stress wave is generated by a short duration impact on the surface and is reflected by internal interfaces or external boundaries. The surface displacement is measured by a transducer and analyzed in the frequency domain. The principle of the method is shown in Figure 2.4. The distance between the receiving surface and the reflecting surface can be calculated as:

$$D = \beta \frac{C_p}{2f} \tag{2.1}$$

where β is a factor related to the shape of the cross-section [14], C_p is the velocity of the P wave in the concrete and f is the peak frequency obtained through frequency domain analysis (for example, FFT) of the signal.

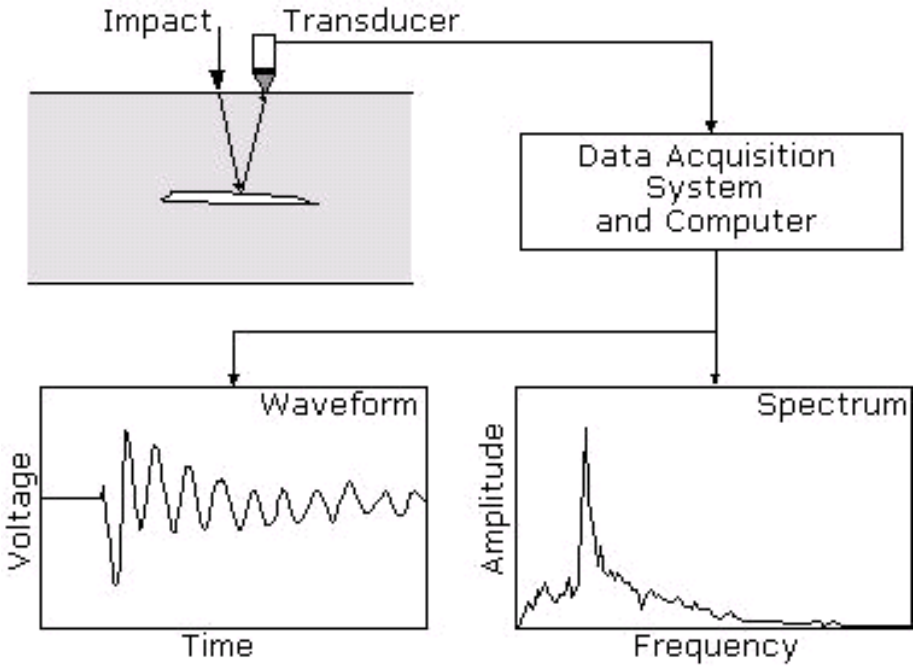


Figure 2.4 Principle of the Impact Echo Method

(www.impact-echo.com/Impact-Echo/impact.htm)

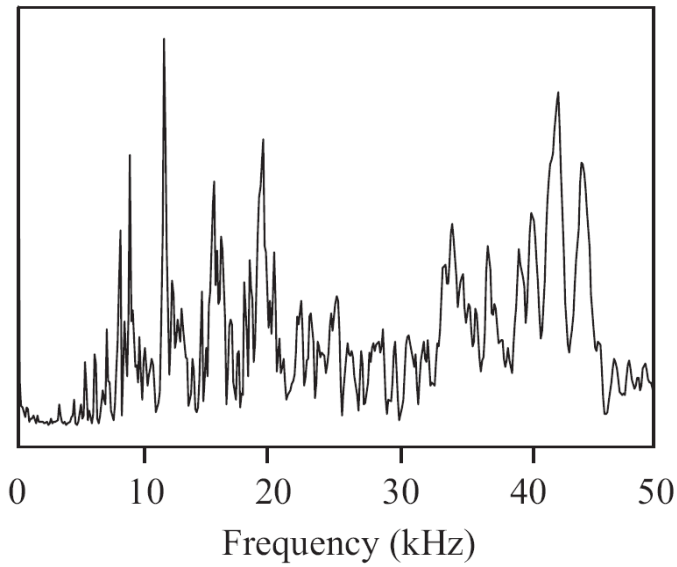


Figure 2.5 Example Spectrum from Impact Echo Tests [15]

The impact echo method can not only detect the presence of the defects, but it also can find the location (depth) of the defects. This method can determine the thickness of the slab and the depth of the defect and needs only one transducer to carry out the test. However, the signal obtained from the impact-echo test in real situations can be difficult to interpret. When the surface of the defect is irregular, the reflection and scattering of the stress wave become very complex. Multiple peaks appear in the frequency domain and it is difficult to identify the peak associated with the defect as shown in Figure 2.5. Even though, some algorithms such as SIBIE [15] have been proposed to identify the peak corresponding to the defects, they requires prior information about the properties and the size of the testing samples which is usually not available for field tests. Also, the method is not sensitive to very shallow defects [16]. There are two reasons for this. First, the frequency of the peak for shallow defects can be very high (the frequency for a 1 inch delamination will be as high as 80 kHz according to Equation (2.1)) and the peak can be difficult to detect. Second, the frequency corresponding to the bending mode of shallow defects will produce false peaks in the frequency domain. The impact echo is not sensitive to those defects that are parallel to the direction of stress propagation [17]. Finally, the sensors have to be coupled with the concrete surface to obtain good measurement and the coupling process is time and labor consuming when inspecting a large area such as a bridge deck.

To increase the efficiency of the traditional impact echo method, Zhu [18] proposed a non-contact impact echo method using air-coupled sensors. Instead of using contact sensors such as accelerometers, this method uses air-coupled sensor microphones to measure the response. This method was reported to be successful in detecting the

presence and locations of delamination and voids in concrete structures. However, the method requires the microphone to be highly directional to record sound in very limited ranges. Also the microphone has to be very close to the surface to be able to pick up the surface response. The analysis of the signal also can be difficult due to the air-coupling effects.

The impulse response (IR) [19] is another NDE method that uses the stress wave generated by an impact on the surface of the concrete. In IR, an impact hammer is used to generate the stress wave with the impacting force measured by a built-in load cell. The response, usually the velocity of the concrete to the impact, is also measured. The transfer function between the impact force and the response can then be computed, from which certain parameters such as dynamic stiffness, mobility and damping can be measured. The integrity of the concrete can then be estimated from the calculated parameters. This method has the same disadvantages as the impact echo as full coupling of the sensors to the ground is needed.

2.2.2 Ultrasonic Methods

Ultrasonic methods also use wave propagation inside concrete. The difference between the impact methods and ultrasonic methods is that the latter uses high frequency (usually greater than 20 kHz) sonic wave as the excitation method, while the impact employs a stress wave resulting from mechanical impacts. One of the commonly used ultrasonic methods is the ultrasonic pulse velocity (UPV). In this method, two transducers are needed: one is used to send and one to receive the ultrasonic wave. By measuring the

arrival time of the signals, the propagation speed of the ultrasonic wave in concrete can be calculated. The test equipment used is shown in Figure 2.6.

The speed of the P-wave in a solid is:

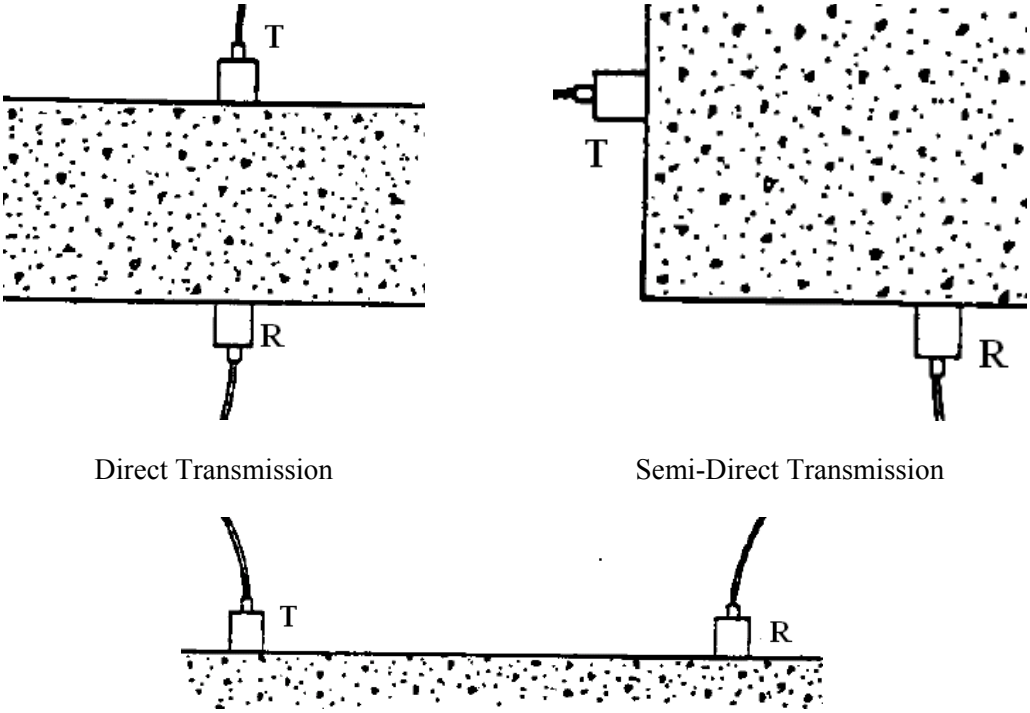
$$C_p = \sqrt{\frac{\lambda + 2\mu}{\rho}} = \sqrt{\frac{E(1-\nu)}{\rho(1-2\nu)(1+\nu)}} \tag{2.2}$$

where λ and μ are Lame's constants, E and ρ are Young's Modulus and density of the solid, and ν is Poisson's ratio.

As the equation shows, the speed is determined by the density and Young's Modulus of the concrete. The defects in a concrete such as crack or delamination are usually of different densities from that of concrete and will lead to a change in the measured pulse velocity. For example, the diffraction of a wave pulse around an air void will cause an increase in the time of propagation and the measured velocity will decrease. By determining the P-wave speed, the uniformity of concrete can be determined. If multiple sensors are used, the 3D image of the internal defect may be obtained through tomography and the synthetic aperture focusing technique (SAFT), as shown in Figure 2.7 [20].

However, there are several problems with this method. First, the transducers have to be coupled to the concrete surface usually by a couplant to ensure that there is no air gap between the surface and the transducer. This will be time consuming if inspecting a large area such as a bridge deck. Second, the accuracy of the method can be affected by other factors such as the temperature and moisture content of the concrete. Third, it might be

difficult to use this method on asphalt coated concrete surfaces due to the difference in mechanical properties and the rough texture of asphalt layers [21].



Direct Transmission

Semi-Direct Transmission

Indirect Transmission

Figure 2.6 Ultrasonic Pulse Velocity Method

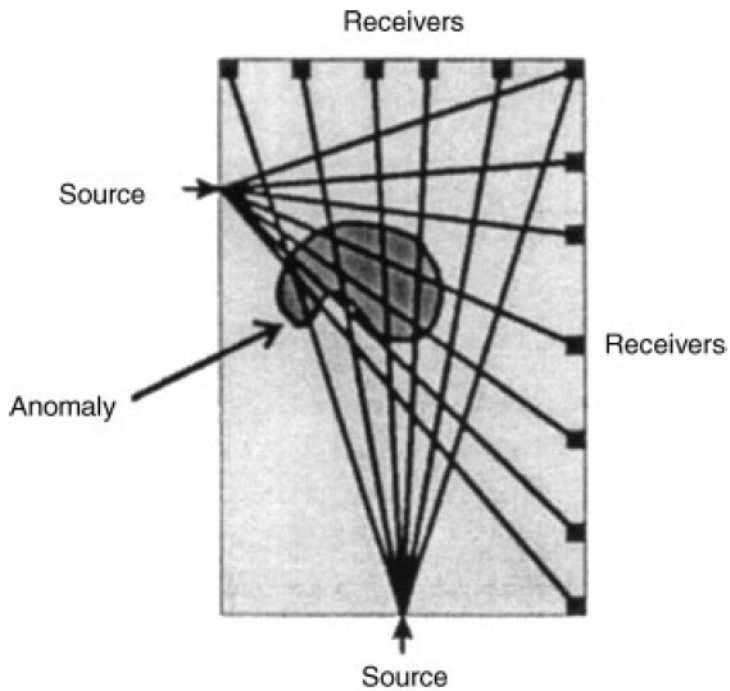


Figure 2.7 Ultrasonic Tomography [20]

2.2.3Ground Penetrating Radar

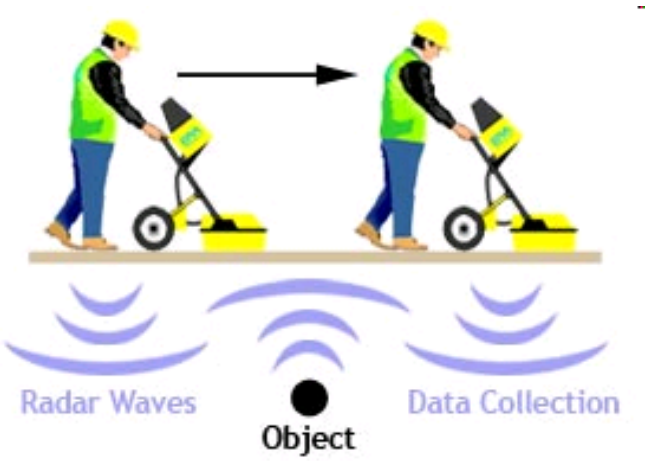
Ground penetrating radar (GPR) is a non-contact method. It uses the interaction between the electro-magnetic (EM) wave and boundaries of materials with different electronic properties. The EM wave will be reflected and backscattered if there's a boundary. The reflected wave is captured by the antenna, as shown in Figure 2.8. The amplitude of the reflected wave is dependent on the relative dielectric constant between the two materials and can be calculated as shown in equation 2.3 [22]:

$$\rho = \frac{\sqrt{\varepsilon_{r1}} - \sqrt{\varepsilon_{r2}}}{\sqrt{\varepsilon_{r1}} + \sqrt{\varepsilon_{r2}}} \tag{2.3}$$

where ρ is the reflection coefficient, and ε_{r1} and ε_{r2} are the dielectric constants of the materials at the interface. If the difference across the interface is large, the EM wave will be reflected back; if the difference is small, the majority of the EM wave will pass

through. By measuring the energy of the reflected wave, the type and location of defects inside the concrete are detected.

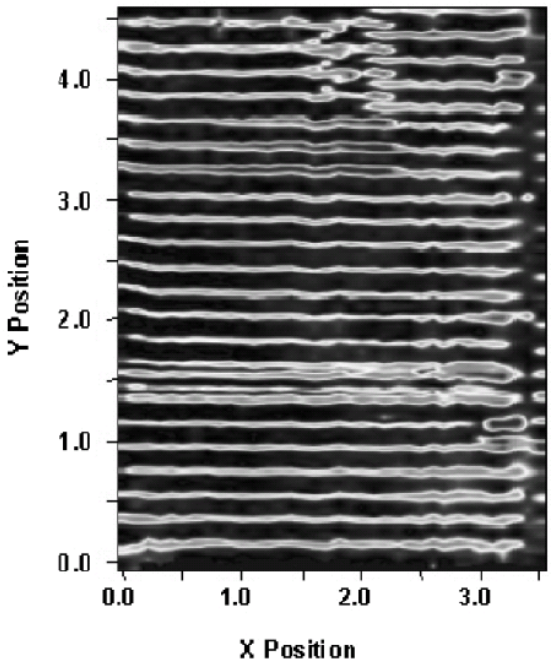
The GPR approach is non-destructive and non-invasive and results can be displayed in real time as a radiogram. It can locate steel reinforcement and damaged or deteriorated areas inside the concrete. Also, the equipment can be carried on a car/truck and can rapidly scan large areas. However, the method has its own disadvantages. The results are presented in the form of a B-scan or C-scan (shown in Figure 2.9 [9]) and require professional knowledge for interpretation. The performance may be affected by many variables including material type, moisture and so on. Also, there is a trade-off between penetration and resolution due to limits on the antenna selection. The radar cannot detect objects smaller than the wavelength. In order to increase the resolution, the frequency of the radar wave needs to be high. However, the increase in frequency leads to a reduction in the penetration capacity. Another problem associated with GPR is the inability to detect voids and cracks filled with air because of the small contrast between the dielectric constants of air and concrete.



(http://www.worksmartinc.net)

Figure 2.8Ground Penetrating Radar

B-scan



C-scan

Figure 2.9 Results from Ground Penetrating Radar [9]

2.2.4 Infrared Thermography

Infrared thermography detects sub-surface defects from the distribution of the temperature field. The heat transfer process is dependent on material properties like thermal conductivity, heat capacity and density. The heat transfer is even only when the material is homogeneous. If there are anomalies inside the materials, the heat flow and the temperature distribution will change in these areas. By observing the variations of surface temperature, defects near the surface can be detected if they generate temperature difference. The presence of air-filled defects such as delaminations or cracks can change the path of the heat flow and can therefore be detected using this method [23]. The surface temperature distribution is obtained indirectly by measuring the infrared radiation with an infrared camera. The relationship between the infrared radiation R and the surface temperature T is:

$$R = e\sigma T^4 \tag{2.4}$$

where e is the emissivity of the surface and σ is the Boltzmann constant.

Infrared technology is a non-contact and non-invasive way to detect defects. This method also has the advantages of being portable and fast. The results are usually displayed in the form of a thermograph (shown in Figure 2.10), which can be readily understood. This method has been used to detect defects in civil structures [24]. Since this method uses infrared radiation to measure the temperature, the results are affected by the variance in the emissivity of the surface, for example, surface moisture, patched areas and varying finishes [25]. Heat flow is needed for this technique to work and an object in thermal equilibrium will not provide useful information. Therefore, heat sources are

needed. Depending on the type of heat sources, this method can be divided into two categories: passive investigation and active investigation. In passive investigation, natural heat sources such as the sun are used. In active investigation, Active heat sources such as infrared radiators are used. For large structures, it takes a very long time and a high energy cost to create sufficient heat flow. In practice, the test is conducted using passive heat sources. The tests are performed in the morning when the sun starts to heat the structure or after sunset when the heat in the structure starts to radiate into the environment. This makes the performance of the method weather-dependent. Active investigation is usually applicable only to small specimens or for localized testing. Also, the infrared camera can only measure the temperature close to the surface, the deeper defects will not be reflected in the surface temperature. Therefore this method is insensitive to deeper damage. Lastly, the high cost of the infrared camera is another limiting factor for this technique.

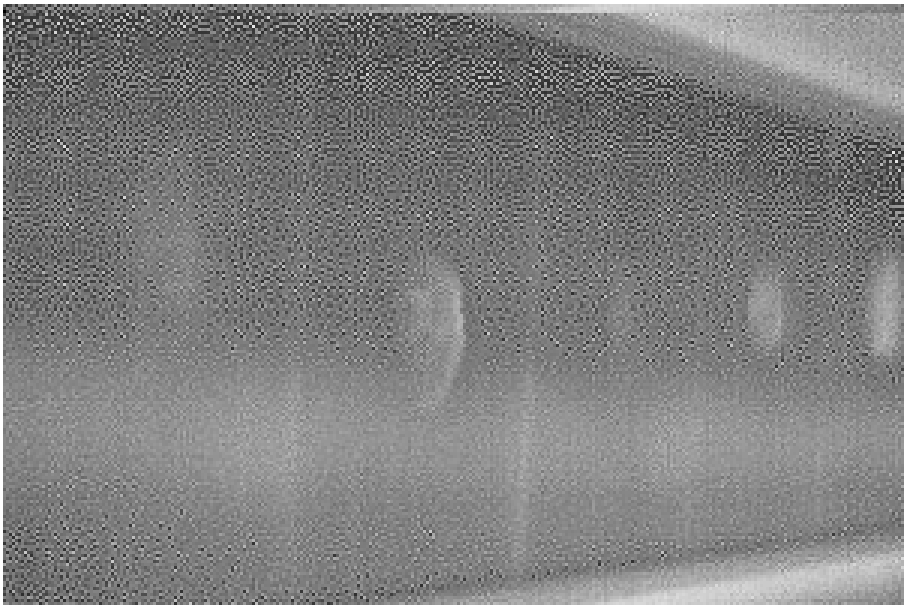


Figure 2.10 Example Image from an Infrared Camera

2.2.5 X-ray Imaging

As mentioned in Section 2.2.4, ground penetrating radar has the problem of penetration. X-rays however, can easily penetrate into concrete. When the X-ray beam passes through a material, the energy is absorbed or scattered. The amount of energy absorbed or scattered is a function of the mass density of the components, and materials with higher mass density absorb or scatter a greater amount of energy. A collector placed behind the specimen receives the scattered signals as shown in Figure 2.11 [21]. In the images obtained, the high density materials are represented by light areas and low density materials are shown as dark areas. Common types of concrete defects are air-filled voids or cracks that have a clear contrast in mass densities. Therefore, these defects can be easily detected through the X-ray imaging. X-ray imaging can provide clear pictures of the internal structure of the specimens and the presence and locations of the defects can be identified with high accuracy if the energy of the X-ray is properly adjusted. There are several limitations associated with this method. First, the cost of the equipment is usually very high. Second, there are safety concerns on the use of radiation. Third, two sides of the specimen need to be accessible. Also, X-ray images are not sensitive to defects that are parallel to the radiation direction.

Traditional X-ray images can only provide an average density contrast in 2-dimensions and information in the third dimension about defects is not available. There are several reports [26-28] on using X-ray computerized tomography (X-ray CT) to obtain images in 3 dimensions (3D) by taking pictures of slices of the specimen in 2D and reconstructing the internal structures into 3D. This process is expensive and requires

long processing time. Due to the above limitations, the use of X-ray imaging has not been widespread.

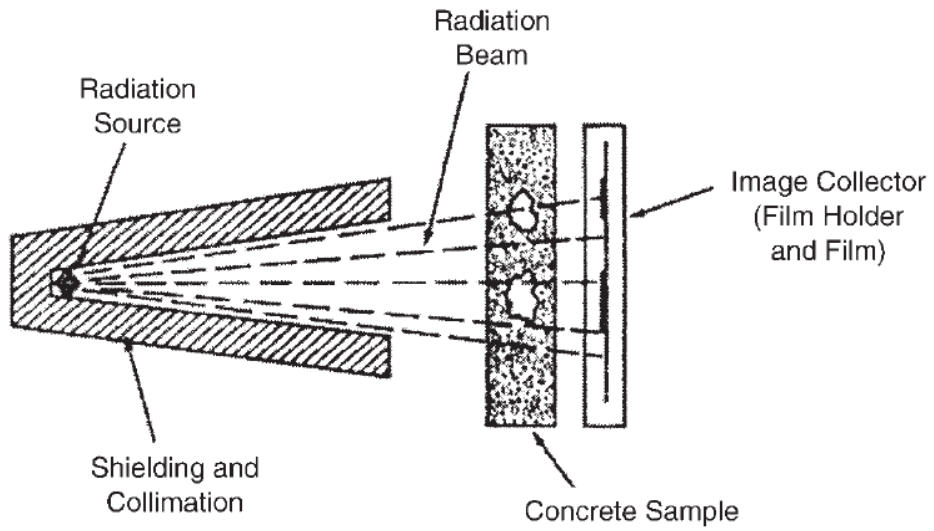


Figure 2.11 X-ray Imaging of Concrete Samples [21]

2.2.6 Sounding Methods

Sounding techniques for non-destructive evaluation (NDE) of concrete decks have been widely used because they are fast, simple and inexpensive. Traditional sounding methods for delamination detection involve: (1) bar/hammer tapping of the deck and listening to the acoustic response, and (2) dragging a chain over the deck as shown in Figure 2.12 and listening to the change in the sound. In both methods, good concrete with no delamination produces a clear ringing sound, while delaminated concrete is characterized by a dull, hollow sound. Standard test procedures are defined in ASTM C4580-2003 [11].

Figure 2.12 Chain Drag Test

Sounding methods have their own problems. The first problem arises due to traffic noise from adjacent lanes. Usually only one lane is closed for inspection and noise is generated by traffic in adjacent lanes as well as from wind and other sources. Figure 2.13 shows the spectrogram of the recorded signals under both quiet and noisy environments. The complex environment makes the sound field difficult to analyze. Furthermore, the traffic noise is non-stationary and broadband. This makes the problem complicated and a simple band-pass filter cannot efficiently eliminate the noise. The second problem results from the fact that the detection is dependent on the subjective interpretation of the inspector, which makes it difficult to document the inspection results. Therefore, improvement of traditional sounding methods may enhance detection.

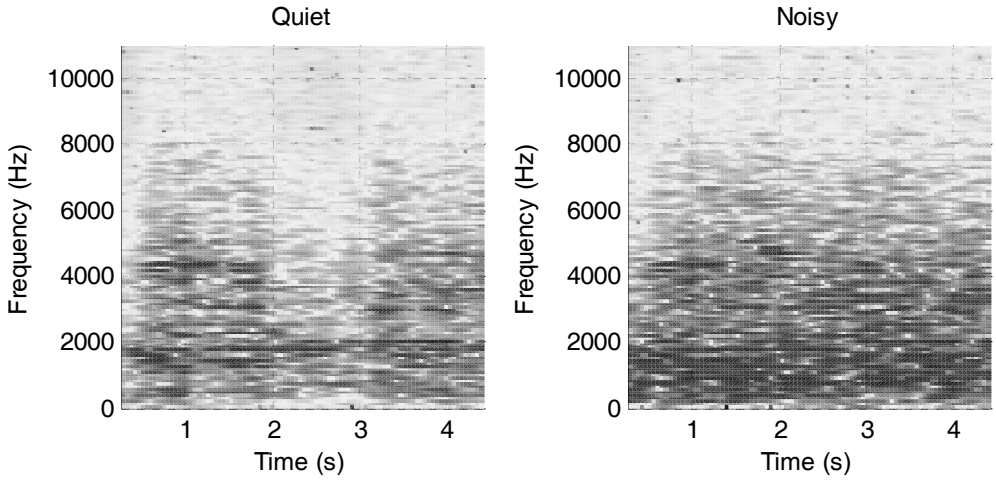


Figure 2.13 Spectrogram of Chain Drag Tests

Although, several attempts have been made to improve sounding methods, research on this topic is still quite limited. In 1977, researchers at the Michigan Department of Transportation (MDOT) designed a cart-like device for delamination detection [29]. The impulse was created by the chattering of two rigid wheels with the concrete and the vibration of the concrete was captured by a transducer coupled to the ground through soft tires and liquid in the wheels. The recorded signals were first truncated such that only the impact 5 ms after tapping was analyzed and then filtered by a fixed band pass filter with cut-off frequencies at 300 and 1200 Hz. The processed signals were recorded on charts. The audible signal was detected through earphones. This method was automatic, but the signal processing algorithm was primitive. Henderson et al. [30] used sound signals created by dragging a chain. The traffic noise was isolated by sound proofing around the chains and recording device. A computer algorithm using linear prediction coefficients (LPC) was used to analyze the recorded signals and perform the detection. Although this technique showed promise, the method had three major drawbacks. First, the traffic noise was reduced only by physical isolation and use of a directional microphone, which can be

ineffective at high noise levels and for complex sound fields encountered on highway bridges. Second, traffic noise is usually non-stationary and simple filtering is often inadequate. Third, the computation takes a considerable amount of time.

2.3 Summary

This chapter described the commonly found defects such as honeycombing and delamination in concrete and several non-destructive techniques for damage detection. Table 2.1 summarizes the advantages and limitations of different methods.

The selection of the NDE method should be based on the target defect. In the case of delamination detection for bridge decks, the sounding method is a good choice because of its advantages of being inexpensive, simple and fast. However it has its own problems: noise contaminated signals and subjective interpretation of results. Research on the improvement of sounding methods has been limited. The research presented here will improve traditional sounding methods by focusing on the problem of noise cancellation and automatic detection.

Table 2.1Summary of Different Non-destructive Evaluation Methods

Methods	Applications and Advantages	Limitations
Impact Echo	♦ Can detect cracks, voids and delaminations ♦ The locations of the defects can also be determined	♦ Analyzing the results is difficult ♦ Shallow delaminations and delaminations parallel to stress propagation cannot be detected ♦ Sensors need to be coupled with surface
Ultrasonic Pulse Velocity	♦ Able to detect different types of detects ♦ Strength of concrete can also be determined ♦ Test procedure is easy	♦ Sensors have to be coupled with surface ♦ Accuracy can be affected by other factors ♦ It does not work well on asphalt overlays
Ground Penetrating Radar	♦ Has a wide range of applications ♦ Equipment is portable and mobile ♦ Inspection procedure is fast and result is provided in real time	♦ Interpreting the results requires professional knowledge ♦ Resolution and penetration needs to be balanced ♦ Air-filled defects cannot be detected
Infrared Thermography	♦ Non-contact method and fast to perform ♦ Equipment is mobile and provides results in real time ♦ Result is easy to understand	♦ External heat sources (active or passive) are needed ♦ Deeper defects cannot be detected ♦ Equipment is expensive
X-ray Imaging	♦ Has very good penetration capacity ♦ Contrast between concrete and air-filled defects is clear ♦ Presence and locations of the damage can be obtained	♦ Radiation is a safety concern ♦ Access to both sides of the specimens is needed ♦ Equipment is very expensive
Sounding	♦ The equipment is very cheap ♦ Inspection process is fast and easy ♦ Results are provided in real-time	♦ High traffic noise may affect the accuracy ♦ Detection process is subjective ♦ Extensive training of the operator is needed

(This page intentionally left blank.)

CHAPTER 3

NOISE CANCELLING ALGORITHMS

As described earlier, sounding tests are often conducted in a noisy environment. Traffic noise combined with other ambient noise such as wind often contaminates the sounding signals, which affects the accuracy of the delamination detection. Eliminating the unwanted noise can enhance the signal and improve the detection performance. Noise cancelling is a basic yet difficult problem. Extensive research has been performed on this topic and various types of algorithm have been proposed and implemented. This chapter describes the technical details and performance of several commonly used algorithms and the selection of an effective algorithm for traffic noise cancellation.

3.1 Evaluation Criteria for Noise Cancelling Algorithms

Chapter 2 summarized several commonly used algorithms. Clearly, different noise cancelling algorithms are designed for different purposes. The performance of algorithms can only be evaluated when some information about the system such as the original signal, the estimated signal, mixing type (instantaneous mixture or convolutive mixture) and filter length are available. To accomplish this, the recordings were obtained through computer simulation: the impact signals were recorded in a quiet lab environment and the noise signal was obtained by recording the traffic noise on a highway bridge. The impact signal and traffic noise were mixed in different ways on the computer to simulate different environments. The performance of the algorithms was evaluated by a numerical

criteria based on orthogonal projections (Vincent etc. 2006). In this method, the estimated signal is decomposed as:

$$\hat{s} = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{3.1}$$

where s_{target} represents the part of $\hat{s}$ from the wanted source (original signal in this case) and e_{interf} , e_{noise} and e_{artif} are the errors due to interference (unwanted sources), measurement noises and artifacts (other causes), respectively. Detailed computation of those components can be found in the reference (Vincent etc. 2006). Based on Equation (3.1), four performance criteria are defined as follows:

$$SDR = 10\log_{10}\frac{\left\|s_{target}\right\|^2}{\left\|e_{interf} + e_{noise} + e_{artif}\right\|^2} \tag{3.2}$$

$$SIR = 10\log_{10}\frac{\left\|s_{target}\right\|^2}{\left\|e_{interf}\right\|^2} \tag{3.3}$$

$$SNR = 10\log_{10}\frac{\left\|s_{target} + e_{interf}\right\|^2}{\left\|e_{noise}\right\|^2} \tag{3.4}$$

$$SAR = 10\log_{10}\frac{\left\|s_{target} + e_{interf} + e_{noise}\right\|^2}{\left\|e_{artif}\right\|^2} \tag{3.5}$$

where *SDR* = the Source to Distortion Ratio;
SIR = the Source to Interference Ratio;
SNR = the Source to Noise Ratio;
SAR = the Source to Artifacts Ratio.

Since the recordings here are simulated noise, there is no contribution from the unwanted sources and measurement noises and some artifacts may be introduced in the

estimated signal due to the limit of the algorithm.. Also, it can be shown from Equations (3.2) to (3.5) that SDR and SAR are equivalent in the absence of interference and measurement noise. In this work, only SDR is used as the performance criteria and the performance of all candidate algorithms below are evaluated using the criteria. The SDR are computed by a MATLAB function coded by Vincent [31].

3.2 Spectral Subtraction

3.2.1 Theoretical Background

Spectral subtraction as a noise cancelling algorithm was briefly described in Chapter 2 [32]. A more detailed description of this algorithm is provided here. The algorithm assumes that the noisy recording is obtained by adding a windowed noise to a windowed signal, which can be expressed in the frequency domain as:

$$X\left(e^{jw}\right) = S\left(e^{jw}\right) + N\left(e^{jw}\right) \tag{3.6}$$

where $X\left(e^{jw}\right)$, $S\left(e^{jw}\right)$ and $N\left(e^{jw}\right)$ represent the Fourier transform of the recording, the signal and noise, respectively. For convenience, “recording” indicates the signal recorded by the microphone the includes unwanted noise; “signal” refers to the acoustic signal created by impacting the concrete using a hammer or other methods and “noise” refers to the ambient sound, such as traffic noise. These definitions are used for the remainder of the chapter.

Assuming that the noise is stationary over the duration of the recording, the spectrum of the noise can be estimated from the recording during quiet period before or after the

signal. The length of the noise recording can be increased by extracting and joining segments from adjacent windows. The spectrum of the signal can be estimated by:

$$\begin{aligned}\hat{S}\left(e^{jw}\right) &= \left[\left|S\left(e^{jw}\right)\right| - \left|\mu\left(e^{jw}\right)\right|\right] e^{j\theta_x} \\ &= H\left(e^{jw}\right) X\left(e^{jw}\right)\end{aligned} \tag{3.7}$$

where $\mu\left(e^{jw}\right)$ is the average value of the noise spectrum obtained from the recording segment with no signal and $H\left(e^{jw}\right)$ is calculated from Equation (3.8).

To reduce the variance in the spectrum of the noise, a longer window in time domain is preferred. However, the actual noise is usually non-stationary indicating that the spectral properties of the noise are changing. The spectrum estimated from a long window will be an averaged over the entire window, which may not be a good estimate of noise spectrum during the period with signal. This will introduce error in the final results. Therefore, a balance between these two has to be considered.

$$H\left(e^{jw}\right) = 1 - \frac{\mu\left(e^{jw}\right)}{\left|X\left(e^{jw}\right)\right|} \tag{3.8}$$

In some instances $H\left(e^{jw}\right)$ may be negative, meaning that the sum of signal plus noise is less than the noise, which can not be the case. Half-wave rectification is used to solve this problem, in which the negative value is replaced by zero. This process can be expressed by

$$H_R\left(e^{jw}\right) = \frac{H\left(e^{jw}\right) + \left|H\left(e^{jw}\right)\right|}{2} \tag{3.9}$$

3.2.2 Performance Evaluation

The performance of the spectral subtraction algorithm was tested using simulated data. The traffic noise recorded from the highway was directly added to the impact signal and the noise in the adjacent window was used as the reference (signal recorded in quiet period) to estimate the signal in the previous window. The algorithm was implemented in MATLAB. The results are shown in Figure 3.1. As can be seen from this figure, the original signal can not be fully recovered. This may be due to two reasons. First, the traffic noise is not stationary and the properties of the noise in the window before the occurrence of the impact are different from the window in which the impact occurs. Second, there might be overlaps between the spectrum of the noise and that of the impact, and when the noise components are subtracted; some components of the impact signal also are likely to be cancelled. The SDR of this algorithm is computed to be -6.514 dB, indicating that the performance of the spectral subtraction algorithm is poor.

3.3 Adaptive Filters

3.3.1 Theoretical Background

As the spectral subtraction algorithm demonstrated, the impact signal cannot be successfully reconstructed from noisy recordings. The main reason is the assumption that the noise signal is short-term stationary for traffic noise at sites. Adaptive filter algorithms can be more effective in solving this problem. One of the commonly used adaptive filters is the least-mean-square (LMS) algorithm [33]. Figure 3.2 shows an adaptive noise cancelling system in which there are two microphones in the system. The primary microphone records the mixture of the source signal (s) and the noise (n_0) and

the reference microphone is used to record a filtered version of the noise (n_1). The adaptive filter consists of a tapped delay line and the weights in the adaptive filter automatically seek an optimal impulse response by adjusting themselves such that the error between the outputs of the filter (y) is the best estimate of the noise in the primary microphone in the sense of least mean square error. The estimated source (z) can be obtained by subtracting the filter output from the primary signal.

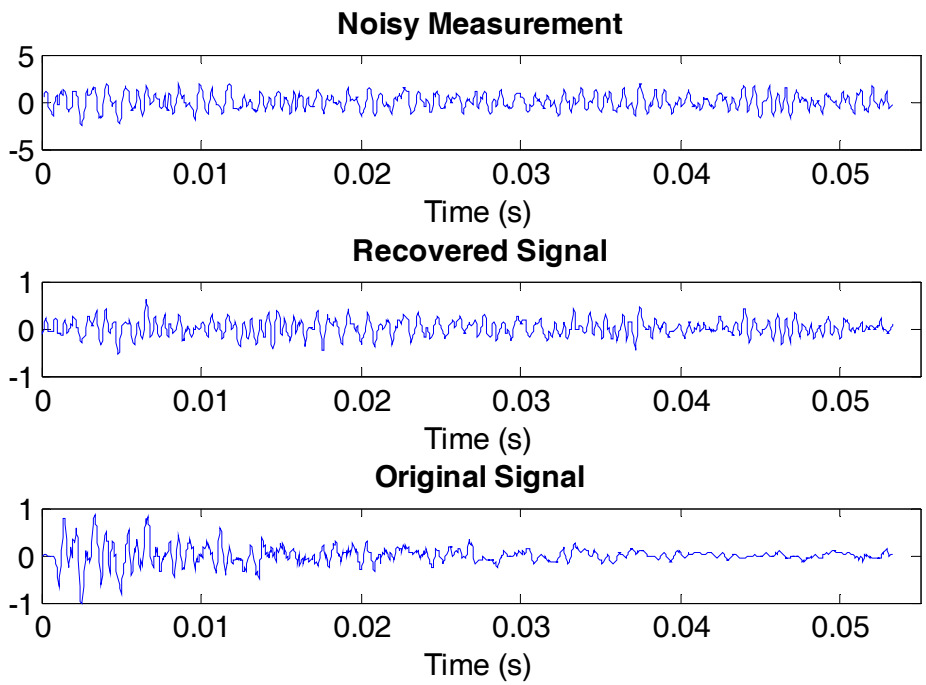


Figure 3.1 Performance of Spectral Subtraction

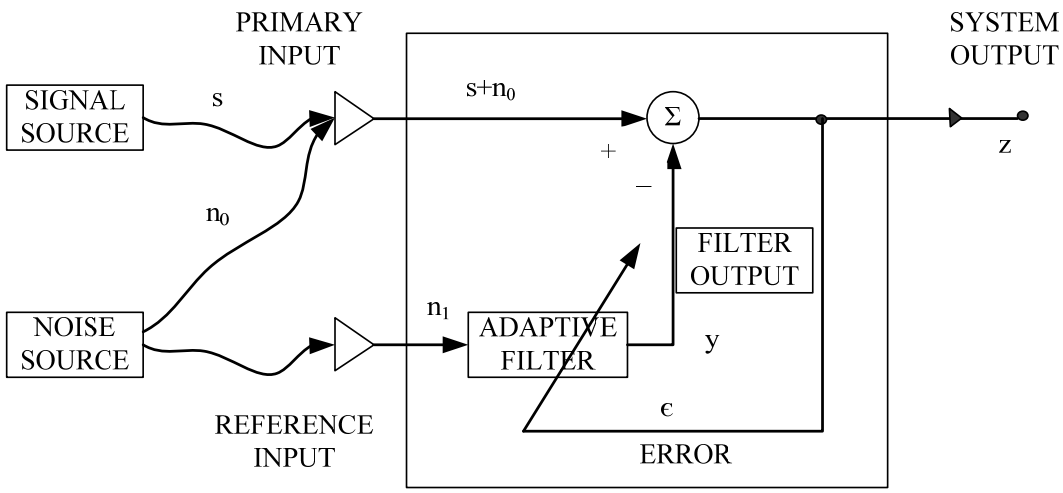


Figure 3.2 Noise Cancelling using Adaptive Filter [33]

The mean square error (MSE) between the estimate source and the original source can be expressed as

$$\begin{aligned} MSE &= E\left[\left(z-s\right)^2\right] \\ &= E\left[\left(s+n_0-y-s\right)^2\right] \\ &= E\left[\left(n_0-y\right)^2\right] \end{aligned} \tag{3.10}$$

where, $E\left[\cdot\right]$ represents the operation of expectation. From Equation (3.10), it can be seen that minimizing the error between the estimated source and the original source is equivalent to minimizing the mean square error between the estimated noise and the noise in the primary microphone. Therefore, the LMS method can be used in the noise cancelling problem.

The output of the adaptive filter can be calculated from

$$y = \sum_{l=1}^{L} w_l n_{1l} = W^T n_1 = n_1^{\ T} W \tag{3.11}$$

The MSE between the filter output and the noise signal can then be described as:

$$\begin{aligned} MSE &= E\left[\left(n_0 - y\right)^2\right] \\ &= E\left[\left(n_0 - n_1^T W\right)^2\right] \\ &= E\left[\left(n_0\right)^2\right] - 2E\left[n_0 n_1^T\right]\cdot W + W^T \cdot E\left[n_1 n_1^T\right]\cdot W \end{aligned} \tag{3.12}$$

The steepest descent algorithm updates the weight vector proportional to the gradient vector:

$$W_{j+1} = W_j - \mu \frac{\partial\left(MSE\right)}{\partial\left(W_j\right)} \tag{3.13}$$

where μ is a factor that controls the rate of adaptation.

From Equation (3.12), the gradient of the MSE can be computed as:

$$\frac{\partial\left(MSE\right)}{\partial\left(W\right)} = 2E\left[n_0 n_1^T\right] + 2W^T \cdot E\left[n_1 n_1^T\right] \tag{3.14}$$

Substituting Equation (3.14) into (3.13), the update law of weight vector is

$$\begin{aligned} W_{j+1} &= W_j - 2\mu n_{0j} n_{1j}^T + 2W^T \cdot n_{1j} n_{1j}^T \\ &= W_j - 2\mu\left(n_{0j} - W^T n_{1j}\right) n_{1j}^T \\ &= W_j - 2\mu\left(n_{0j} - y_j\right) n_{1j}^T \\ &= W_j - 2\mu e_j n_{1j}^T \end{aligned} \tag{3.15}$$

where, e_j is the error between the filter output and the desired signal. For the adaptive noise canceling algorithm, the desired signal is the source ($s+n_0$). Therefore, the error in Equation (3.15) is in fact the estimated signal (z), as shown below

$$e_j = (s + n_0) - y = z \tag{3.16}$$

Therefore, the update law for the weight vector becomes

$$W_{j+1} = W_j - 2\mu z_j n_{1j}^{\ T} \tag{3.17}$$

In the LMS method, the function that the algorithm tries to minimize is calculated from the MSE in the current step. Therefore, the performance and convergence rate of the algorithm may be affected by the transient response. To improve the performance, a modified LMS algorithm called the recursive least square (RLS) algorithm is used in this section. Instead of minimizing the MSE from the current step, the RLS algorithm minimizes the total MSE over N steps, as shown below:

$$C = \sum_{i=1}^{N} \beta(n,i) \left| e(i) \right|^2 \tag{3.18}$$

where $\beta(n,i)$ is the weighting or "forgetting" factor. The derivation of the update law is similar to that in the LMS method and is described in detail. The convergence rate of the RLS algorithm is much faster than that of the LMS method but the computation time is longer.

3.3.2 Performance Evaluation

To evaluate the performance of the adaptive filter, several difference cases were simulated. The noisy recordings of different conditions were obtained by mixing the scaled impact signal with the noise signal as shown below:

$$m(t) = \alpha s(t) + n(t) \tag{3.19}$$

where m = the simulated measurement;

α = scaling coefficient that controls the SNR;

s = the clean signal;

n = the traffic noise signal.

Case 1: The recording of the primary microphone was simulated by directly adding the traffic noise to the impact signal, the scaling factor is 1. The recording of the secondary microphone was simulated as a filtered version of the same traffic noise used for the primary microphone. The length of the adaptive filter was the same as that of the filter used to create the reference signal. In this case, a filter length of 5 was used.

Case 2: The primary and reference signals were the same as in Case 1 but the length of the adaptive filter was 2.

Case 3: The primary and reference signals were the same as in Case 1 but the length of the adaptive filter was 4.

Case 4: The primary signal was the same as in Case 1, but the reference signal was also a mixed version of the filtered traffic noise and impact signal. The scaling factor was 0.316. The length of the filter was assumed to be 5 for both the adaptive filter and the actual filter.

Case 5: This case was identical to Case 4 except that the scaling factor in the reference signal was 0.0316.

The RLS algorithm performs very well if the length of the adaptive filter is equal to or greater than the actual filter (such as in Case 1), as shown in Figure 3.3. The source

signal was masked in the noisy recording, but was effectively and rapidly recovered by the RLS algorithm. The SDR of Case 1 was calculated to be 10.51dB.

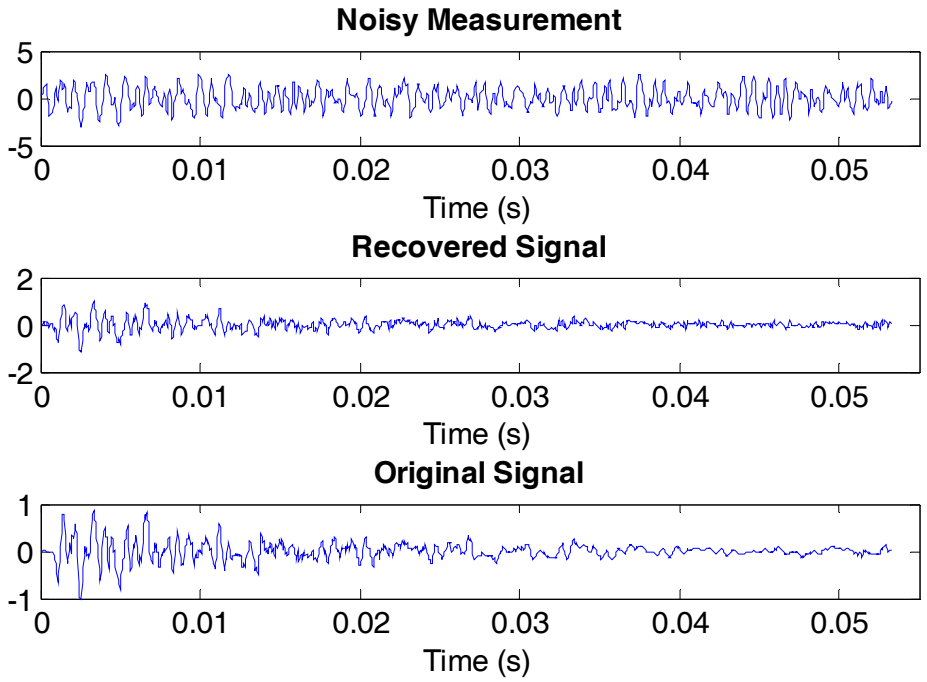


Figure 3.3 Performance of RLS Adaptive Filter (Case 1)

However, if the length of the filter is underestimated, the performance will drop because the effect of the source filter cannot be fully represented by filters with a shorter length. Figure 3.4 shows the results for Case 2. The SDR for this case is -6.212 dB, which is considerably lower than for the ideal case. When the length of the filter was increased to 4 (Case 3), the result obtained is shown in Figure 3.5 and the performance index SDR increases to 1.658 dB.

The comparison above indicates that the SDR is a good performance measure. It is very difficult to judge which case has better performance by visual inspection. However, the SDR is able to characterize the change in performance.

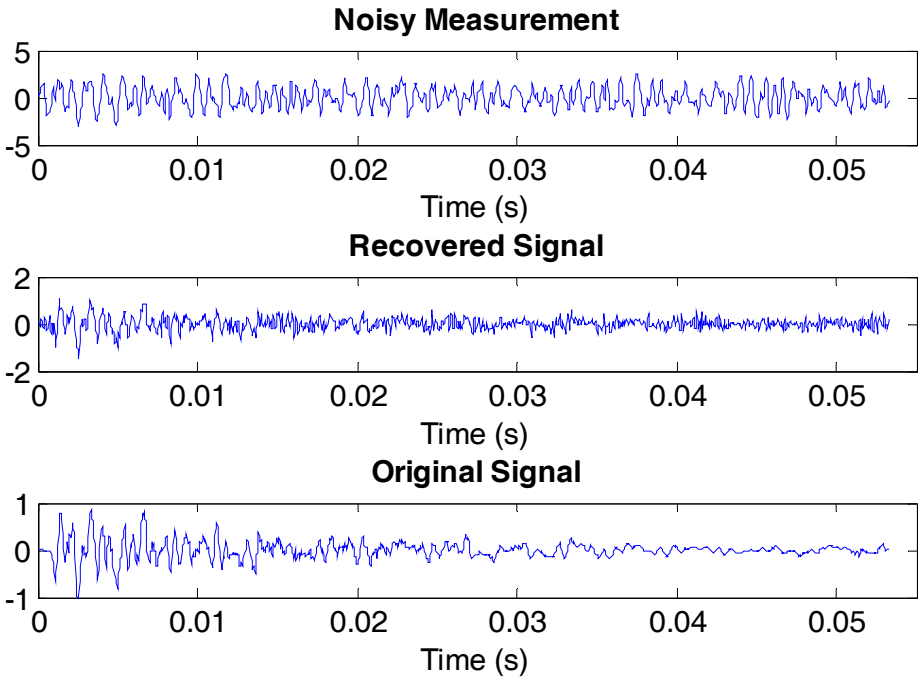


Figure 3.4 Performance of RLS Adaptive Filter (Case 2)

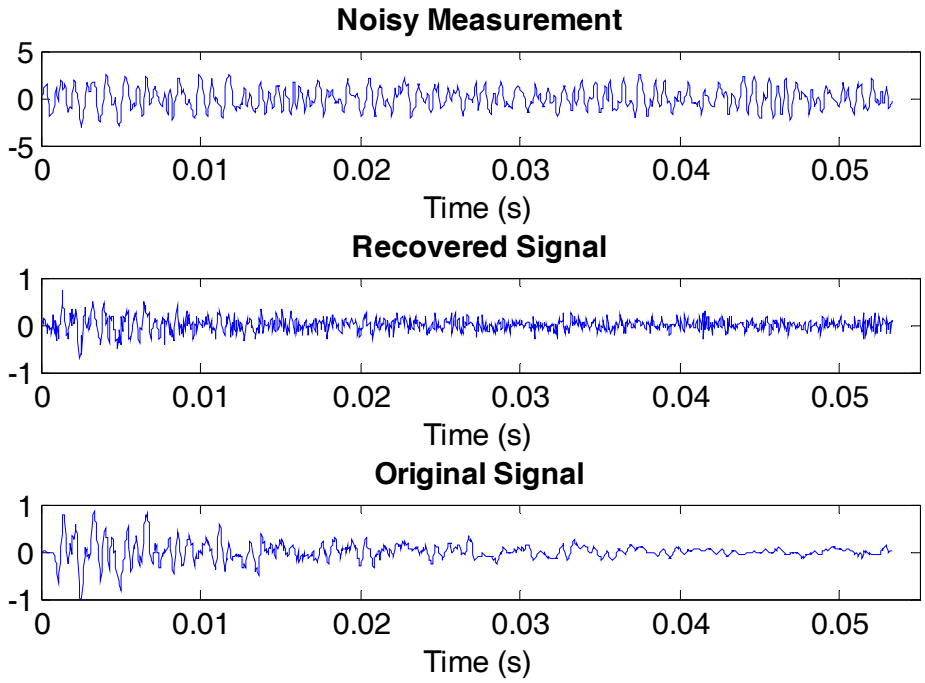


Figure 3.5 Performance of RLS Adaptive Filter (Case 3)

Another problem with this type of noise cancelling algorithm is that when some source components leak into the reference signal, the source in the primary signal will inevitably be cancelled. This leads to distortion and reduces the performance. Figure 3.6 shows the performance of Case 4 where the source and the noise are mixed at the signal to noise ratio of -5 dB in the reference recording. It can be seen that the signal is significantly distorted. The SDR of the recovered signal is -6.212 dB. If the signal to noise ratio in the reference signal becomes -50 dB (Case 5), the performance improves since only a fraction of the source signal is cancelled. This increase in performance is also reflected in the waveform as shown in Figure 3.7. The SDR in this case increases to -2.334 dB, but is still much lower than the SDR for the ideal case.

From the above discussion, it can be concluded that the adaptive filter can efficiently cancel the unwanted noise under ideal conditions. (i.e. when the length of the adaptive filter is equal to or greater than the actual filter and there is no signal component in the reference signal). However, the performance drops quickly if these requirements are not statisfied.

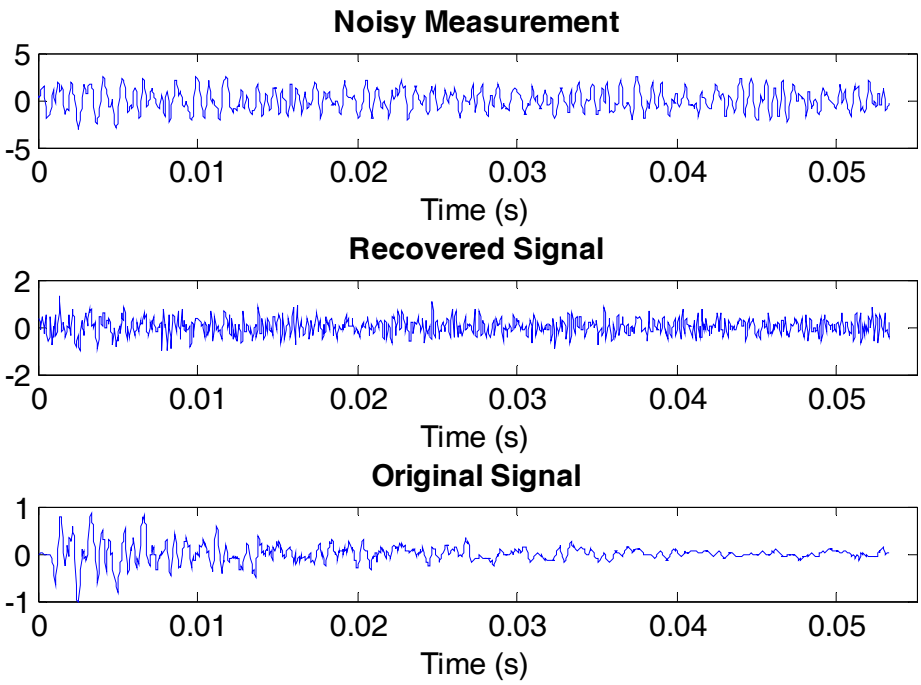


Figure 3.6 Performance of RLS Adaptive Filter (Case 4)

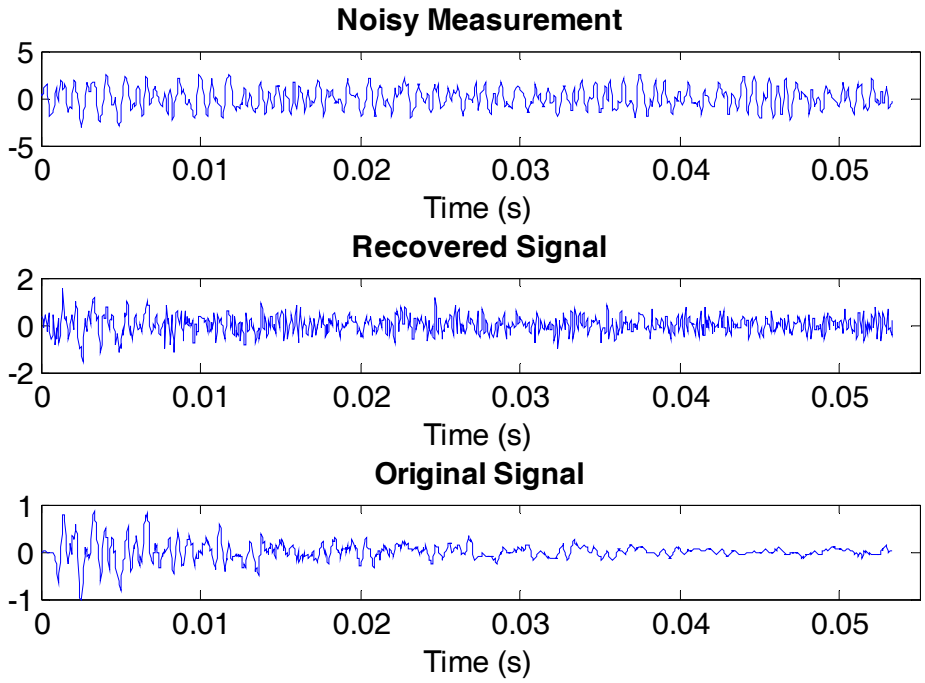


Figure 3.7 Performance of RLS Adaptive Filter (Case 5)

3.4 Independent Component Analysis

3.4.1 Theoretical Background

Case 5 in the previous section shows that the presence of a small portion of the source signal in the reference recording leads to a significant decrease in the SDR of the reference signal. Independent component analysis (ICA) is employed in this section to solve the problem. The concept of ICA was first proposed by Comon in 1994 [34] and illustrated in Figure 3.8. The algorithm assumes that the sources are mutually independent. The de-mixing of the recordings can be performed by maximizing the independence between the outputs of the algorithm. Once the independence is maximized, the outputs will be scaled versions of the original sources. The maximization of

independence is realized by the adaptation of a de-mixing matrix. The detailed derivation of the method is described as follows [35]:

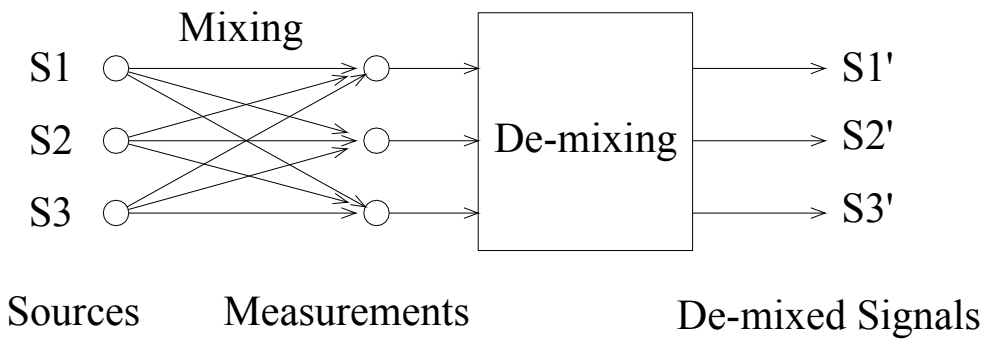


Figure 3.8 Independent Component Analysis

The output of the de-mixing matrix (W) can be expressed as:

$$Y = WX \tag{3.20}$$

where X = matrix of mixed signals.

Y = matrix of de-mixed signals.

W = de-mixing matrix.

The probability density function (PDF) of de-mixed signals is assumed to be $f_Y(y,W)$. The objective is to adjust the de-mixing matrix such that the output signals Y_i and Y_j are independent. If the output signals are independent, the PDF of de-mixed signal can be expressed as

$$\tilde{f}_Y(y,W) = \prod_{i=1}^{m} \tilde{f}_{Yi}(y_i,W) \tag{3.21}$$

where $\tilde{f}_{Yi}(y_i,W)$ is the marginal PDF of the Y_i output signal.

The objective can therefore be achieved by minimizing the "difference" between $f_Y(y,W)$ and $\tilde{f}_Y(y,W)$. In a statistical sense, one common way to measure the difference between two PDFs is the Kullback-Leibler (KL) divergence. The KL divergence between two PDFs f_X and g_X is computed as

$$D_{f_x \| g_x} = \iint f_X(X) \log\left(\frac{f_X(X)}{g_X(X)}\right) dx \tag{3.22}$$

Substituting the expressions for $f_Y(y,W)$ and $\tilde{f}_{Yi}(y_i,W)$ into Equation (3.22) yields

$$\begin{aligned} D_{f_Y \| \tilde{f}_Y} &= \iint f_Y(Y) \log\left(\frac{f_Y(Y)}{\prod_{i=1}^{m} \tilde{f}_{Y_i}(y_i,W)}\right) dy \\ &= \int f_Y(Y) \log f_Y(Y)\, dy - \sum_{i=1}^{m} \int f_Y(Y) \log \tilde{f}_{Y_i}(Y_i)\, dy \\ &= -h(Y) - \sum_{i=1}^{m} \tilde{h}(Y_i) \end{aligned} \tag{3.23}$$

where $h(\bullet)$ is the entropy of a random variable that can be calculated as

$$h(x) = \int_{-\infty}^{+\infty} f_X(X) \log f_X(X)\, dx \tag{3.24}$$

The deepest descent method described in Section 3.3 can then be used to minimize the KL divergence. In order to derive the update law, the gradient of the KL divergence need to be found. The gradient of the entropy $h(Y)$ is found by:

$$
\begin{aligned}
\frac{\partial h(Y)}{\partial W} &= \frac{\partial}{\partial W}\left(h(WX)\right) \\
&= \frac{\partial}{\partial W}\left(h(X) + \log\left|\det(W)\right|\right) \\
&= \frac{\partial}{\partial W}\left(\log\left|\det(W)\right|\right) \\
&= \frac{1}{\det(W)}\frac{\partial}{\partial W}\left(\log\left|\sum_{k=1}^{m} w_{ik} A_{ik}\right|\right) \\
&= \frac{A_{ik}}{\det(W)} = \left(W^{-1}\right)^{T}
\end{aligned}
\tag{3.25}
$$

$\tilde{h}(Y_i)$ can be expressed by truncating the Gram-Charlier (GC) expansion [36] of the corresponding PDF at different level. For example, PDF can be expressed as

$$
\begin{aligned}
\tilde{f}_{Yi}(y_i) \approx & \frac{1}{\sqrt{2\pi}}\exp\left(-y_i^2\right) \\
& \cdot\left(1 + \frac{\kappa_{i,3}}{3!}H_3(y_i) + \frac{\kappa_{i,2}^2}{4!}H_4(y_i) + \frac{\kappa_{i,6} + \kappa_{i,3}^2}{6!}H_6(y_i)\right)
\end{aligned}
\tag{3.26}
$$

where $H_k(y_i)$ are Hermite Polynomials and $\kappa_{i,k}$ are the cumulants of Y_i.

By taking the log of the equation above and using the Taylor expansion, $\tilde{h}(Y_i)$ can be expressed as

$$
\begin{aligned}
\tilde{h}(Y_i) \approx & \frac{1}{2}\log(2\pi e) - \frac{\kappa_{i,3}^2}{12} - \frac{\kappa_{i,4}^2}{48} - \frac{\left(\kappa_{i,6} + 10\kappa_{i,3}^2\right)^2}{1440} \\
& + \frac{3\kappa_{i,3}^2\kappa_{i,4}}{8} + \frac{\kappa_{i,3}^2\left(\kappa_{i,6} + 10\kappa_{i,3}^2\right)}{24} + \frac{\kappa_{i,4}^2\left(\kappa_{i,6} + 10\kappa_{i,3}^2\right)}{24} \\
& + \frac{\kappa_{i,4}\left(\kappa_{i,6} + 10\kappa_{i,3}^2\right)^2}{64} + \frac{\kappa_{i,4}^3}{24} + \frac{\left(\kappa_{i,6} + 10\kappa_{i,3}^2\right)^3}{432}
\end{aligned}
\tag{3.27}
$$

The derivative of the cumulants $\kappa_{i,3}$ can be calculated as

$$\begin{aligned}\frac{\partial \kappa_{i,3}}{\partial W} &= \frac{\partial}{\partial W}\left(E\left[Y_i^3\right]\right) \\ &= E\left[\frac{\partial}{\partial W}\left(Y_i^3\right)\right] = E\left[Y_i^2 \frac{\partial}{\partial W}\left(\sum_{j=1}^{m} w_{ij} X_i\right)\right] \\ &= E\left[Y_i^2 X\right]\end{aligned} \tag{3.28}$$

The derivatives of other cumulants can be derived in a similar way. The final update law for the ICA is

$$\begin{aligned}W(n+1) &= W(n) + \Delta W(n) \\ &= W(n) - \eta \frac{\partial}{\partial W}\left(D_{f\|\tilde{f}}\right) \\ &= W(n) + \eta\left[W(n)^{-T} - \varphi(y) X^T\right] \\ &= W(n) + \eta\left[I - \varphi(y) X^T W(n)^T\right] W(n)^{-T} \\ &= W(n) + \eta\left[I - \varphi(y) y^T\right] W(n)^{-T}\end{aligned} \tag{3.29}$$

where $\varphi(y)$ is the activation function derived from the Gram-Charlier expansion described above and η is the learning rate factor, controlling the rate of adaptation and convergence of the algorithm. Depending on the order of the Gram-Charlier series, $\varphi(y)$ can have different expressions. One typical activation function is

$$\begin{aligned}\varphi(y) = &\frac{1}{2} y_i^5 + \frac{2}{3} y_i^7 + \frac{15}{2} y_i^9 + \frac{2}{15} y_i^{11} - \frac{112}{3} y_i^{13} \\ &+ 128 y_i^{15} - \frac{512}{3} y_i^{17}\end{aligned} \tag{3.30}$$

The iteration is continued until convergence criteria are met. The resulting W is the de-mixing matrix that will separate individual sources from the mixture. The separated sources can be computed using equation (3.20) Although, the algorithm is complex to derive but it is very simple to execute.

Recently, researchers have proposed and developed different ICA algorithms to improve its performance. One of the main differences among these algorithms is the estimation of the PDF of Y_i (the estimated original sources). That is, different forms of equation 3.26 will lead to different ICA algorithms. In this research, an ICA algorithm called EFICA [37] is used. The accuracy given by the residual variance reaches the Cramer-Rao lower bound [38] and therefore this algorithm is asymptotically efficient or Fisher efficient.

It also needs to be noted that the recordings from the microphones have to be "different" enough to contain enough "information" about the sources. Otherwise, it will lead to a singular or ill-conditioned correlation matrix and the ICA algorithm will become unstable or inaccurate.

3.4.2 Performance Evaluation

To test the performance of the linear ICA described above, both instantaneous/linear mixtures and convolutive mixtures were used. In the instantaneous mixtures, the recordings were simulated through linear superposition of the sources as

$$x_i(n) = \sum_{i=1}^{m} A_{im} s_m(n) \tag{3.31}$$

In convolutive mixtures, the recordings are the combinations of filtered sources, shown below

$$x_i(n) = \sum_{j=1}^{d} \sum_{\tau=1}^{M_{ij}} h_{ij}(\tau) s_j(n-\tau) \tag{3.32}$$

Here, x_i is the ith observed signal or recording, s_m is the mth source signals, A is the mixing matrix, and h_{ij} is the filter between the ith microphone and the mth source.

The difference between Equations (3.31) and (3.32) is that the elements in the unknown mixing matrix A in (3.31) is replaced by an unknown filter h_{ij} and the matrix multiplication is replaced by convolution.

The performance of EFICA was also tested using both the instantaneous mixtures and the convolutive mixtures. For the instantaneous mixtures, one input channel was obtained by the direct addition of the traffic noise and the impact signal; the other input channel is also a linear addition of the traffic noise and impact signal but at a different ratio. For the convolutive mixtures, the first channel was the same as that of the instantaneous channel, but the second channel was a mixture of the filtered version of the traffic noise and impact signal. The length of the filter was assumed to be 5. The outputs of EFICA for both cases are shown in Figure 3.9 and Figure 3.10, respectively. As can be seen from the results, EFICA performs very well for instantaneous mixtures and the original signal was successfully extracted from the noisy recordings. However, EFICA cannot separate signals from the convolutive mixtures. The reason for this is that the convolution operation brings delayed versions of sources into the mixture. The delayed versions of

sources are considered independent by the algorithm and there are now more independent sources than recordings. The ICA problem therefore becomes indeterminate. The different performances of EFICA on these two cases are also reflected in the SDR of the recovered signals: the SDR is 71.34 dB for linear mixtures and is only -10.81 dB for convolutive mixtures.

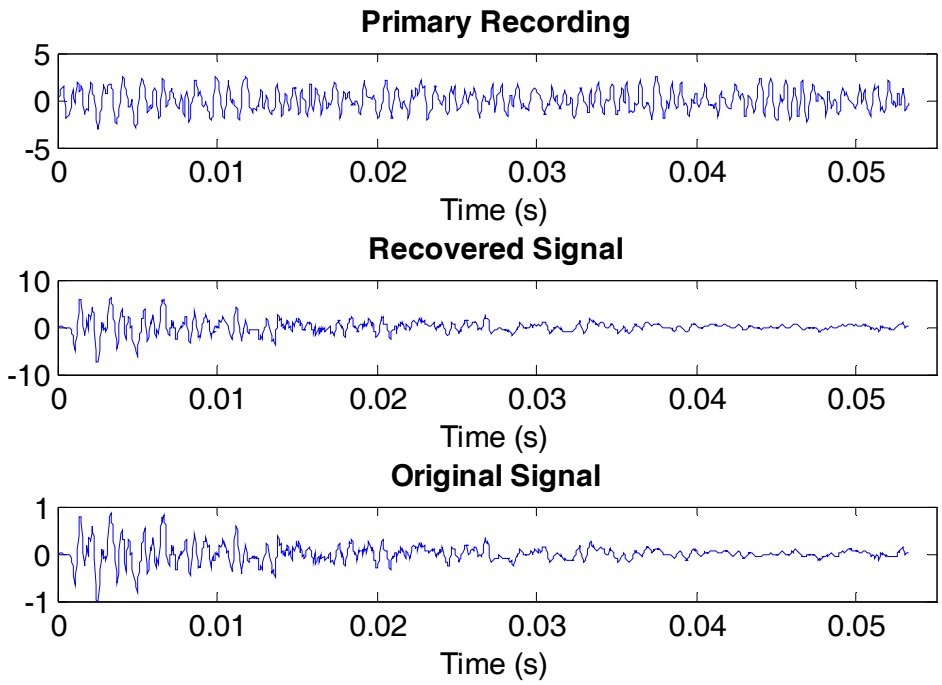


Figure 3.9 Performance of EFICA for Instantaneous Mixture

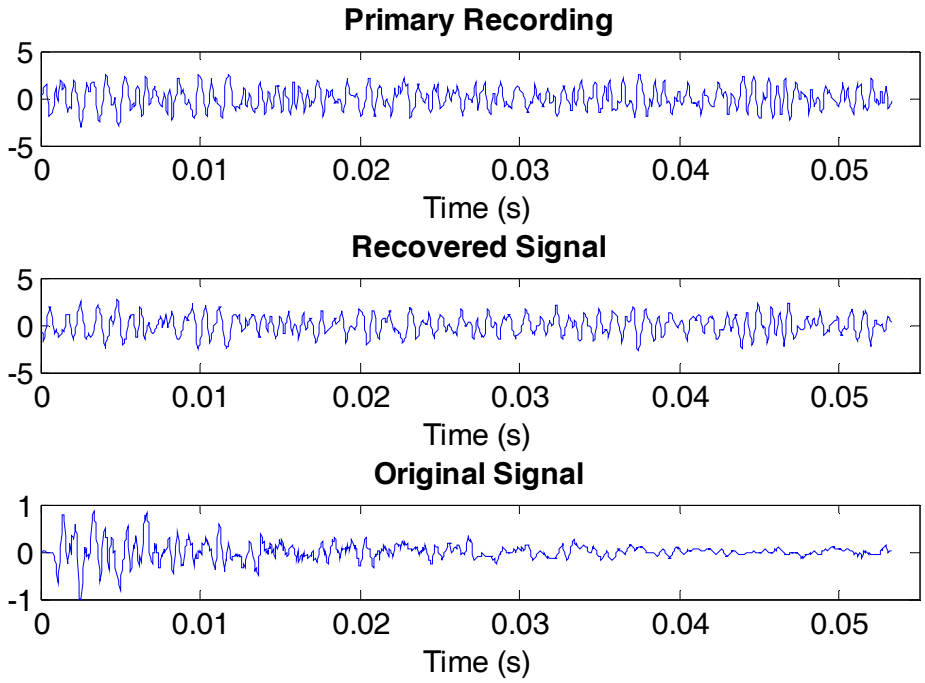


Figure 3.10 Performance of EFICA for Convolutive Mixture

3.5 Modified ICA

3.5.1 Theory Background and Procedures

As discussed in the previous section, in order for traditional ICA to work, the following two requirements must be satisfied: (1) the recordings must be a linear mixture of the sources; and (2) the recordings must be different enough. To satisfy the first requirement, the microphones should be placed as close as possible, but this conflicts with the second requirement since the microphones that are close are likely to record very similar signals. To meet the second requirement, the microphones should be placed at different locations, but this will make the recordings from the two microphones become convolutive mixtures, from which the sources cannot be separated by traditional ICA

algorithms. Intuitively, the second condition is of more importance and has to be satisfied first. This requires a solution to the convolutive problem.

From Equation (3.32), it can be seen that the convolutive mixture is in fact a linear mixture of shifted and scaled version of the sources. If the shifted version of the recordings were de-mixed by instantaneous ICA (such as EFICA), the output is a shifted and scaled version of the original sources. Based on this concept, a modified ICA algorithm in the time domain [39] is used to perform the source separation of convolutive mixture. The procedure for the modified ICA algorithm is described as follows.

Step 1: A "convolutive sphering" on the recordings is performed, as shown below. In this step, the delayed version of the recording is used as additional recordings.

$$\tilde{X} = \begin{bmatrix} x_1(n) & x_1(n-1) & \dots & x_1(n-L+1) \\ x_1(n-1) & x_1(n-2) & \dots & x_1(n-L) \\ \dots & \dots & \dots & \dots \\ x_1(n-L) & x_1(n-L-1) & \dots & x_1(1) \\ x_2(n) & x_2(n-1) & \dots & x_2(n-L+1) \\ \dots & \dots & \dots & \dots \\ x_2(n-L) & x_2(n-L-1) & \dots & x_2(1) \\ \dots & \dots & \dots & \dots \\ x_m(n-L) & x_m(n-L-1) & \dots & x_m(1) \end{bmatrix} \tag{3.33}$$

where $\tilde{X}$ is the rearranged input with a dimension of $mL \times (n-L)$, x_i is the ith observed signal or recording, L is the number of delays, n is the length of the block from where the source can be separated or estimated, and m is the number of recordings.

Step 2: The rearranged input $\tilde{X}$ is de-mixed by traditional ICA algorithms (such as EFICA). The output can be calculated using Equation (3.33). It can be shown that the de-mixed outputs c are a delayed and scaled source signals [40]:

$$c(n) = W\tilde{x}(n) \tag{3.34}$$

where W is obtained through an ICA algorithm.

Step 3: The similarities or the distance between each de-mixed output (or independent components) from Step 2 is calculated based on correlation-based criteria. To do this, vector $\tilde{c}_i$ and a time shift operation are defined as

$$\tilde{c}_i = \begin{bmatrix} c_i(L+1) & c_i(L+2) & \dots & c_i(N-L) \end{bmatrix}^T \tag{3.35}$$

$$D^k \tilde{c}_i = \begin{bmatrix} \tilde{c}_i(L+1+k) & c_i(L+2+k) & \dots & c_i(N-L+k) \end{bmatrix}^T \tag{3.36}$$

Then the distance between the ith and j^{th} independent components can be calculated as:

$$D_{ij} = \left\| \tilde{c}_j - \tilde{C}_i \left(\tilde{C}_i^T \tilde{C}_i \right)^{-1} \tilde{C}_i^T \tilde{c}_j \right\|^2 \tag{3.37}$$

where

$$\tilde{C}_i = \begin{bmatrix} D^{-L}\tilde{c}_i & D^{-L+1}\tilde{c}_i & \dots & D^{L-1}\tilde{c}_i & D^L\tilde{c}_i \end{bmatrix} \tag{3.38}$$

Step 4: The independent components from Step 2 are grouped into m groups based on the similarity matrix D in Step 3, where m is the number of sources. Here, a hierarchical clustering algorithm using an average-linkage method is used. The method is described as follows:

Assign each IC to a cluster. If there are n ICs, there are now n clusters;

Find the closest (most similar) pair of clusters and merge them into a single cluster. The number of clusters is now reduced by one and becomes $n-1$;

Compute distances between the new clusters using the average-linkage strategy (the new distance is the average distance of the two merged clusters);

Repeat steps 2 and 3 until the number of clusters is reduced to the target. In this case, the target is the number of sources.

Step 5: The contribution of source i to $\tilde{X}$ in Step 1 is determined by the inverse of the de-mixing process, in which the de-mixing matrix corresponding to source i is computed based on the similarity matrix in Step 3 and clustering:.

$$\tilde{X}^i = W^{-1}\mathrm{diag}\left[\lambda_1^i \quad \ldots \quad \lambda_{mL}^i\right]c \tag{3.39}$$

where $\tilde{X}^i$ is the contribution of source i to $\tilde{X}$ and c is the independent components computed in step 2. λ_k^i are the weighting factors computed from

$$\lambda_k^i = \left(\frac{\sum_{j\in K_i, j\neq k} D_{kj}}{\sum_{j\notin K_i, j\neq k} D_{kj}}\right)^{\alpha} \tag{3.40}$$

where K_i are the indices belonging to cluster i and α is a positive factor that controls the "hardness" of the weighting.

The influence of source i on microphone k is defined as

$$\hat{s}_k^i(n) = \sum_{p=1}^{L} \tilde{X}^i_{(k-1)L+p}(n+p-1) \tag{3.41}$$

Step 6: The sources are reconstructed from $\tilde{X}^i$ by inverting the "convolutive sphering" process in Step 1.

The process of the algorithm is briefly shown in Figure 3.11

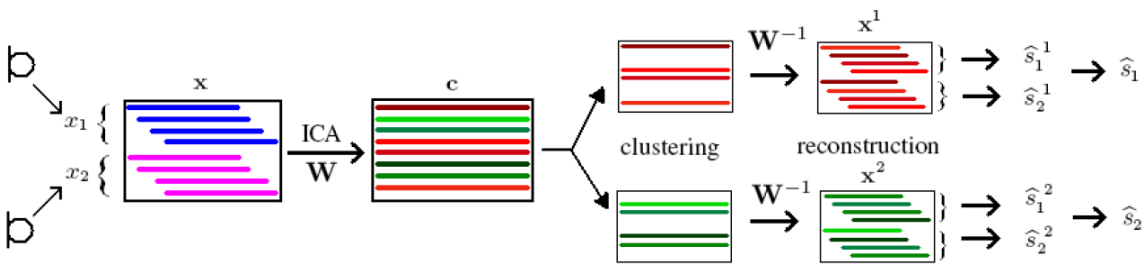


Figure 3.11 Modified ICA [39]

3.5.2 Performance Evaluation

To compare the modified ICA algorithm and the traditional ICA described in Section 3.4, both an instantaneous mixture and a convolutive mixture was used to test the performance. The signal used for performance evaluation was the same as in Section 3.4. The results are shown in Figure 3.12 and Figure 3.13, respectively, and indicate that the impact signal was successfully recovered by the modified ICA algorithm for both types of mixtures. The SDRs of the algorithm for instantaneous and convolutive mixtures were 3.609 dB and 1.210 dB, respectively. Even though the SDRs of the modified ICA is not as high as that of the EFICA for the linear mixture case, the performance on convolutive mixtures far exceeds that of EFICA. In fact, when the recovered signal was played back,

no significant difference was detected between the original source and the recovered signal from the modified ICA.

This algorithm still requires that the delay has to be predefined for successful separation. However, since both microphones are located on the impacting cart and are separated only by a small distance, the estimated delay is not large and can be estimated through

$$L = \frac{F_s}{v/\left(d_1 - d_2\right)} \tag{3.42}$$

where F_s is the sampling frequency (Hz), d_1 and d_2 are the distances between the microphones and the impact point and v is the velocity of sound in air.

If the distance difference of the two microphones is one meter and the velocity of the sound in air to be 340 m/s, a delay of 25 samples is enough for successful separation at a sampling frequency of 8000 Hz.

It can be seen from the results that the modified ICA works on both instantaneous and convolutive mixtures. The delay needs to be pre-defined for successful separation, but can be easily estimated.

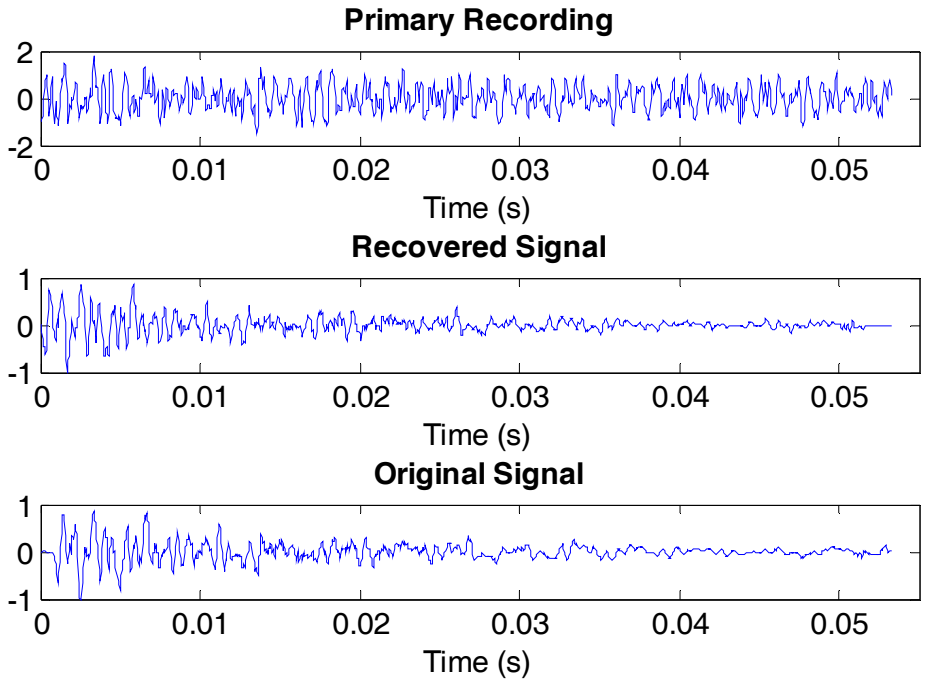


Figure 3.12 Performance of Modified ICA for Instantaneous Mixture

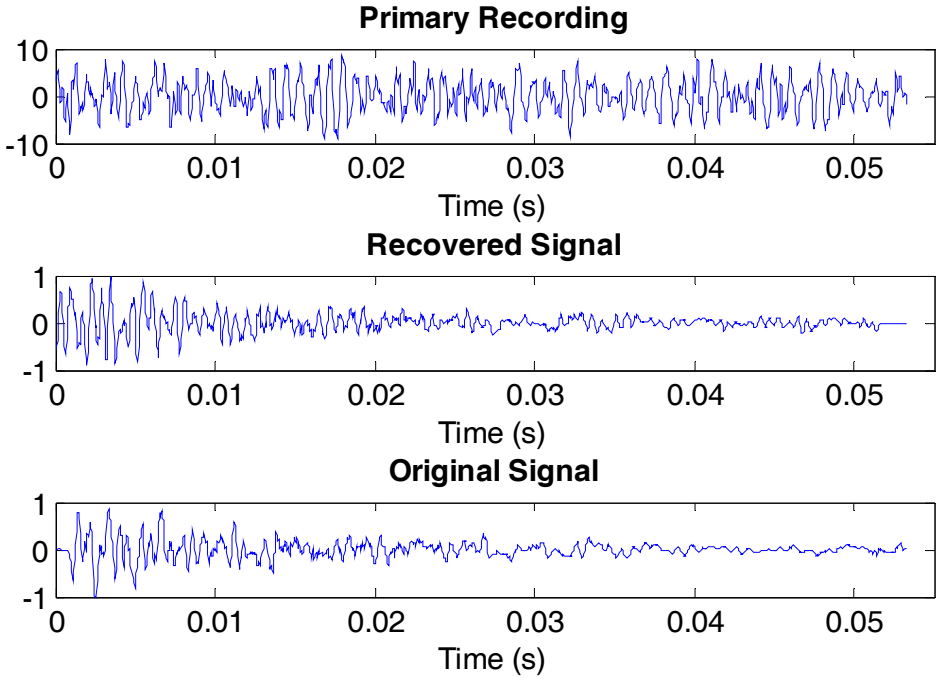


Figure 3.13 Performance of Modified ICA for Convolutive Mixture

3.6 Selection of Noise Cancelling Algorithms

Even though there are numerous types of noise cancelling algorithms, not all of them can be used for acoustic delamination detection. Field inspection indicates that the following four requirements must be satisfied by an effective algorithm:

1. The algorithm must be able to cancel/separate non-stationary sources to be effective with changing the traffic noise in adjacent lanes.

2. No prior information about sources should be required since the traffic noise cannot be predicted or controlled and the impact sound changes from case to case.

3. There should be no strict requirement on the recording device, such as directionality, etc.

4. The algorithm should work for convolutive mixtures, if two or more microphones are used.

Table 3.1 compares different noise cancelling algorithms mentioned in this chapter the with respect to the above four requirements.

Table 3.1 Comparison of Noise Cancelling Algorithms.

	Requirement 1	**Requirement 2**	**Requirement 3**	**Requirement 4**
Spectral Subtraction	X	√	√	N.A
RLS	√	√	X	√
ICA	√	√	√	X
Modified ICA	√	√	√	√

To quantitatively compare the performance of the noise cancelling algorithms, Table 3.2 lists the SDRs of the algorithms. Since the recordings at the site are usually convolutive mixtures, only the results for the convolutive mixture are listed.

Table 3.2 Performance for Convolutive Mixtures.

	Spectral Subtraction	**RLS**	**ICA**	**Modified ICA**
SDR (dB)	-6.514	-6.212 (high signal leakage) -2.334 (low signal leakage)	-10.81	1.210

It is obvious from Table 3.1 and Table 3.2 that the modified ICA meets all the requirements for field inspections and has the best overall performance under real scenarios (convolutive mixtures). Therefore, the modified ICA was selected as the noise cancelling algorithm for acoustic delamination detection.

3.7 Summary

This chapter described the evaluation criteria for noise cancelling algorithms and the technical details of commonly used noise cancelling algorithms.

The performance evaluation of noise cancelling needs to be simple and objective.An objective performance measure, the SDR, was introduced. Then, the technical details of four commonly used noise cancelling or source separation algorithms (spectral subtraction, recursive least square adaptive filter, independent component analysis and modified independent analysis) were described. The SDR and time-domain signal comparisons were used to evaluate the performance of each individual algorithm. Each algorithm performed differently under different types of recordings and has its own advantages and disadvantages.

Spectral subtraction is very simple and easy to implement. It uses recordings from only one microphone. The noise in the recordings is estimated from the spectrum during the quiet period and is then subtracted from the spectrum of the impact period. However, this algorithm requires that the noise signal be short-term stationary, which is hard to guarantee in field inspections.

In the RLS algorithm, an adaptive filter is used to estimate the noise recorded by the primary microphone using the recording from the reference microphone. Noise in the primary recording is estimated from the reference recording by minimizing the MSE between the output of the adaptive filter and the desired output. The signal is recovered by subtracting the estimated noise from the primary microphone. RLS adaptively adjusts the coefficients of the filter and can work with non-stationary signals, but requires the signal from the reference microphone to be pure noise, which is hard to satisfy in field inspection. It also needs a good estimate of the filter length.

A Fisher efficient ICA algorithm called EFICA is employed to release the requirement that there should be no signal in the recording by the reference microphone. This algorithm maximizes the independence between the outputs to separate sources from the mixture by an adaptive de-mixing matrix. The coefficients of the de-mixing matrix are adaptively changed. The algorithm can separate sources without prior information about them. However, it requires that the recordings from the two microphones be linear mixtures of the signal and noise, which is not the case for the signals recorded at bridge sites.

A modified ICA is therefore proposed to separate the signal from a convolutive mixture. This method can be used to separate both linear and convolutive mixtures. Even though the delay still needs to be estimated before the separation, this is not a problem in practice, because the separation between the two microphones from the impact point is relatively small compared to the speed of sound. The delay can be estimated without much error.

The candidate algorithms are then compared and evaluated by considering both the requirements of field inspection and the performance for the convolutive mixtures, which is representative of signals recorded in the field. The modified ICA performed the best and was selected for the remainder of the research.

(This page intentionally left blank.)

CHAPTER 4

FEATURE EXTRACTION

After the noise in the recordings was removed by implementing the noise canceling algorithm described in Chapter 3, the next step in the delamination detection is to relate the characteristics of the acoustic signals with the existence of delamination. As mentioned in Chapter 2, the delamination of the concrete bridge deck is characterized by a dull, hollow sound. This criterion is subjective and difficult to implement in an automatic detection algorithm. An objective criterion is needed to separate "the hollow sound" and "the solid sound". The characteristics of the signal can be obtained by extracting features that quantify the acoustic signals. This step also reduces the dimension of the signal to avoid "the curse of dimensionality" [41]. The extracted features are further selected based on different selection criteria to eliminate features that are irrelevant to target concepts [42]. This chapter describes and compares different feature extraction algorithms and selects the best algorithm for delamination detection.

4.1 Feature Extraction of Acoustic Signals

Acoustic signals are usually quantified using different models. Each signal in these models is represented by parameters, called features of the signal. Different models represent a signal in different ways and extract different features of the signal. For example, the Fourier Transform (FT) expresses the signal in the frequency domain and extracts frequency features of the signal while the Wavelet Transform represents the

signal in the wavelet domain and extracts different features of the same signal. Several features for acoustic signals are described in the following sections.

4.1.1 Sub-band Energy

Frequency components are probably the most widely used features in the processing of acoustic signals since they have a clear physical meaning. To obtain the features in the frequency domain, the Fourier Transform (FT) that represents the signal in the forms of sinusoids with different frequencies is used. When the signal used for detection is digitized and discrete, the Discrete Fourier Transform (DFT) is used. The computation of DFT is described below.

$$X_k = \sum_{n=0}^{N-1} x_n \exp\left(-\frac{2\pi i}{N} kn\right) \ , \ k = 0,1,...,N-1 \tag{4.1}$$

where $N/2+1$ is the number of discrete frequencies. Upon taking the DFT of the signal, the signal is represented by frequency domain features and the dimension of the feature vectors is N. In order to reduce the dimension of the feature vectors, N should be small. However, a small N leads to a decrease of resolution in the frequency domain. A bigger N yields higher resolution in the frequency domain, but this increases the dimension of the feature space and defeats the purpose of dimension reduction. Also, due to the short duration of the impact signal used for delamination detection, the variance of the DFT can be large and its repeatability is poor. Sub-band energy can be used to reduce the dimension and the variance of the frequency domain features. The entire frequency domain is evenly divided into several (for example, 20) sub-bands and the energy in each sub-band is calculated as follows and used as features for delamination detection:

$$E_i = \sum_{\omega \in F_i} \left| X(\omega) \right|^2 \tag{4.2}$$

This is equivalent to passing the signal through a different series of band-pass filters with different cut-off frequencies. This filter series is called a filter bank. The shape of the filter bank in this case is rectangular and the filters are evenly spaced on the frequency axis, as shown in 0. The energy of the filtered signal is extracted as features. Rectangular Filter Bank

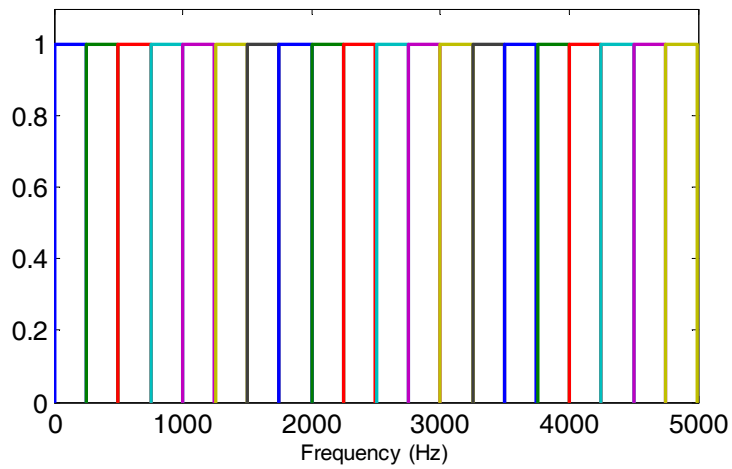


Figure 4.1 Rectangular Filter Bank

4.1.2 Energy of Wavelet Packet Tree

Another set of commonly used features in signal processing are obtained in the wavelet domain. Similar to the Fourier Transform, the Wavelet Transform (WT) decomposes the signal using different basis functions. The difference between the WT and the FT is in the selection of the basis functions. In the Fourier Transform, the basis functions are a family of sinusoids with infinite support in the time domain, while the basis functions for WT are scaled and shifted versions of wavelet functions, usually with

a finite support. The wavelet transform provides more flexibility in choosing the type of basis functions. In addition, the short support of the basis functions can capture transient information about the signal. Also the scaling and shifting of basis functions make the wavelet representation capable of representing the signal at different resolution levels in both the time and frequency scales. The wavelet is defined by two bases, the scaling function and the wavelet function. The scaling function captures the base shape of the signal and the wavelet function is responsible for capturing details of the signal. In the frequency domain, the scaling function is equivalent to filtering the signal through a low pass filter and the wavelet function is a high pass filter. Having defined the basis functions, the wavelet transform can be expressed as [43]:

$$X(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)\psi\left(\frac{t-b}{a}\right)dx \tag{4.3}$$

where $\psi(\square)$ is the basis function, and a and b are the scaling and shift factors.

In the traditional wavelet transform, the signal is decomposed into two components by the scaling and wavelet functions. These two components are called approximation and detail coefficients, respectively. In the second step, the approximation coefficients are further decomposed by the scaling and the wavelet functions at the second level. This process continues until the required level of decomposition is reached. In this way, the signal is expressed by the approximation and detailed coefficients at different levels. The process of the wavelet transform is shown in Figure 4.2.

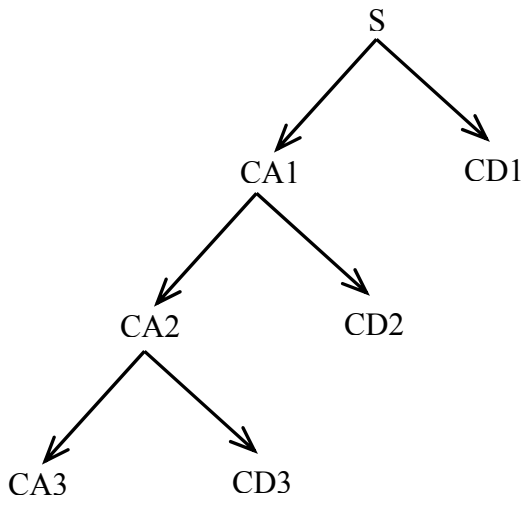


Figure 4.2Wavelet Decomposition [44]

The wavelet decomposition is complete, meaning that no redundant information is included. This completeness of the wavelet transform is good for representing the signal but may not be good for feature extraction. The purpose of the feature extraction is to find the features for classification purposes. Redundancy in this case provides more flexibility in the selection of features. The wavelet packet is a redundant way of representing the signals, can provide such flexibility, and is therefore is used in this research. The wavelet packet decomposition and the wavelet transform is the same except that in each step, both the approximation coefficients and the detail coefficients are decomposed, which is equivalent to passing the coefficients from the previous level through a high pass and a low pass filter. Figure 4.3 shows an example of level 3 wavelet packet decomposition. After the signal was decomposed, the Shannon entropy of each sub-band at the lowest level in the wavelet packet tree can be used as features. This method can also be considered as a filter bank method. The shape or the frequency response of each filter is dependent on the type of wavelet function used.

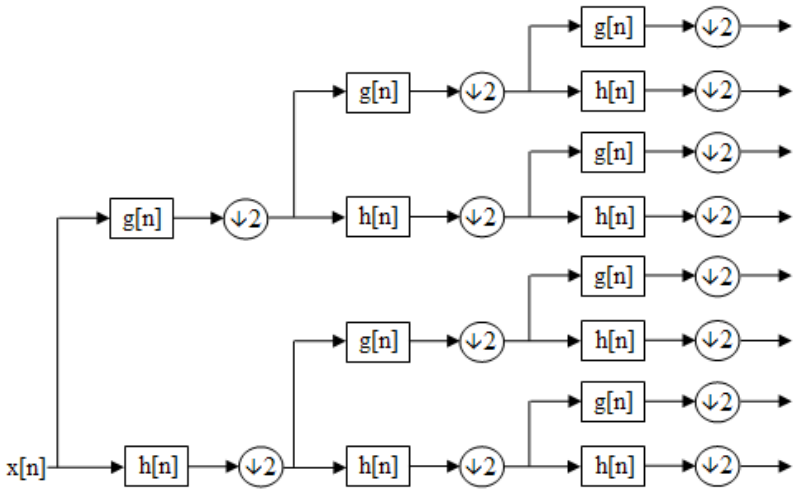


Figure 4.3Wavelet Packet Decomposition [45]

4.1.3 Psycho-Acoustic Features

Even though detection of delamination by the "hollowness" of the impact signal may be subjective, it is undeniable that that the human ear is good at detecting differences in sounds. To understand how the human ear discriminates sound, research has been conducted in the field of psycho-acoustics [46]. Many feature extraction algorithms based on psycho-acoustic models have been proposed. Mel-Frequency Cepstral Coefficients (MFCC) [47] is one of the psycho-acoustics based feature extraction algorithms for speech recognition and detection. MFCC approximates the human auditory system's response more closely than the linearly-spaced frequency bands shown in Section 4.1.1

MFCC can be calculated using the following steps:

1. Split signal into frames; in this case, each impact signal is considered a frame.

2. For each frame, smooth with the Hamming window as shown in Figure 4.4 and then compute the Fast Fourier Transform (FFT).

3. Calculate the power spectrum of the framed signal by squaring the FFT.

4. Filter the power spectrum obtained in Step 3 through a Mel-frequency filter bank. The filters are equally distributed on the Mel-scale. The relationship between the Mel-scale frequency and regular frequency measurement (Hz) is:

$$m = 1027.01048 \log_e \left(1 + \frac{f}{700} \right) \tag{4.4}$$

where m is the Mel-frequency and f is in Hz.

Different filter shapes may be used. In this study, the triangular shape is selected as shown in Figure 4.5. Adjacent filters are overlapped over half of the bandwidth.

5. Apply the discrete cosine transform (DCT) to the log of the spectrum filtered by the Mel-frequency filter banks. The DCT can be calculated as:

$$DCT_x(k) = w(k) \sum_{n=1}^{N} x(n) \cos \frac{\pi (2n-1)(k-1)}{2N} \ , \ k = 1, \ldots, N \tag{4.5}$$

where: $w(k) = \begin{cases} 1/\sqrt{N}, & k = 1 \\ \sqrt{2/N}, & 2 \le k \le N \end{cases}$

6. Obtain the MFCC by applying a filter to the output of the DCT (to smoothen the MFCC). In this case, the filter has a half sine wave shape.

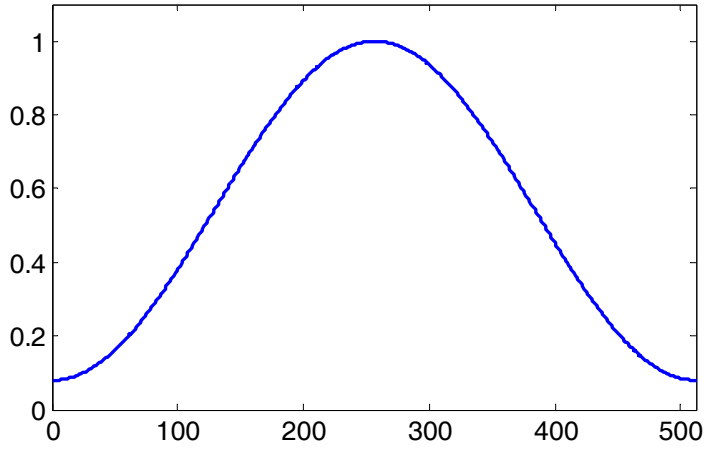


Figure 4.4 Hamming Window

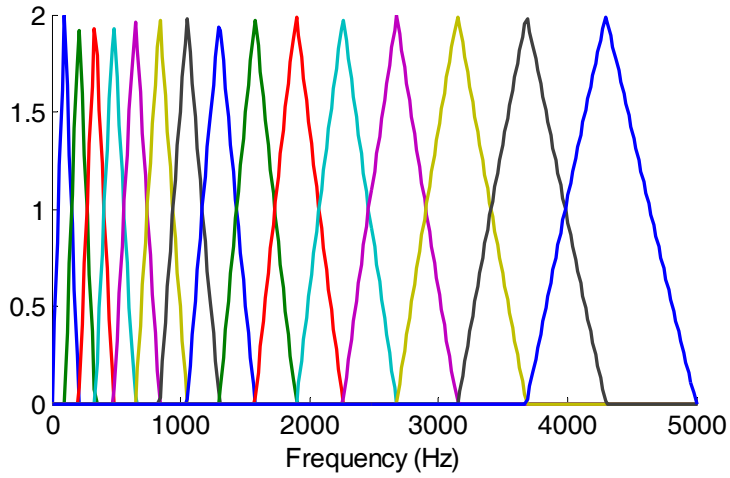


Figure 4.5 Mel-Frequency Filter Banks

There are different variations of the MFCC. For example, other types of frequency scales such as the Bark scale [46] which ranges from 1 to 24 Barks, corresponding to the first 24 "critical bands" of the human hearing system based on the results of psychoacoustic experiments, can be used. In addition to the different frequency scales, the shape of the filter bank can be different to optimize the performance of the feature extraction algorithms. For example, the shape of the filter bank at different frequencies can be optimized by Principal Component Analysis [48].

4.1.4 Principal Component Analysis

When the data is transformed from the "data space" to the "feature space", it is desirable that the original data is represented by "effective" features with a lower dimension while retaining most of the information. One way of measuring the effectiveness is the amount of energy loss during the dimension reduction. A good representation will keep as much energy as possible using the least number of features. Principal component analysis (PCA) [48] can be used to achieve this goal. It finds the optimal linear transformation such that the extracted features are the best representation of the original signal in the sense of mean square error.

Assume the dimension of the original data x is n, and that the data can be represented by n orthogonal unit basis vectors u_i, where $i = 1,...,n$, as

$$x = y_1 u_1 + ... + y_n u_n \tag{4.6}$$

Therefore,

$$y_j = x^T u_j \tag{4.7}$$

The extracted features can be found by truncating Equation (4.6) as:

$$\hat{x} = y_1 u_1 + ... + y_m u_m \tag{4.8}$$

where m is the dimension of the features and $m < n$.

The error between the original data and the extracted features is found to be:

$$\begin{aligned} \varepsilon &= x - \hat{x} \\ &= \sum_{j=m+1}^{n} y_j u_j \end{aligned} \tag{4.9}$$

The mean square error is:

$$\begin{aligned} MSE &= E\left(\varepsilon^2\right) \\ &= E\left[\left(\sum_{i=m+1}^{n} y_i u_i\right)^T \left(\sum_{j=m+1}^{n} y_j u_j\right)\right] \\ &= \sum_{i=m+1}^{n} E\left(y_j^2\right) \end{aligned} \tag{4.10}$$

where $E(\cdot)$ is the expectation operation and

$$\begin{aligned} E\left(y_j^2\right) &= E\left[\left(u_j^T x\right)\left(x^T u_j\right)\right] \\ &= u_j^T R u_j \end{aligned} \tag{4.11}$$

where R is the correlation matrix of x.

Substituting Equation (4.11) into (4.10) yields:

$$MSE = \sum_{j=m+1}^{n} u_j^T R u_j \tag{4.12}$$

where $u_j^T u - 1 = 0$

The minimization of the mean square error can be found by using a set of Lagrangian multipliers and setting the derivative of the mean square error to zero.

$$\begin{aligned} \frac{\partial \xi}{\partial u_j} &= \frac{\partial}{\partial u_j}\left[\sum_{j=m+1}^{n} u_j^T R u_j - \lambda\left(u_j^T u - 1\right)\right] \\ &= 2Ru_j - \lambda u_j = 0 \end{aligned} \tag{4.13}$$

It can be shown that u_j is the eigenvector of the correlation matrix R. The mean square error can then be found to be:

$$MSE = \sum_{j=m+1}^{n} \lambda_j \tag{4.14}$$

where λ_j are the eigenvalues of the correlation matrix R.

To minimize the mean square error, λ_j needs to be the smallest $n-m$ eigenvalues. Hebbian learning algorithms [35] can be used to find the eigenvectors and eigenvalues of the correlation matrix. The advantage of PCA is that it is a non-parametric analysis and the answer is unique and independent of any assumption about the probability distribution of the data.

The procedure of extracting features using PCA is briefly described below:

1. Obtain the transformation matrix using the generalized Hebbian learning algorithm.

2. Extract the features of the testing signals by multiplying these signals with the transformation matrix obtained in Step 1.

4.1.5 Independent Component Analysis

Independent Component Analysis (ICA) described in Chapter 3 can also be used to find the effective features of a signal. Unlike PCA which maximize the energy contained in the extracted features, ICA extracts features by maximizing the amount of information. In this approach, the loss of information is minimized in the process of dimension reduction. The procedure of feature extraction using ICA [49] is briefly described as follows:

1. Extract independent components from the training signal.

2. Select dominant independent components by relative importance of basis vectors. In this case, the L_2 norm of the column of the de-mixing matrix is used as the criteria for relative importance.

3. Transform the dominant ICs into the frequency domain. Signals were filtered by the filters obtained from the dominant ICs.

4. Scale the energy of the filtered signals logarithmically and compute the cepstral coefficients of the log-energy as features.

This method is similar to the MFCC described in Section 4.1.3, except that the shape of the filter bank is computed from the dominant ICs.

4.2 Performance of Different Features

Several commonly used feature extraction algorithms were described in the previous section. Different algorithms extract different features of the signal. In order to select the best features for the detection of concrete delamination, it is necessary to evaluate and compare the results of different feature extraction algorithms. This section first introduces the criteria for the evaluation of different feature extraction algorithms and then the performance of the algorithms is evaluated against these criteria.

4.2.1 Criteria for Evaluation

For acoustic methods of concrete delamination detection, the features of the acoustic signal need to have two properties. First, the features must be repeatable, i.e., the features

of the signal obtained under the same test conditions must be consistent. Second, the features of the signal must be separable, i.e., the difference between features from the solid and delaminated concrete must be large so that they can be easily separated from one another. The evaluation criteria must be numerical and dimensionless so that an objective comparison can be made.

Assuming that the features from solid and delaminated concrete are random variables, repeatability can be measured by the coefficient of variation. For multiple random variables, the repeatability of the extracted feature can then be calculated as the weighted mean value of the coefficient of variation for solid and delaminated concrete:

$$RPT = \sqrt{\frac{(N_S - 1)\left(\frac{\sigma_S}{\mu_S}\right)^2 + (N_D - 1)\left(\frac{\sigma_D}{\mu_D}\right)^2}{N_S + N_D - 2}} \tag{4.15}$$

where μ_S and μ_D are the mean values of the extracted features for signals from solid and delaminated concrete; σ_S and σ_D are the standard deviations of the features for solid and delaminated signals; N_S and N_D are the number of samples in the groups of solid concrete and delaminated concrete, respectively. A high repeatability value indicates poor repeatability of the test.

The separability of the features can be formulated as a hypothesis test: whether the features from solid concrete and from delaminated concrete has the same mean. The t statistic-based separability measure is:

$$SEP = \frac{\left|\mu_S - \mu_D\right|}{\sqrt{\frac{\left(N_S - 1\right)\sigma_S^2 + \left(N_D - 1\right)\sigma_D^2}{N_S + N_D - 2}}} \tag{4.16}$$

A higher separability measurement indicates better separation

Another separability criterion comes from information theory: the mutual information. Each feature contains a certain amount of information about the type of the concrete and mutual information is one way of measuring the information. In delamination detection, it measures the information about the type of concrete from which the signal originates. The type of concrete is called a class label. The computation of this information theoretic measurement is briefly described below [50].

Suppose the type of concrete (or class label) is a random variable. The uncertainty of the class label can be calculated as:

$$H\left(C\right) = -\sum_{c} P\left(C\right)\log\left(P\left(C\right)\right) \tag{4.17}$$

where $P\left(C\right)$ is the probability density function (PDF) of the class label C.

After observing a feature vector $\underline{x}$, the conditional entropy becomes:

$$H\left(C \mid \underline{x}\right) = -\int_{\underline{x}} p\left(\underline{x}\right)\left(\sum_{c} p\left(C \mid \underline{x}\right)\log\left(p\left(C \mid \underline{x}\right)\right)\right)d\underline{x} \tag{4.18}$$

where $p\left(\underline{x}\right)$ is the PDF of the feature vector $\underline{x}$ and $p\left(C \mid \underline{x}\right)$ is the conditional PDF of $\underline{x}$ given C.

The loss of uncertainty after the observation of a feature is called the mutual information between the feature and the class label and can be calculated as:

$$\begin{aligned} I\left(C,\underline{x}\right) &= H\left(C\right) - H\left(C \mid \underline{x}\right) \\ &= -\sum_{c}\int_{\underline{x}} p\left(c,\underline{x}\right)\log\left(\frac{p\left(c,\underline{x}\right)}{P\left(c\right)p\left(\underline{x}\right)}\right)d\underline{x} \end{aligned} \tag{4.19}$$

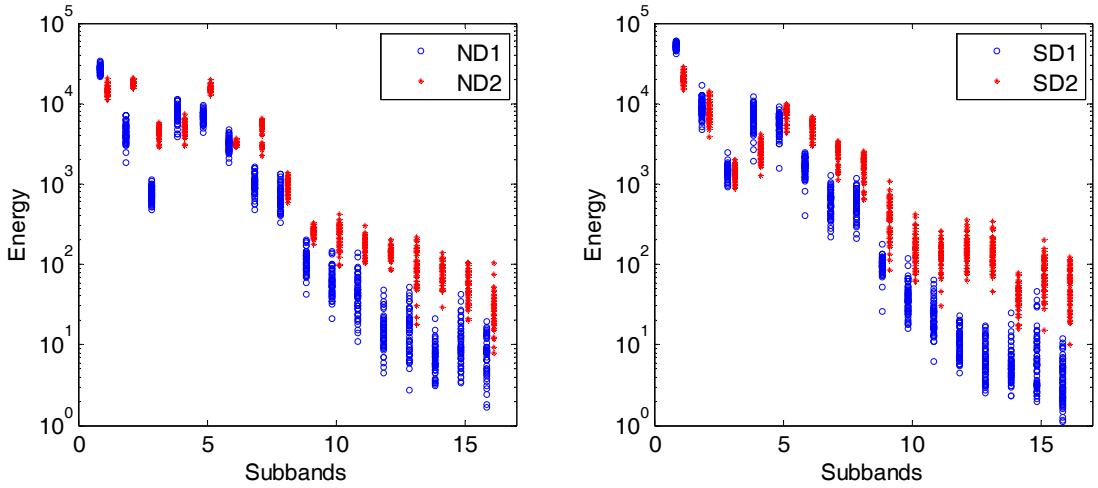
In the actual computation, the probability distribution functions are obtained directly from the available samples and estimated from the histogram of each variable. The larger the value of mutual information, the more information is contained in the feature.

To test the performance of different features, data from lab experiments were used. The test setup and experimental process is discussed later in Chapter 6. The same experiments were performed on two different days to check the repeatability and to make the detection results more representative. On the first day, 53 impact signals were obtained from non-delaminated (ND1) or solid concrete and 66 impacts were recorded from a shallow delamination (SD1) or delaminated concrete. On the second day, 52 impacts on solid concrete (ND2) were recorded and 66 impacts were from delaminated concrete (SD2). Therefore, there were 105 impact signals from solid concrete (ND) and 132 from delaminated concrete (SD). The measures of repeatability and separability were calculated for these signals.

4.2.2 Performance of Sub-band Energy

The performance of sub-band energy as a candidate feature was evaluated by repeatability and separability measures using the signals mentioned in the beginning of the section. Figure 4.6(a) compares the repeatability of the sub-band energy of signals

from non-delaminated signals (ND1 vs. ND2) and Figure 4.6(b) compares the signal from delaminated signals (SD1 vs. SD2). Figure 4.7 shows the difference between the non-delaminated signals and the shallow delaminated signals. Due to the wide range of the feature values, the vertical axis (sub-band energy) was plotted on a log scale. As can be seen from Figure 4.6, the repeatability for signals obtained on the same day is good while the repeatability between days is not as good. The data obtained on different days forms two clusters for most of the sub-bands. From Figure 4.7, it can also be observed that the energy of all sub-bands are mixed, indicating poor separability.



(a) ND1 vs. ND2 (b) SD1 vs. SD2

Figure 4.6 Repeatability of the Sub-band Energy

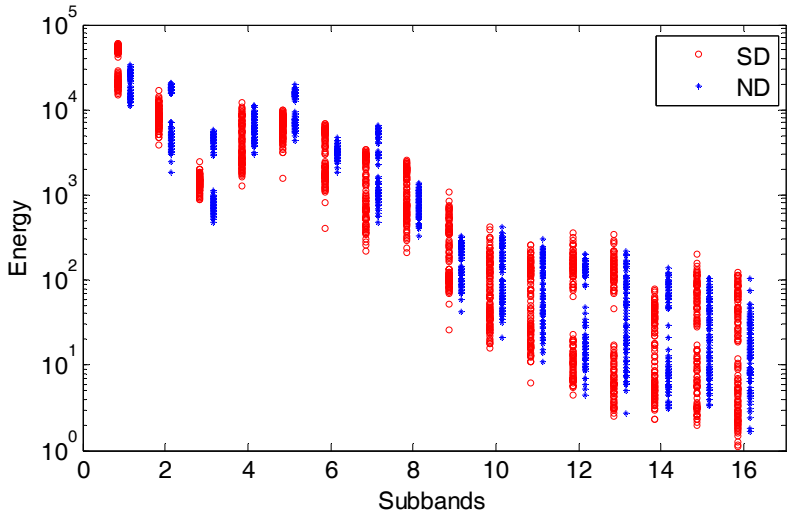


Figure 4.7 Separability of the Sub-band Energy

To quantitatively compare each individual feature, the numerical results of repeatability, separability and the mutual information between the class labels for different features are calculated and shown in Figure 4.8 to Figure 4.10. Several observations can be made. First, different features have different separability and mutual information. The reason for this comes from the fact that information that differentiates the solid concrete from the delaminated concrete does not exist in all the features. The existence of the features that cannot effectively separate the two groups will lead to a decrease in the accuracy of the detection. It is therefore necessary to select the features that are useful for detection or classification purposes. Second, even though separability and mutual information measure the ease of separating the no delamination case from the delaminated case, they are inconsistent for some sub-bands. For example, the separability measure is very small for the first three sub-bands indicating poor repeatability, while the value of the mutual information for these sub-bands is high. The reason for this is that the

separability measure provides the "distance" between the two classes while the mutual information measures the amount of the information.

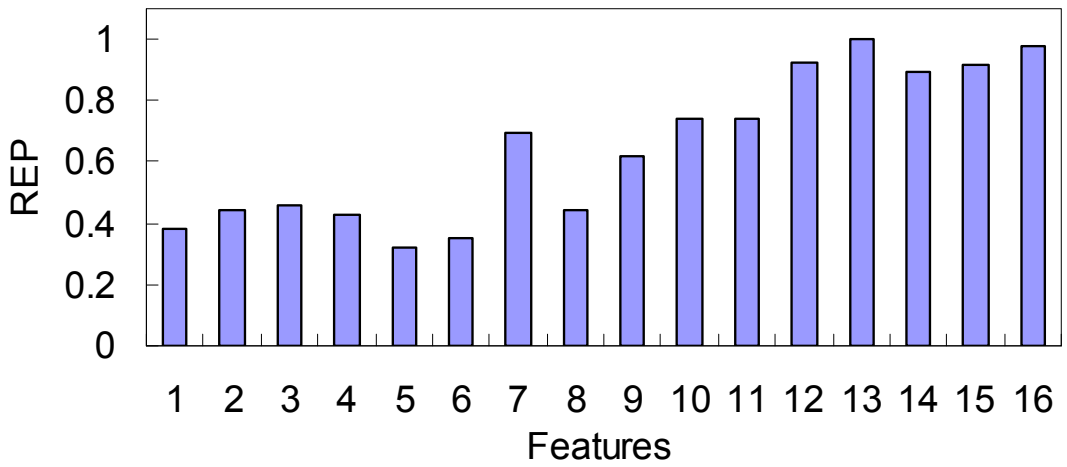


Figure 4.8 REP of the Sub-band Energy

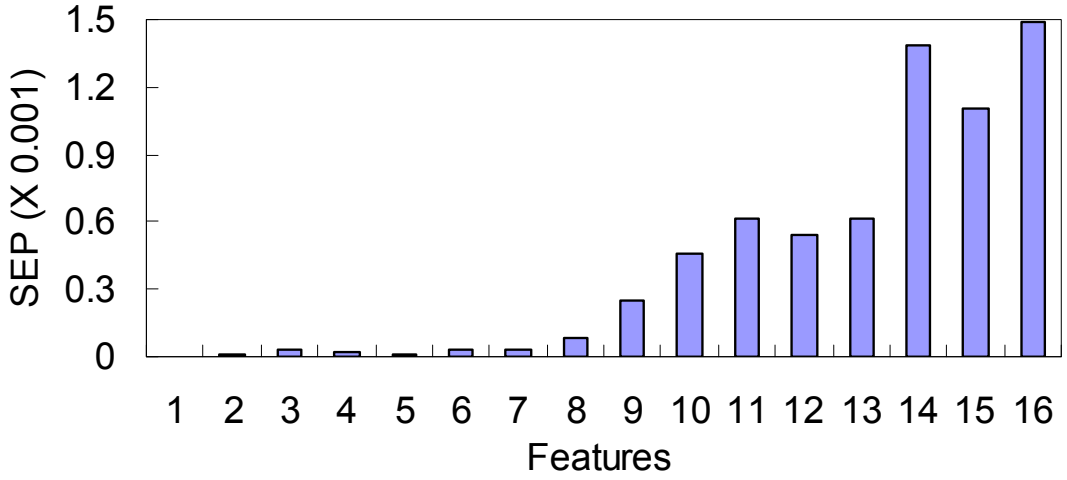


Figure 4.9 SEP of the Sub-band Energy

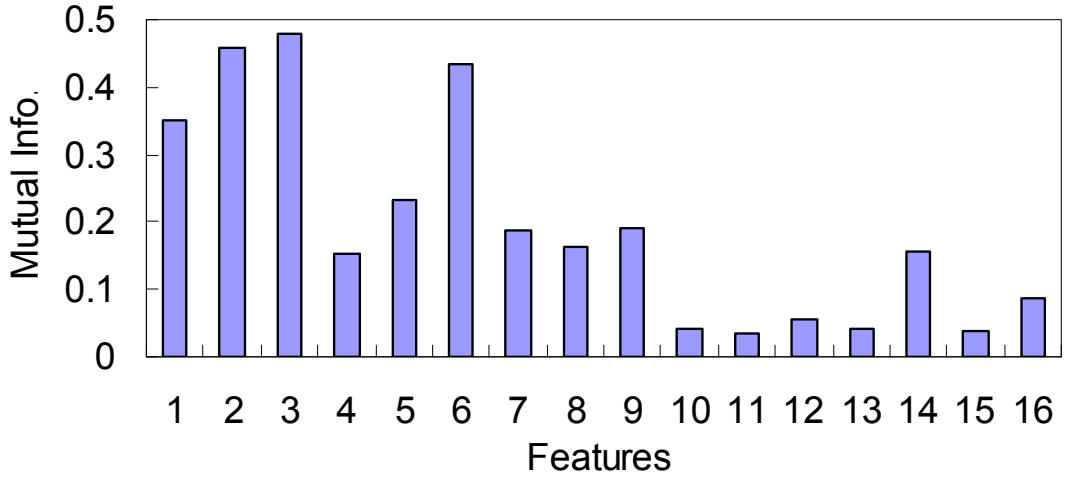
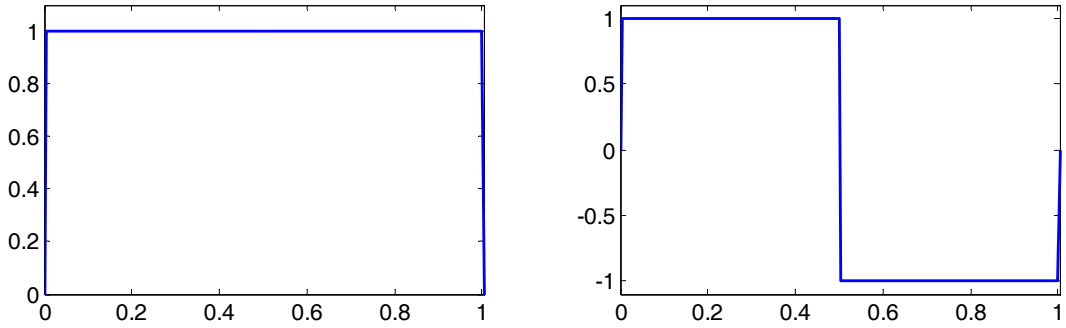


Figure 4.10 Mutual Information of the Sub-band Energy

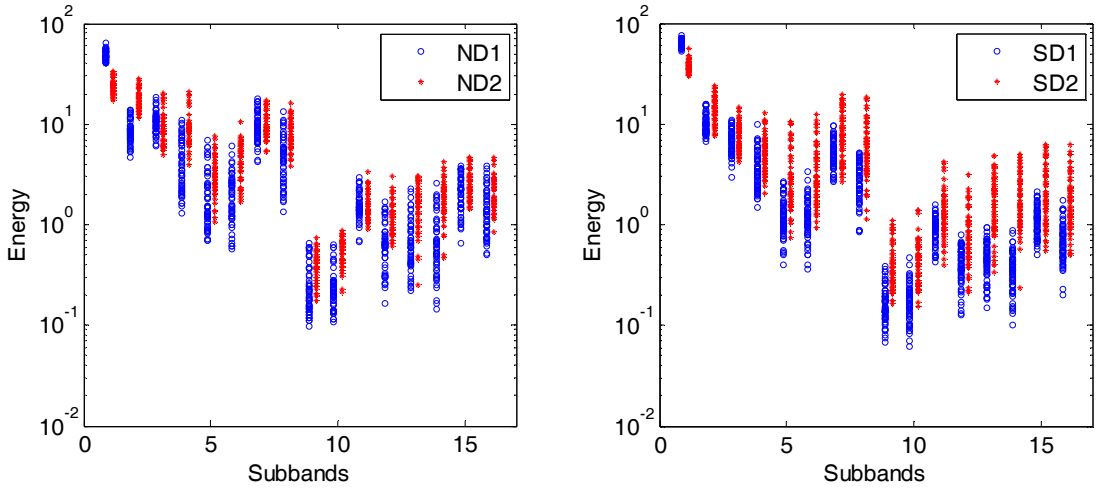
4.2.3 Performance of the Wavelet Packet Tree

This section evaluates the performance of the wavelet packet decomposition. For simplicity, the HAAR wavelet is used. The shapes of the scale and wavelet functions for the HAAR wavelet are shown in Figure 4.11. The signals are decomposed to a level of 4 and the energy of 16 sub-bands are extracted as features. Figure 4.12 shows the repeatability of the features based on level 4 HAAR wavelet packet decomposition. Similar to sub-band energy, data points obtained on different days forms two clusters indicating good repeatability on the same day but poor repeatability between different days. The separability (shown in Figure 4.13) is very poor for energy of all sub-bands, indicating that the sub-band energy of level-4 HAAR wavelet packet decomposition is not a good option for differentiating signals from delaminated concrete and those from non-delaminated concrete.



(a) Scaling Function (b) Wavelet Function

Figure 4.11 AAR Wavelet



(a) ND1 vs. ND2 (b) SD1 vs. SD2

Figure 4.12 Repeatability of the WP Tree

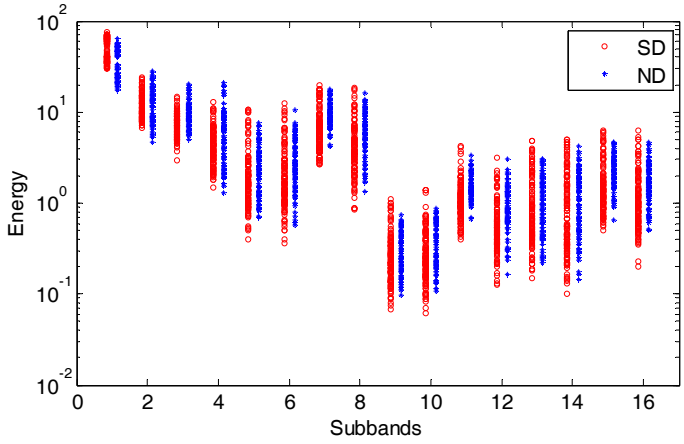


Figure 4.13 Separability of the WP Tree

The numerical performance measures are plotted in Figure 4.14 to Figure 4.16 for comparison. Similar to sub-band energy, the repeatability of different branches is different. The SEP values of branches 9-16 are much higher than the first eight branches, but the value of the mutual information does not show this pattern, indicating that there is inconsistency between the mutual information and separability measures.

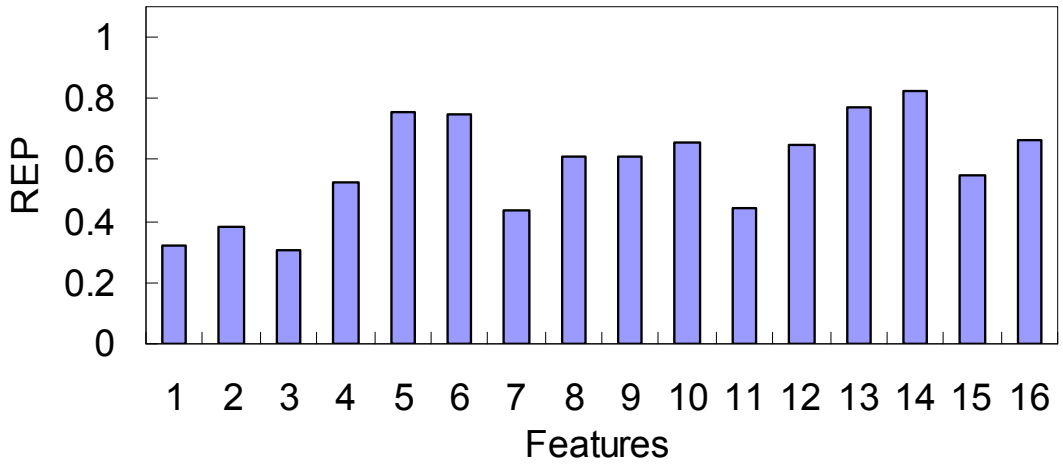


Figure 4.14 REP of the Wavelet Packet Tree

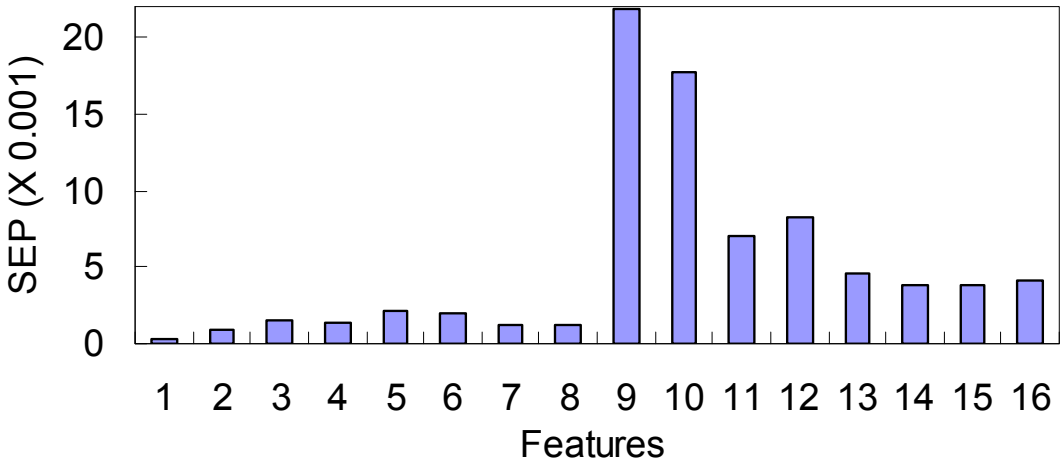


Figure 4.15 SEP of the Wavelet Packet Tree

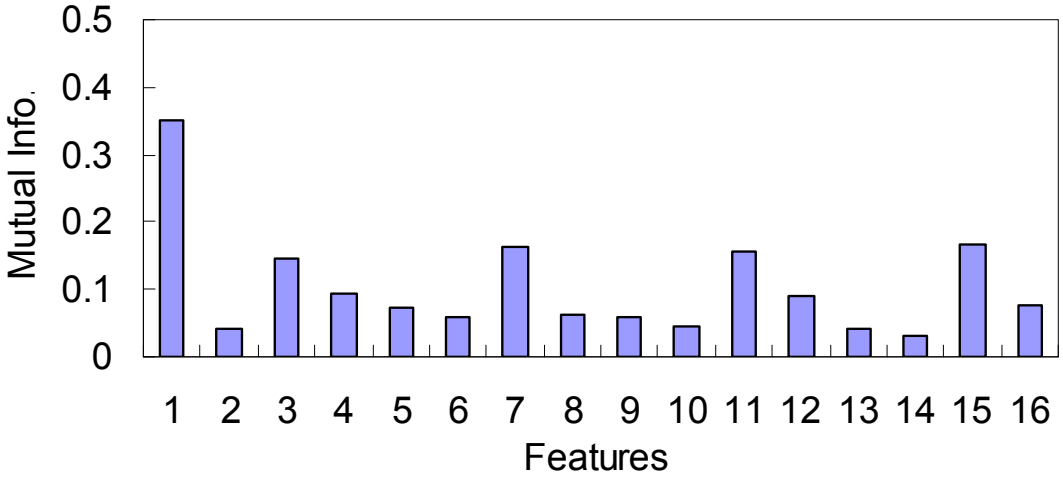
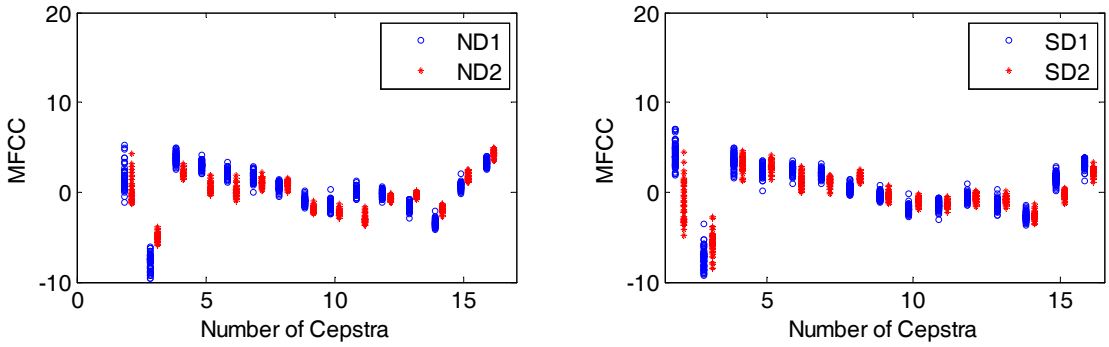


Figure 4.16 Mutual Information of the Wavelet Packet Tree

4.2.4 Performance of MFCC

In calculating the MFCC, a filter bank consisting of 50 triangular filters that are evenly spaced on Mel-scale is used. A total of 16 cepstra coefficients were computed. The repeatability of MFCC is shown in Figure 4.17. The variation of MFCCs is small across different days indicating good repeatability. The separability of the MFCC between the solid and delaminated concrete is shown in Figure 4.18. The difference between the solid concrete (ND) and delaminated concrete (SD) is small.



(a) ND1 vs. ND2 (b) SD1 vs. SD2

Figure 4.17 Repeatability of the MFCC

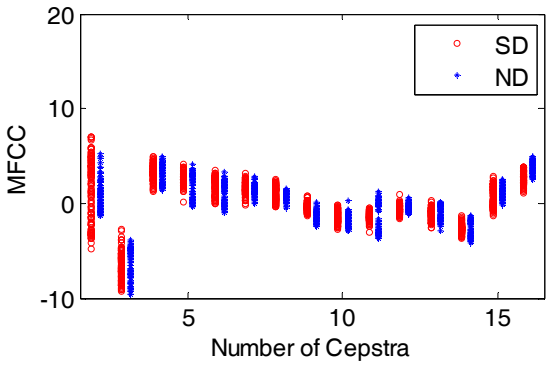


Figure 4.18 Separability of the MFCC

The numerical performance criteria for different coefficients are shown in Figure 4.19 to Figure 4.21. The high REP values of the second and fifteenth MFCC result from the high variation between different days, meaning that these features are not stable and may not be good choices for detection purposes. Inconsistency between SEP and mutual information measures also occurs for several features.

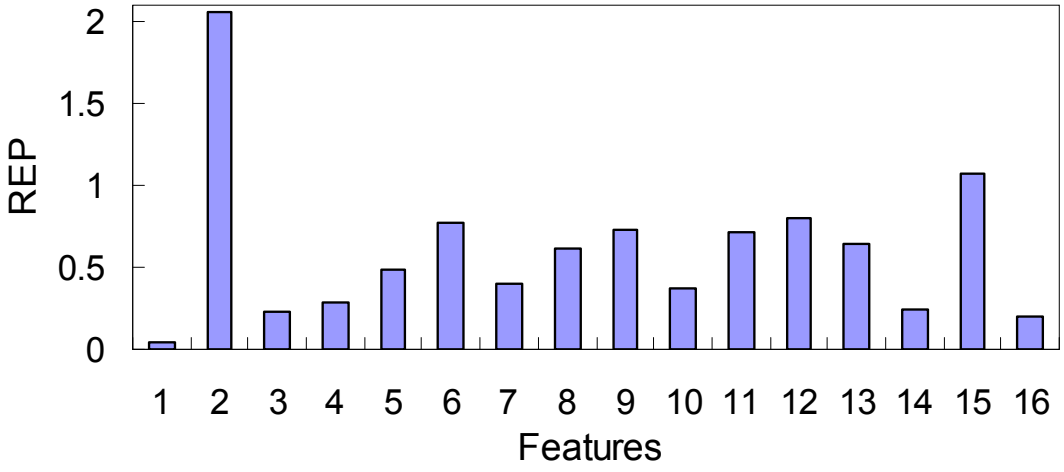


Figure 4.19 REP of the MFCC

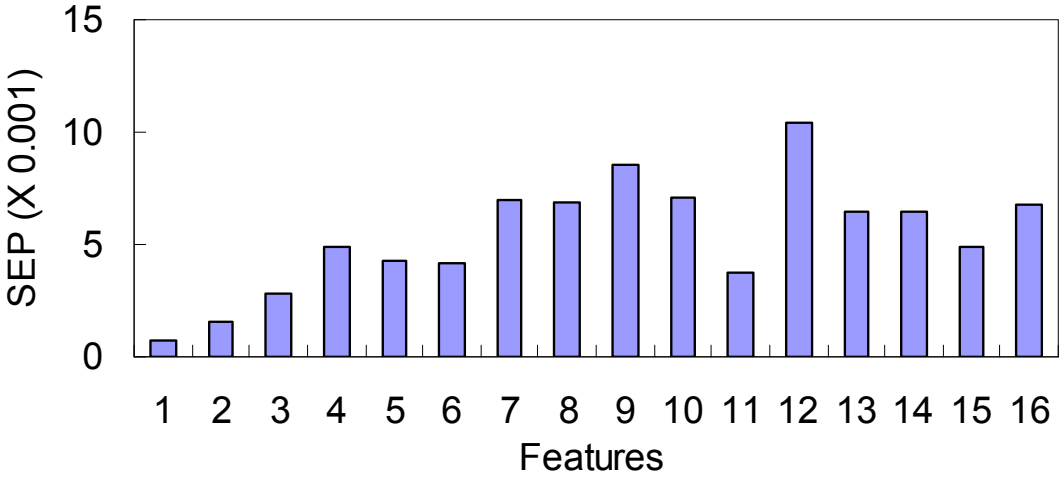


Figure 4.20 SEP of the MFCC

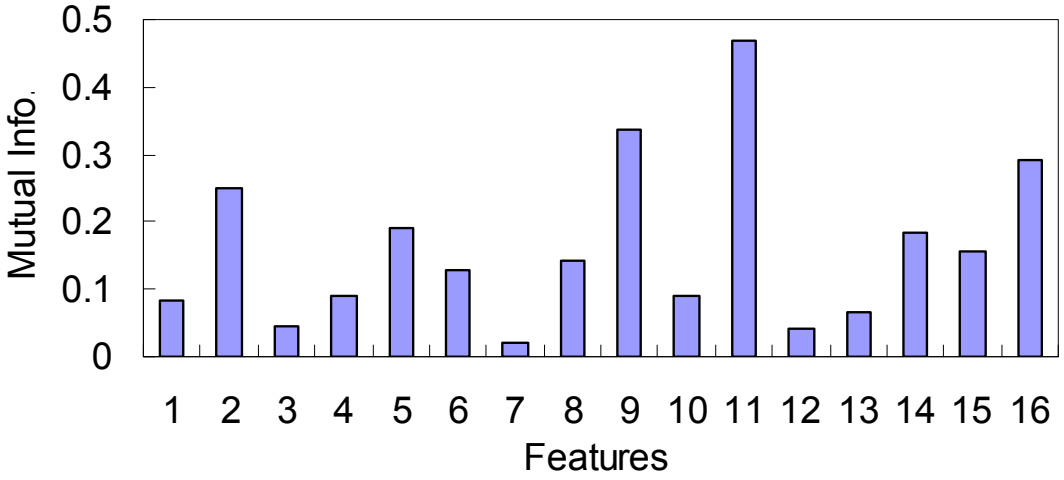
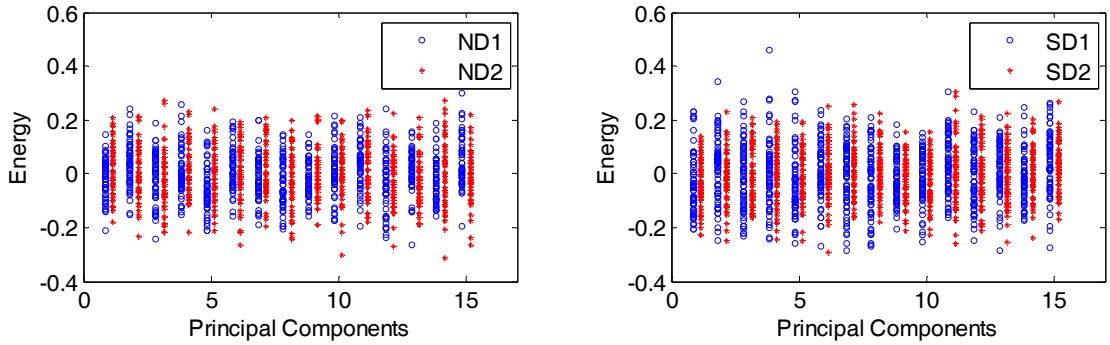


Figure 4.21 Mutual Information of the MFCC

4.2.5 Performance of Features Extracted by PCA

To check the performance of PCA, the first 16 dominant principal components of the signal were extracted as features. Figure 4.22 shows that the repeatability of the features is very good. However, the extracted features of the delaminated and solid signals have a large overlap as shown in Figure 4.23.



(a) ND1 vs. ND2 (b) SD1 vs. SD2

Figure 4.22 Repeatability of the PCA

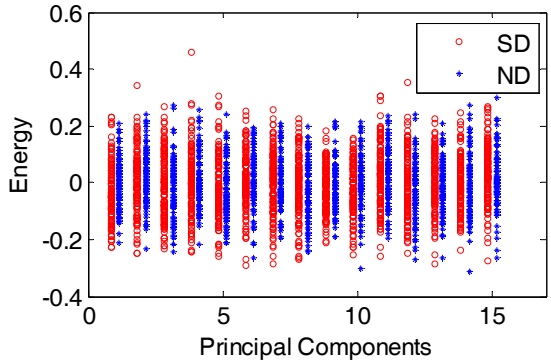


Figure 4.23 Separability of the PCA

The repeatability, separability and mutual information measures of the features extracted by PCA are shown in Figure 4.24 to Figure 4.26. The REP values of several features were too high and were not shown in Figure 4.24. Several conclusions can be observed here. The good repeatability indicated in the Figure 4.22 is not supported by the numerical repeatability measure. The reason for this is that the absolute values of these features are close to zero and Equation(4.15) becomes ill-conditioned: a small variance may lead to a high repeatability measure. The separability of PCA is poor according to Figure 4.23, but the values of SEP are very high. This also results from the ill-conditioning problem. The absolute value of the variance for features extracted by PCA is small which makes Equation (4.16) become ill-conditioned. The values of mutual

information of most principal components are very small indicating that these principal components do not contain much information about their class labels. The results are consistent with Figure 4.23. The small value of mutual information may result from the high dimension of the original signal. The number of principal components is only sixteen in this case, which is very small compared with the high dimension of the acoustic signal. Therefore the information contained in the extracted features is not enough to represent the original signal and differentiate between signals from the solid concrete and ones from delaminated concrete.

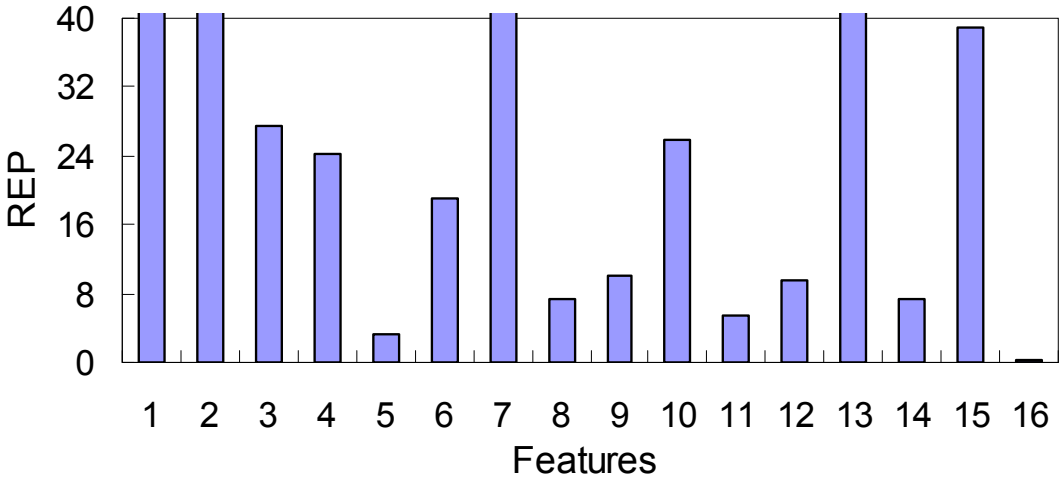


Figure 4.24 REP of the PCA

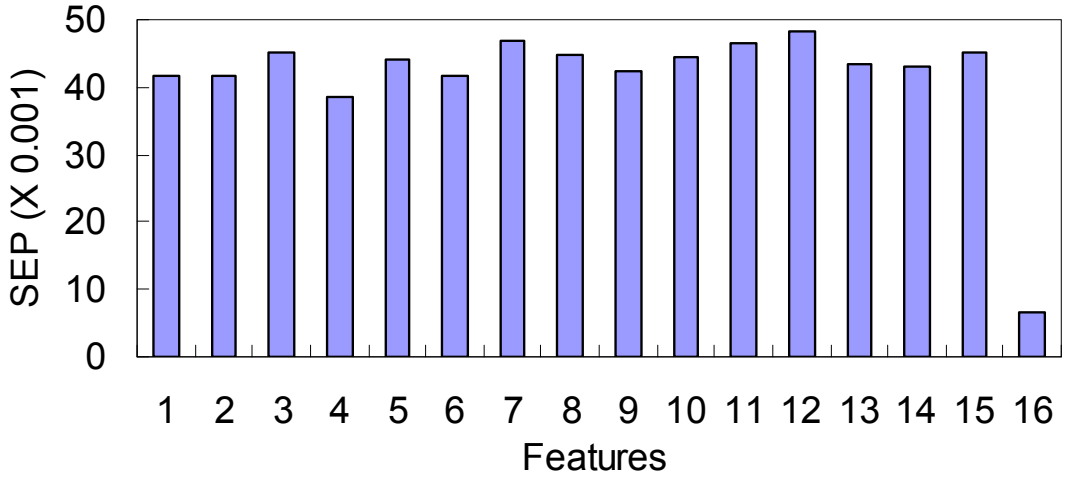


Figure 4.25 SEP of the PCA

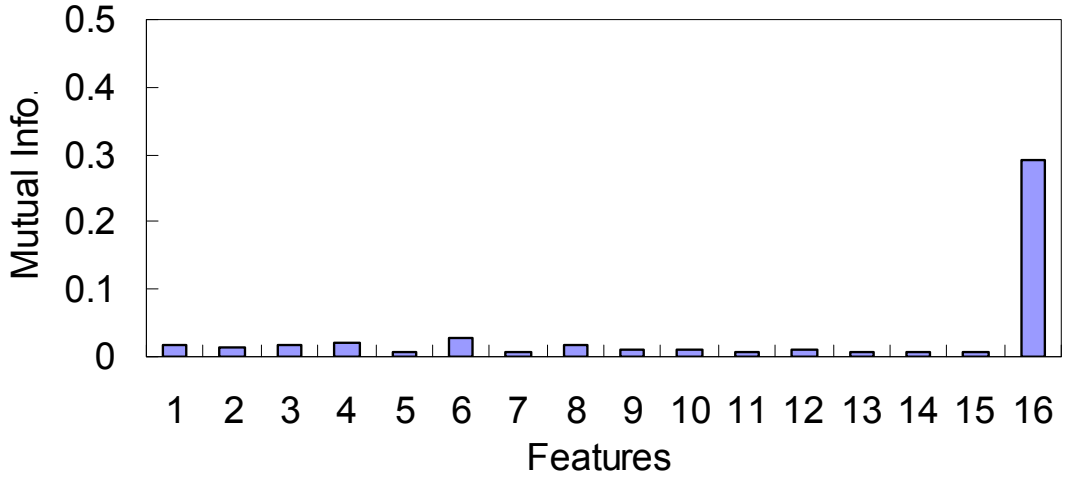
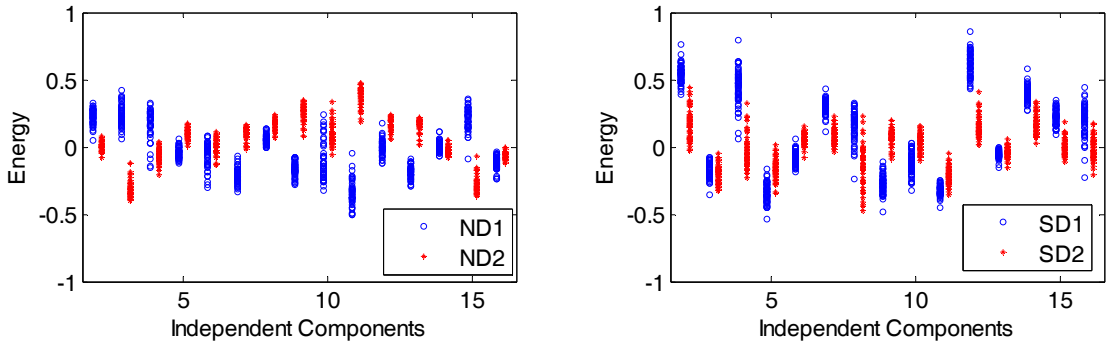


Figure 4.26 Mutual Information of the PCA

4.2.6 Performance of Features Extracted by ICA

The features extracted by the ICA-based algorithm described in Section 4.1.5 is discussed next. 16 cepstral coefficients were extracted by using 25 filter banks determined by ICA. Figure 4.27 depicts the repeatability of the features. As can be seen, the features are not repeatable between days but repeatable within a day. Figure 4.28 shows the difference between the signals from delaminated signals and those from solid

concrete. The separation between the two types of signals is not clear. The reason for this is that the independent components of the input signal are computed by maximizing the independence of the output signals. In this process, no information about the concrete type or class label is included, therefore features extracted by this method is indiscriminant.



(a) ND1 vs. ND2 (b) SD1 vs. SD2

Figure 4.27 Repeatability of the ICA

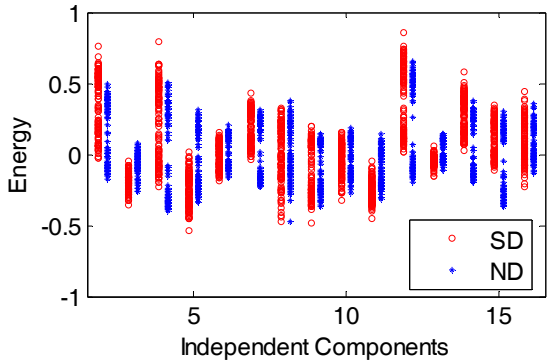


Figure 4.28 Separability of the ICA

Figure 4.29 to Figure 4.31 show the numerical measures for repeatability, separability and mutual information. The repeatability for most features is poor compared with features previously discussed. The high separability measure result from the ill-conditioned problem mentioned in the previous section and values of very high SEP features are not shown in Figure 4.30. The separability measure is not consistent with the

results shown in Figure 4.28. The mutual information values indicate that ICA may have a better performance than PCA since ICA features have relative high mutual information values.

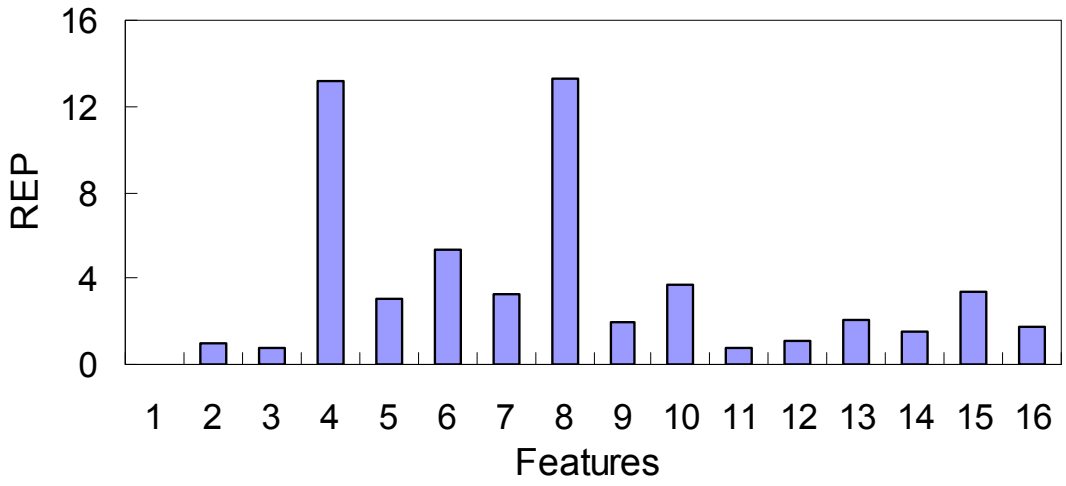


Figure 4.29 REP of the ICA

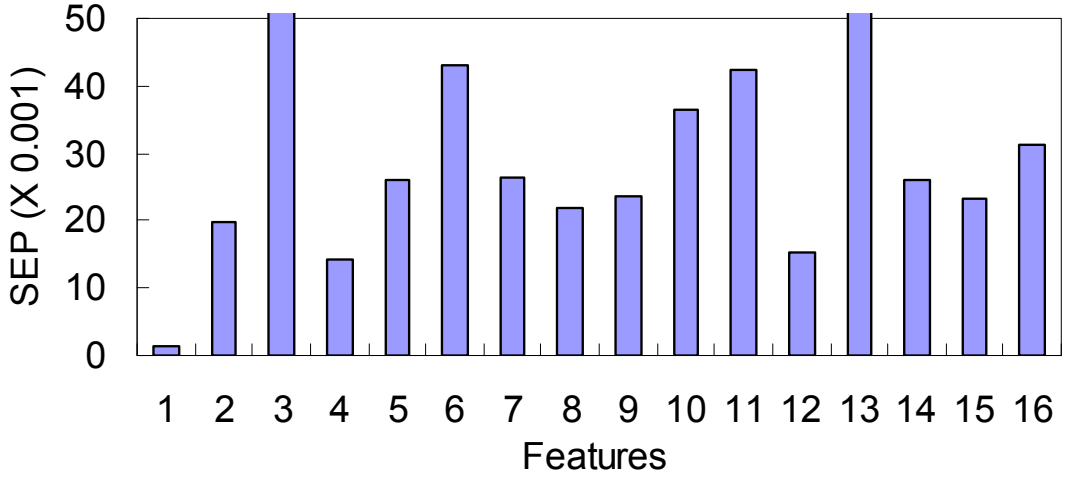


Figure 4.30 SEP of the ICA

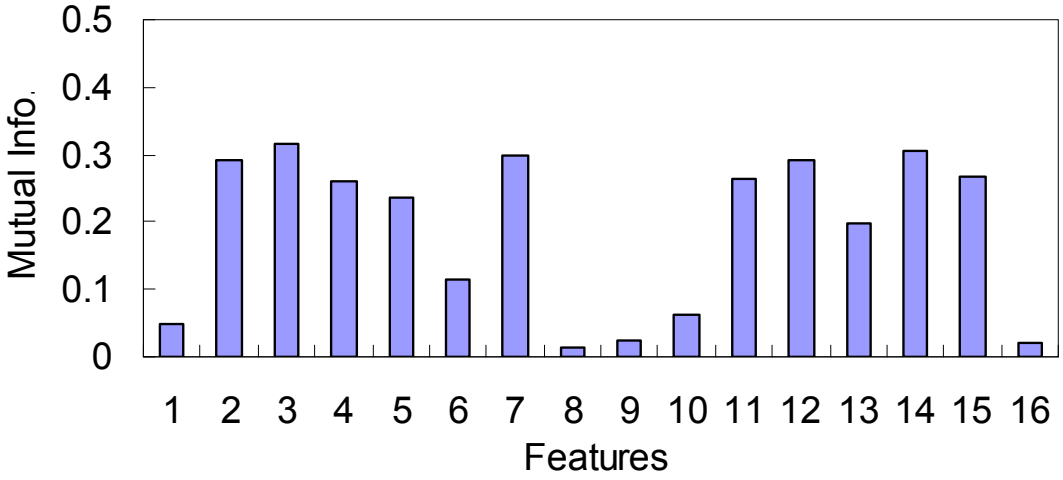


Figure 4.31 Mutual Information of the ICA

4.2.7 Summary of the Section

The performance of features extracted by different algorithms was evaluated using different evaluation criteria. Features extracted by different algorithm had different performances in terms of REP, SEP and mutual information and it is difficult to select a good feature extraction algorithm that maximizes the performance. Different features extracted by the same method also show different performances. Features that are useful for delamination detection should be retained and others eliminated. Repeatability and separability measures can have an ill-conditioning problem if the values of the features are close to zero. Mutual information is a more consistent measure across difference features.

4.3 Selection of the Feature Extraction Algorithm

The previous section listed and compared the performance of different feature extraction algorithms. Different algorithms performed differently when measured using

different criteria. This section describes how the feature extraction algorithm for the delamination detection was selected. In the first section, applicant algorithms were evaluated based on their ranking in terms of repeatability, separability and mutual information. In the second section, the performance of feature extraction algorithms was tested using the data mentioned in Section 4.2.1and the algorithms were compared and selected based on the rate of misclassification.

4.3.1 Algorithm Selection Based on Weighted Rank

Performance of different algorithms was calculated from the numerical performance: REP, SEP and mutual information. Since different features extracted by the same algorithm performed differently, it was necessary to select features that had the best performance. Based on the research on several commonly used classifiers described in Chapter 5, the performance reached or became close to the optimal performance when the number of features was four. The performance of feature extraction algorithms were calculated from four features that had the best performance for each criterion. For example, when comparing different algorithms in terms of repeatability, the four features with the lowest REP will be selected and the mean REP value of the selected features was calculated and considered as the repeatability measure of this algorithm. After the performance of each algorithm was calculated, they were ranked from good to poor according to the numerical performance measure. The algorithm with the lowest rank was selected as being the best. Figure 4.32 to Figure 4.34 show the performance of different algorithms. The SEP values of PCA and ICA were too high and cannot be completely shown in Figure 4.33.

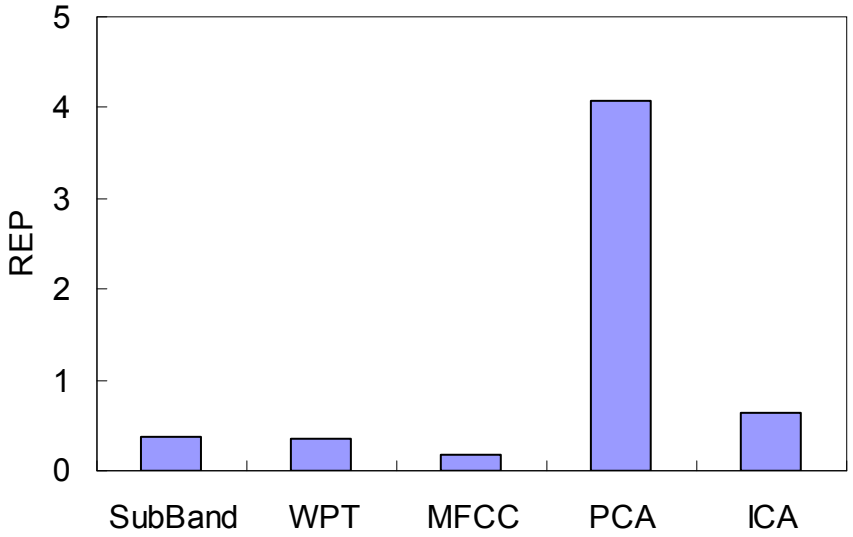


Figure 4.32 REP of different algorithms

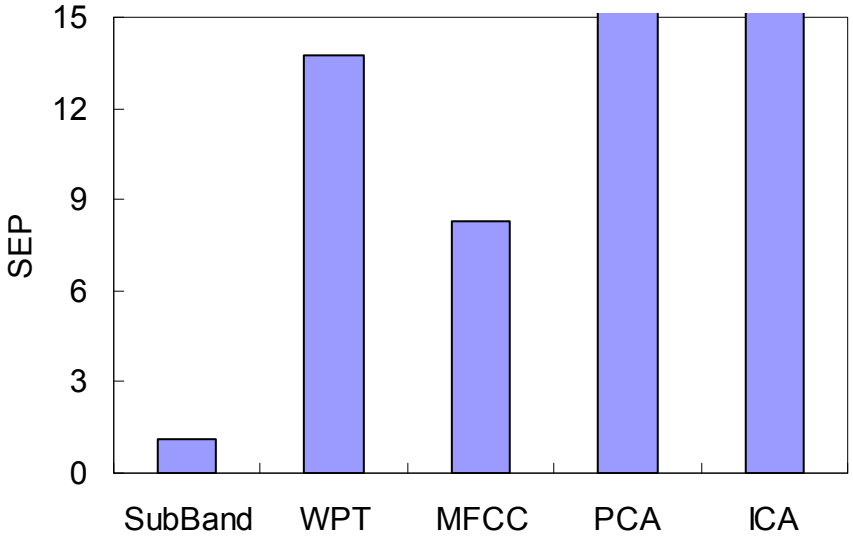


Figure 4.33 SEP of different algorithms

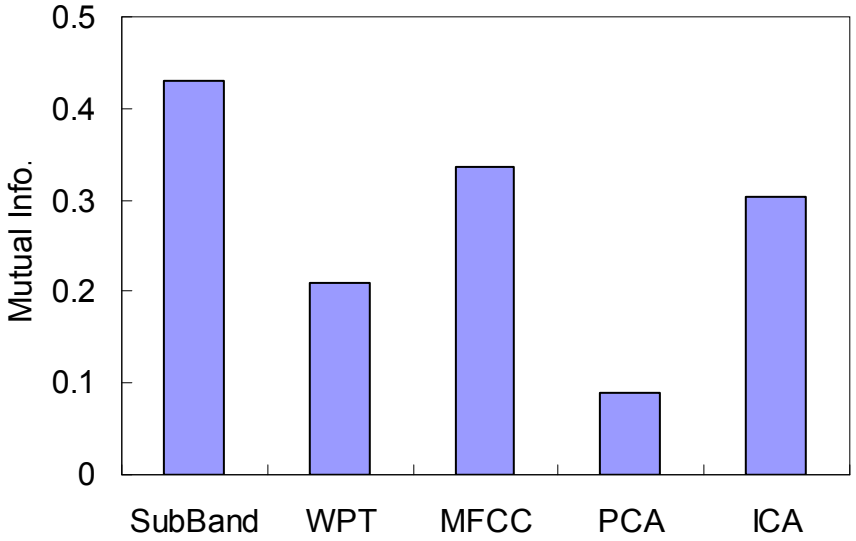


Figure 4.34 Mutual Information of different algorithms

Clearly, different algorithm performed differently under different criteria. MFCC had the best performance in terms of repeatability, while sub-band energy has the best performance in terms of separability and mutual information. To take this into account, a weighted rank was used. The weights assigned to REP, SEP and mutual information were 0.5, 0.25 and 0.25, respectively. The weights were assigned based on the following considerations: Repeatability and separability are equally important in the selection of the algorithm and were assigned equal weight of 0.25. SEP and mutual information are essentially two different ways of measuring of separability, therefore the weight for separability was equally distributed between SEP and mutual information. The rank of each individual criterion and the weighted rank of different algorithms are summarized in Table 4.1. It can be seen that MFCC has the best overall performance. The results were further confirmed using experimental data in the next section.

Table 4.1 Rank of Different Feature Extraction Algorithms

	Sub-band	**WPT**	**MFCC**	**PCA**	**ICA**
REP	2	3	1	5*	4
SEP	5	3	4	2*	1*
Mutual Info.	1	4	2	5	3
Weighted Rank	2.5	3.25	**2**	4.25	3

where * means the performance index has the ill-conditioning problem.

4.3.2 Algorithm Selection Based on Error Rates

The purpose of comparing different feature extraction algorithm is to find the algorithm that has the best accuracy in the detection of concrete delamnation. In this section, the performance of the feature extraction algorithms was measured using the error rate when classifying experimental data using a simple Bayesian classifier.

The first step was to select the features that are useful for delamination detection because "unwanted" features will decrease the accuracy of the detection and increase the computational load for classification. The mutual information was used to select "useful" features in this study. Based on the results from Chapter 5, the optimal number of features is four. The first four features with the highest mutual information values were selected as useful features and were used to train the classifier. The trained classifier was then used to classify different types of data and the error rate (defined below) of different algorithm was used as the criterion to evaluate the performance of the candidate feature extraction algorithms for the damage detection.

The data mentioned in Section 4.2.1was used. From each group (ND1, ND2, SD1 and SD2), 10 impact signals were randomly selected as the training signals. The remaining signals were used as testing signals. The features of these 40 signals were extracted and the mutual information was calculated based on the training signals. The four features with the highest mutual information were selected as the effective feature for detection. The features of the testing signals were also extracted and then selected based on the effective features obtained in the training step. The effective features of the training signals were used to train a linear Bayesian classifier. Detailed information about the classifier will be described in Chapter 5. The trained classifier was then used to classify the testing signals into two groups: solid or delaminated. The number of misclassifications was recorded and the error rate was computed as the ratio of the number of misclassification and the number of testing signals. Due to the variance of the signals, the effective features selected based on the mutual information may change with the selection of the training signals and the error rate may change accordingly. Considering this, the average error rate of 100 runs was used to compare the performance between different feature extraction algorithms. The performance of different algorithms is summarized in Table 4.2.

Table 4.2 Error Rate of Different Feature Extraction Algorithms

Algorithm	Sub-band Energy	Energy of WP Trees	MFCC	PCA	ICA
Error Rate (%)	15.79	30.38	9.23	49.38	19.01

It can be observed that the weighted ranking in Table 4.1 agree well with the ranking of error rate in Table 4.2, indicating that the repeatability and mutual information are both important in the selection of feature extraction algorithms. It can also be seen that MFCC

has the lowest weighted rank and the smallest error rate. MFCC is therefore selected as the feature extraction algorithm to detect delamination in concrete bridge decks.

4.4 Summary

This chapter focused on the problem of how to extract different features of the impact signals and how to select the best feature extraction algorithm for the purposes of delamination detection was selected.

Five commonly used feature extraction algorithms were introduced. The sub-band energy extracts the distribution of the energy of the signal in different sub-bands; this is the same as finding the energy of the signal filtered through a series of rectangular filters in the frequency domain. Energy in the wavelet packet tree is equivalent to passing the signal through filter banks determined by the wavelet function at different levels and computing the energy of the outputs. MFCC is the spectrum of the log of the power spectrum. The power spectrum is computed by filtering the signal through a series of triangular filters in the frequency domain whose centers are evenly spaced on the Mel-scale. PCA reduces the dimension by finding a linear transformation of the signal such that the mean square error between the signal with reduced dimension and the original signal is minimized. This keeps the features that contain most of the energy of the signal but the information is not necessarily effective in separating the delaminated signals from the solid signals. ICA find the “basis” of the signals in a statistical sense such that most of the “information” about the original signal is retained. The spectrum of the basis found by ICA is used to replace the triangular filters in MFCC. The cepstral coefficients of the filter bank output is computed and used as the features.

The performance of the feature extraction algorithms were compared and evaluated against various criteria including repeatability, separability and mutual information between the class labels and the features. Different features from the same feature extraction algorithm performed differently since not all the features contain information that differentiate between solid and delaminated concrete. Also, performance measures of repeatability and separability can be ill-conditioned in certain cases. Mutual information between extracted features and class labels are more consistent across all the algorithms.

The effectiveness of different feature extraction algorithm was evaluated using a weighted rank and the error rate of the classification. It was found that both repeatability and separability are important in the evaluation of the feature extraction algorithm and that MFCC has the best overall performance. MFCC was therefore selected as the feature extraction algorithm for use in this study.

(This page intentionally left blank.)

CHAPTER 5

PATTERN RECOGNITION AND DELAMINATION DETECTION

In Chapter 4, different feature extraction algorithms were used to find the optimal features for delamination detection. Mel-frequency Cepstral Coefficient (MFCC) was found to be most efficient. Once features of the acoustic signal are extracted and selected, the next task in the delamination detection is to differentiate the signals recorded on the solid concrete from those recorded on delaminated concrete. This problem can be formulated as a classification problem and the task is to classify the recorded signal into two groups: signals from solid concrete and signals from delaminated concrete. There are infinite numbers of ways of drawing a dividing line between the two groups. Rather than drawing the boundary empirically "by eye", the line should be drawn optimally with respect to certain criteria. Different classification algorithms optimize different criteria. It is therefore necessary to compare and evaluate different algorithms to select the classifier that has the best performance for delamination detection.

As discussed in the previous chapter, the repeatability and separability of the features extracted from impact signals are not good, which makes the classification task complicated. The classifier needs to accommodate the variance between different tests and to separate signals from different types of concrete. In this chapter, four commonly used classifiers are compared and evaluated. These are Bayesian classifier, support vector

machine (SVM), multi-layer perceptron (MLP), and radial basis function (RBF) network. After comparing the performance of these classifiers, the best classifier is selected.

5.1 Detection Algorithms

This section briefly describes the theoretical background of the four classifiers mentioned earlier. The classification is essentially an optimization problem: the classifier tries to minimize or maximize the cost function depending on certain criteria. The parameters of the classifiers are trained by the data using a training data set and the trained classifiers are then used to classify new data.

5.1.1 Bayesian-Based Classifier

Bayesian-based classifiers are derived from total probability and Bayes rules [51] and the target is to minimize the probability of classification error. For the case of delamination detection, there are only two classes: solid (labeled as C_1) or delaminated (labeled as C_2). Given the observed feature x_i, the classification problem can be formulated as:

$$d_i = \begin{cases} 1, \text{ if } x_i \in C_1 \\ -1, \text{ if } x_i \in C_2 \end{cases} \tag{5.43}$$

For the Bayesian classifier, if the feature is less than a certain threshold, the data points is classified as class 1; if the feature is greater than the threshold, the data is classified as class 2. Based on this, the Bayesian decision rule can be expressed as:

$$d_i = \begin{cases} 1, \text{ if } x_i < x_0 \\ -1, \text{ if } x_i > x_0 \end{cases} \tag{5.44}$$

where x_0 is the threshold of the Bayesian classifier, as shown in Figure 5.1.

The goal here is to find an optimal x_0 such that the probability of misclassification (P_e) is minimized. Assuming that a priori probability of C_1 and C_2 are the same, the P_e can be derived as:

$$\begin{aligned} P_e &= P(C_1 \mid x \in C_2) + P(C_2 \mid x \in C_1) \\ &= P(C_1)\left[\int_{-\infty}^{x_0} P(x \mid C_2)\,dx + \int_{x_0}^{\infty} P(x \mid C_1)\,dx\right] \end{aligned} \tag{5.45}$$

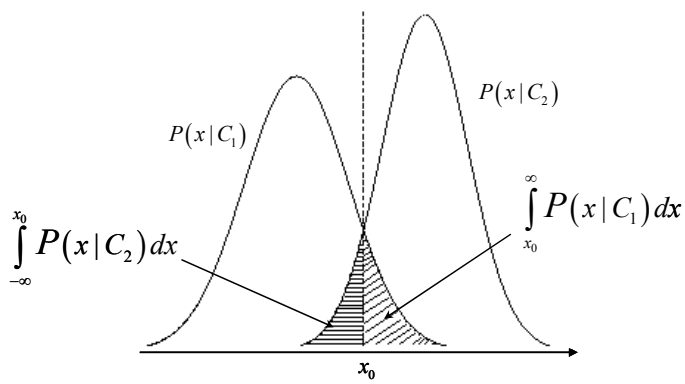


Figure 5.1 Threshold of Bayesian Classifiers

$P(C_1 \mid x \in C_2)$ can be expressed by the area on the left of the threshold under the curve of $P(x \mid C_2)$ in Figure 5.1 and similarly, $P(C_2 \mid x \in C_1)$ is the area on the right of the threshold under the curve of $P(x \mid C_1)$. The probability of misclassification is then proportional to the shaded area shown in Figure 5.1. From the figure, it is obvious that when the threshold is at the intersection of the two curves, the shaded area is minimized and therefore, the misclassification probability is minimized. Therefore, the Bayesian rule can be expressed as:

$$d = \begin{cases} 1, \text{ if } P(x \mid C_1) > P(x \mid C_2) \\ -1, \text{ if } P(x \mid C_1) < P(x \mid C_2) \end{cases} \tag{5.46}$$

The decision surface of the Bayesian classifier can be expressed as the zeros of the following equation:

$$g(x) = P(x \mid C_1) - P(x \mid C_2) = 0 \tag{5.47}$$

The most commonly encountered probability density function in practice is the normal (Gaussian) distribution. In this case, the expression of the threshold can be further simplified.

The conditional probability density function of a jointly normal vector x can be expressed as:

$$P(x \mid C_i) = \frac{1}{(2\pi)^{l/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right) \tag{5.48}$$

where Σ_i is the covariance matrix of each class and μ_i is the mean value.

The decision surface for class i can be expressed as:

$$g_i(x) = -\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) - \frac{l}{2} \ln(2\pi) - \frac{1}{2} \ln|\Sigma_i| \tag{5.49}$$

For un-correlated x, the covariance matrix is diagonal and the decision surface becomes a quadratic function. Further, if the variance of all elements of x are equal, the decision surface reduced to be a hyper-plane, and the Bayesian classifier becomes a linear classifier.

From the above description, it can be seen that the decision surface can be found if the underlying distribution is known. In practice, this assumption is not necessarily true and the distribution or the parameters of the underlying distribution needs to be estimated. There are different ways to estimate unknown information such as maximum likelihood estimation [52] or expectation maximization algorithm [53].

5.1.2 Support Vector Machine

The Bayesian classifier described in the previous section tries to find the decision surface by minimizing the probability of misclassification. However, it requires prior information about the underlying distribution. Even though this information can be obtained through hypothesis testing or parameter estimation, the performance depends on how well the information about the underlying distribution is estimated. Linear classifiers which are not dependent on the underlying distribution of the training data provide one solution to this problem [54]. Linear classifiers try to classify data into different groups by a hyper-plane. As can be seen from Figure 5.2, there are infinite numbers of hyper-planes that can separate the two classes. The problem then is to find the optimal hyper-plane. One commonly used decision surface is the hyper-plane that maximizes the margin of separation as shown in Figure 5.2.

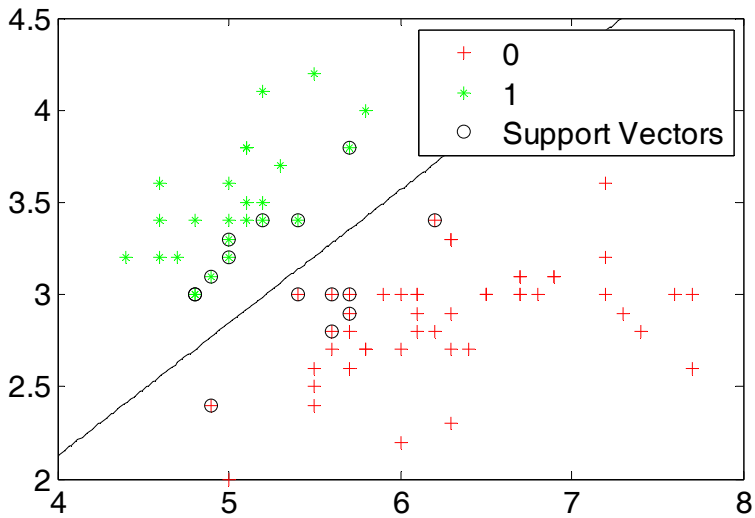


Figure 5.2 Support Vector Machine

For a linear classifier, given data point x_i, the decision can be expressed as:

$$d_i = \begin{cases} 1, \text{ if } w^T x_i + b > 0 \\ -1, \text{ if } w^T x_i + b < 0 \end{cases} \tag{5.50}$$

where, $g(x) = w^T x + b$ is the decision surface (hyper-plane in this case) and, w and b are the weighting and bias vectors, respectively.

The distance of the data point x_i to the decision surface can then be written as:

$$r = \frac{g(x_i)}{\|w\|} \tag{5.51}$$

If the training data is linearly separable, it is always possible to find a hyper-plane that satisfies Equation (5.52) below by scaling the weighting and bias vectors.

$$g(x_i) = \begin{cases} w^T x_i + b \geq 1, \forall x \in C_1 \\ w^T x_i + b \leq -1, \forall x \in C_2 \end{cases} \tag{5.52}$$

The equality is satisfied for the points that are closest to the decision plane. These points are called support vectors.

The margin between the two classes can then be expressed by the distance between the support vectors and the decision plan as:

$$r = \frac{g\left(x_{s1}\right) + g\left(x_{s2}\right)}{\|w\|} = \frac{2}{\|w\|} \tag{5.53}$$

where, x_{s1} and x_{s2} are the support vectors in two classes.

From Equation (5.53), it can be seen that maximizing the margin between the two classes is equivalent to minimizing the norm of the weight vectors under the constraint of Equation (5.52). In fact, Equation (5.52) can be ombined with Equation (5.50) to yield:

$$d_i\left(w^T x_i + b\right) \geq 1 \tag{5.54}$$

This optimization problem can be solved using Lagrange multipliers. The Lagrangian function can be constructed as:

$$J\left(w, b, \lambda_i\right) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \lambda_i \left[d_i \left(w^T x_i + b \right) - 1 \right] \tag{5.55}$$

where, λ_i are the Lagrange multipliers and N is the number of training data sets.

The Lagrangian function in Equation (5.55) becomes stationary at the optimal solution and therefore:

$$\frac{\partial J}{\partial w} = w - \sum_{i=1}^{N} \lambda_i d_i x_i = 0 \tag{5.56}$$

$$\frac{\partial J}{\partial b} = \sum_{i=1}^{N} \lambda_i d_i = 0 \tag{5.57}$$

Substitute (5.54) and (5.56) into (5.55) and by using the Kuhn-Tucker conditions [55], the optimization of the Lagrangian function is equivalent to minimizing:

$$J(\lambda_i) = \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \lambda_i \lambda_j d_i d_j x_i^T x_j \tag{5.58}$$

under the constraints of (5.57) and non-negative Lagrange multipliers λ_i .From the training data, the optimal Lagrange multipliers λ_i can be obtained. Substituting this into Equations (5.56) and (5.52), the optimal hyper-plane can be uniquely determined.

If the classes are not linearly separable, it is always possible to construct a non-linear transformation that transforms the input features to a higher dimensional space in which the two classes becomes linear separable. Such a transformation function is called a kernel function. Different kernel functions may have different effect on the performance of the classifier. In this study, the performance of a quadratic kernel function and a linear kernel function are compared.

5.1.3 Multi-Layer Perceptron

The SVM described in the previous section is a linear classifier and linearly non-separable data needs to be transformed into a higher dimensional space through a non-linear transformation to obtain linearly separable data. A non-linear classifier, on the other hand, does not require such a transformation to separate linearly non-separable data. The multi-layer perceptron (MLP) [35] is such non-linear transform.

A multi-layer perceptron model is a network of several layers of neurons, as shown in Figure 5.3. It consists of three parts an input layer, hidden layers (may be more than

one layer), and an output layer. The input layer consists of a series of neurons that receive the input data, in this case, the extracted features of the impact signals. The hidden layer consists of several layers of neurons, which take the outputs of the previous layer as inputs and compute the output and feed it to neurons in the next layer. The output layer receives the outputs from the hidden layer and computes the final output.

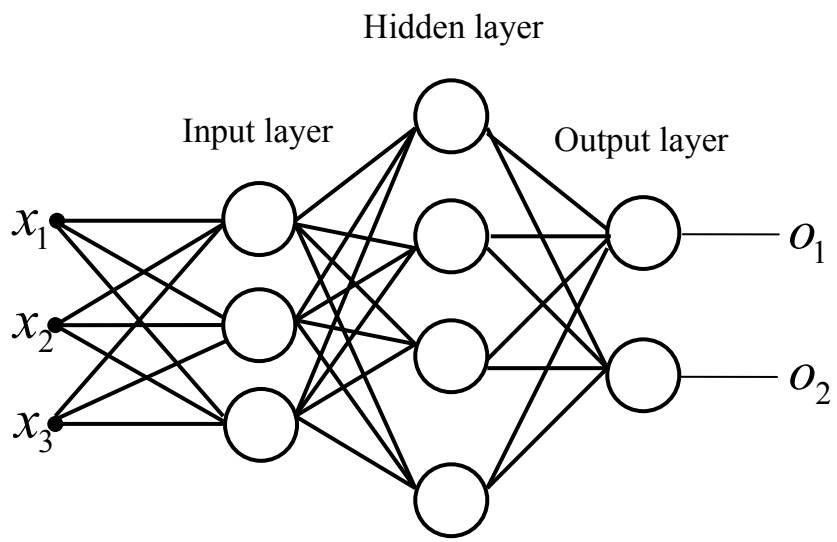


Figure 5.3 Multi-Layer Perceptron

Each neuron in the system is a computation unit. The neuron can compute the weighted sum of the inputs and the summation is fed into an activation function that produces the output. Figure 5.4 shows the signal flow of a perceptron in the output layer connected with a perceptron in the hidden layer. The difference between a perceptron in the output layer and perceptron in other layers is that the output layer perceptron has a desired output. The difference between the actual and desired is called the error signal and is used to update the synaptic weights in the back propagation step described below.

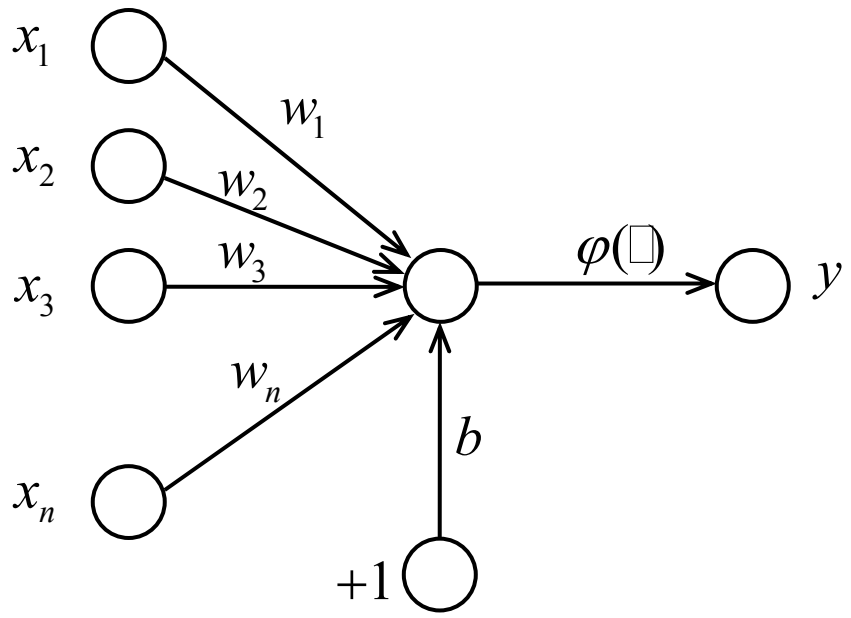


Figure 5.4 Signal-Flow Graph of the Perceptron

The computation consists of two phases. The first phase is called the feed forward process, in which the input is fed into and passed through the system. The output is computed and compared with the desired output to obtain the error signal. The second phase is called back propagation, in which the error signal is used to update the synaptic weighs in the network such that the mean square error between the system output and the desired output is minimized. The gradient decent algorithm is used to find the update law of the synaptic weights.

The feed forward process is simple and straightforward. The problem is to find how the synaptic weight is updated by using the error signals. Assume that the output layer is the kth layer, the mean square error between the actual output and the desired output can be computed as:

$$\varepsilon(n) = \frac{1}{2}\sum_k e_k^2 \tag{5.59}$$

where e_k is the error signal given by:

$$e_k = d_k - y_k \tag{5.60}$$

where d_k is the desired output and y_k is the actual output. The actual output can be computed from:

$$y_k = \varphi(v_k) \tag{5.61}$$

where, $\varphi(\cdot)$ is the activation function of the layer and v_k is the weighted sum of the input from the previous layer, in this case the jth layer.

v_k is calculated through:

$$v_k = \sum_j w_{j,k} y_j \tag{5.62}$$

Using the gradient descent method, the update law of the weight can then be derived as:

$$\begin{aligned} \Delta w &= -\eta \frac{\partial \varepsilon}{\partial w} \\ &= -\eta \frac{\partial \varepsilon}{\partial v}\frac{\partial v}{\partial w} \end{aligned} \tag{5.63}$$

where, η is the factor that controls the learning rate and convergence of the MLP.

From Equation (5.62), it follows that

$$\frac{\partial v}{\partial w_{j,k}} = y_j \tag{5.64}$$

where y_j is the output of the *previous* layer.

For output layer,

$$\frac{\partial \varepsilon}{\partial v_k} = \frac{\partial \varepsilon}{\partial e_k}\frac{\partial e_k}{\partial y_k}\frac{\partial y_k}{\partial v_k} = -\sum_k e_k \varphi_k'\left(v_k\right) \tag{5.65}$$

Therefore, the update law for the output layer is:

$$\Delta w_{j,k} = \eta y_j \sum_k e_k \varphi_k'\left(v_k\right) \tag{5.66}$$

For the layer that is one layer prior to the output layer, say the jth layer,

$$\begin{aligned}\frac{\partial \varepsilon}{\partial v_j} &= \frac{\partial \varepsilon}{\partial v_k}\frac{\partial v_k}{\partial v_j} = \frac{\partial \varepsilon}{\partial v_k}\frac{\partial}{\partial v_j}\left(\sum_j w_{j,k} y_j\right) \\ &= \frac{\partial \varepsilon}{\partial v_k}\sum_j w_{j,k}\frac{\partial y_j}{\partial v_j} \\ &= \frac{\partial \varepsilon}{\partial v_k}\sum_j w_{j,k}\varphi_j'\left(v_j\right)\end{aligned} \tag{5.67}$$

where $\frac{\partial \varepsilon}{\partial v_k}$ is the local gradient in the *next* step.

Therefore, the update for the jth layer is:

$$\Delta w_{j-1,j} = \eta y_{j-1}\frac{\partial \varepsilon}{\partial v_{j+1}}\sum_j w_{j,j+1}\varphi_j'\left(v_j\right) \tag{5.68}$$

The second phase—the back propagation—starts with the output layer using Equation (5.68), and then iteratively using Equation (5.65) the synaptic weights of the entire network can be updated from the output layer to the input layer.

In this research, the MATLAB neural network toolbox was used to design the classifier.

5.1.4 Radial Basis Function

The Radial Basis Function (RBF) network is another way to transform the input into a higher dimensional space for classification purposes. It consists of three layers: the input layer that takes the input and sends it to the hidden layer (there is only one hidden layer in RBF). The hidden layer consists of a certain number of neurons that map the input into a higher dimensional space using radial basis functions. The output of the system is a weighted sum of the hidden layer output. Figure 5.5 shows the architecture of the RBF.

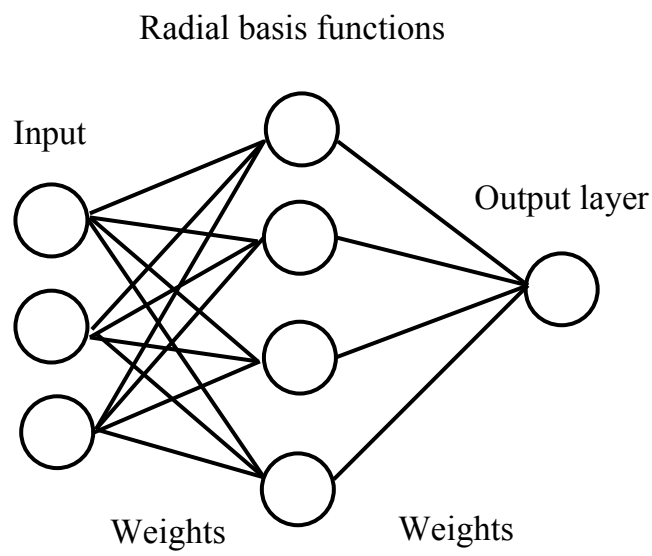


Figure 5.5 Architecture of Radial Basis Function Network

The output of the RBF network can be expressed as:

$$F(x) = \sum_{i=1}^{N} w_i \varphi\left(\left\| x - x_i \right\|\right) \tag{5.69}$$

where $\varphi\left(\left\|x-x_i\right\|\right)$ is a set of functions symmetric about the center x_i, w_i is the weight for each function, and N is the number of basis functions.

In reality, the number of training data may be greater than the number of underlying basis functions, the information provided is over-complete and the problem becomes ill-conditioned. In this case, the results of the RBF network described above may become an "over-fit", meaning that the network works very well for the training data but may not work well for other data.

To solve this problem, regulation theory [56] was proposed. The basic idea is to provide more degrees of freedom to the solution by adding some functions that embed prior information about the solution. The most commonly used function is the linear differential of the solution. This comes from the assumption that the mapping from input to the output is smooth or differentiable. The problem becomes the minimization of the regulated error given by:

$$\varepsilon = \sum_{i=1}^{N}\left[d_i - F\left(x_i\right)\right]^2 + \lambda\left\|DF\right\|^2 \tag{5.70}$$

where d_i is the desired output, λ is a positive real number called the regularization parameter, and D is a linear differential operator.

To solve this optimization problem, the output of the RBF, $\varphi(\cdot)$, is approximated by a family of Green's functions, $G\left(\left\|x-t_i\right\|\right)$, centered at t_i. The Green's functions can be derived from the RBF, $\varphi(\cdot)$. In this way, Equation (5.28) can be reformulated as [35]:

$$\varepsilon = \sum_{i=1}^{N}\left[d_i - \sum_{i=1}^{m1} w_i \underline{G}\left(\left\|x_i - t_i\right\|\right)\right]^2 + \lambda w^T G_0 w \tag{5.71}$$

where,

$$d = \begin{bmatrix} d_1 & d_2 & \dots & d_N \end{bmatrix}^T;$$

$$w = \begin{bmatrix} w_1 & w_2 & \dots & w_{m1} \end{bmatrix}^T$$

$$\underline{G} = \begin{bmatrix} G(x_1,t_1) & G(x_1,t_2) & \dots & G(x_1,t_{m1}) \\ G(x_2,t_1) & G(x_2,t_2) & \dots & G(x_2,t_{m1}) \\ \dots & \dots & \dots & \dots \\ G(x_N,t_1) & G(x_N,t_2) & \dots & G(x_N,t_{m1}) \end{bmatrix}$$

$$G_0 = \begin{bmatrix} G(t_1,t_1) & G(t_1,t_2) & \dots & G(t_1,t_{m1}) \\ G(t_2,t_1) & G(t_2,t_2) & \dots & G(t_2,t_{m1}) \\ \dots & \dots & \dots & \dots \\ G(t_{m1},t_1) & G(t_{m1},t_2) & \dots & G(t_{m1},t_{m1}) \end{bmatrix}$$

Once framed as a minimization problem in Equation (5.71), the unknown parameters in the RBF network can then be updated by an optimization algorithm such as the gradient descent method.

The important parameters in the RBF network are: the number of RBFs (N), the location of the center, the width of the RBF (or the variance, σ), and the synaptic weights that connect the hidden RBF with the output layer. Usually, the number of RBFs and the width of the RBF are selected by the user, while the location of the center and the synaptic weights are optimized by using the training data.

5.2 Performance Evaluation

As described in the previous section, there are different ways to classify the extracted features into different groups. Different algorithms map the input features using different

methods and optimize the different discriminant criteria. This section compares the performance of different algorithms and evaluates the effect of such parameters as the number of features and training samples. After the performance is evaluated, a decision can then be made as to which algorithm should be used for delamination detection.

As mentioned in Chapter 4, Mel-frequency Cepstral Coefficients (MFCC) had the best discriminant capacity and therefore is used as input features to test the performance of the different classifiers. In Chapter 4, the error rate was computed from four extracted features having the highest value of mutual information. In this section, a more thorough evaluation of the number of features is presented. The training samples need to be representative of the entire population. Therefore, 10 training samples (around 20% of the total population) were randomly selected from each group (ND1, ND2, SD1, SD2). The remainder of the signals was used to test the performance of the classifier.

Even though the number of training samples was the same, the training samples were randomly selected. Since the selection of training samples was random, the results could be different. Figure 5.6 shows the error rates of 100 runs. To accommodate the variance due to the random selection of the training samples and to have a more fair comparison between different cases and algorithms, the 95% one-sided confident interval (CL) of the error rates was used.

As discussed in Chapter 4, the number of features plays an important role in the classifier performance of the classifier. The error rates of different numbers of features are compared in this section to find the optimal number of features for the purpose of delamination detection.

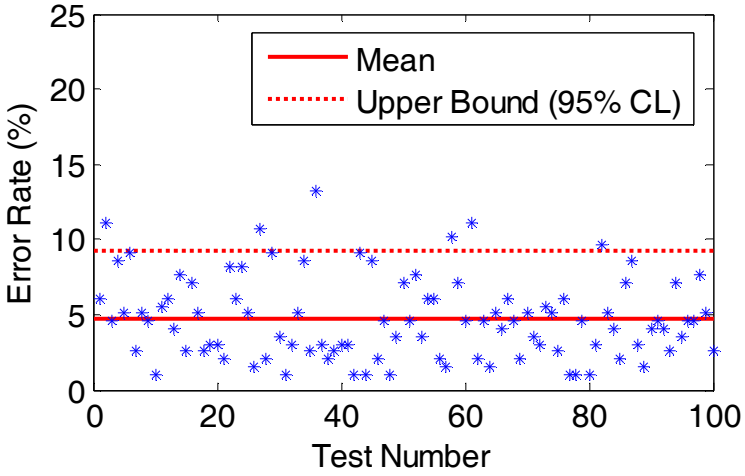


Figure 5.6 Variation of Error Rate due to Random Selection

5.2.2 Performance of Bayesian Classifier

To simplify the computation, several assumptions are made in this section. First, since the prior probability of the delaminated concrete and the solid concrete were unavailable, they were assumed to be 50% for both. Second, the underlying distribution of the extracted features was assumed to be normal. Third, the covariance matrix described in Section 5.1.1 was taken to be diagonal.

In this section, two cases are considered. Case one (Linear Bayesian Classifier) assumes that the diagonal elements of the covariance are equal, meaning that the extracted features are independent and have the same variance. As described in the previous section, the Bayesian classifier in this case becomes a linear classifier. The second case (Quadratic Bayesian Classifier) is more general and only assumes that different features are independent of each other. In this case, the resulting decision surface is a quadratic function.

The error rate of linear Bayesian classifier (Case 1) with different number of features is plotted in Figure 5.7.

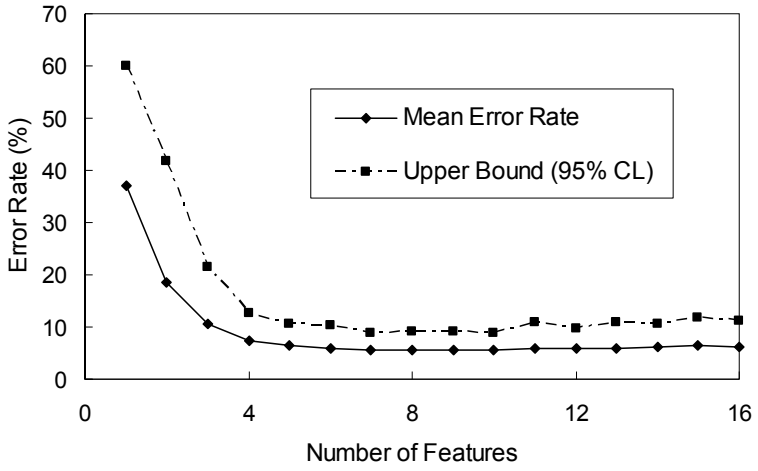


Figure 5.7 Performance of Linear Bayesian Classifier

As the results indicate, the error rate drops with an increase in the number of features, but the increase in the performance is limited. The optimal performance is reached when the number of feature is seven and the optimal performance has a mean errorate of 5.46%, and the upper bound of the 95% CL is 8.85%.

The error rate versus the number of features for the quadratic Bayesian classifier is plotted in Figure 5.8.

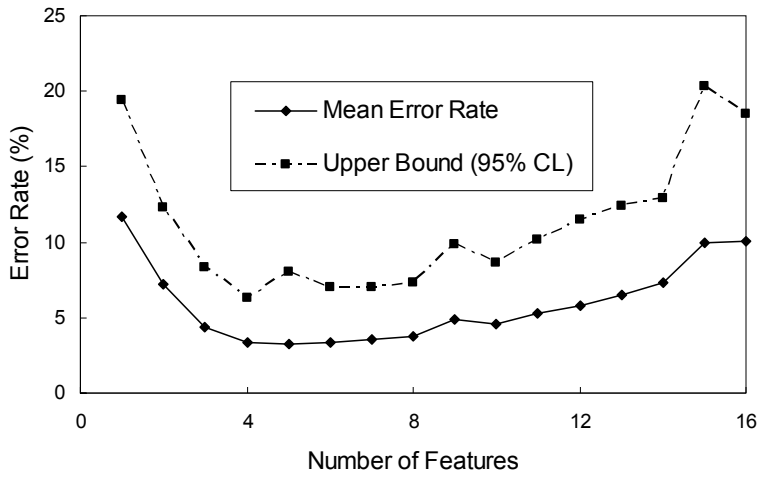


Figure 5.8 Performance of Quadratic Bayesian Classifier

The results show that with an increase in the number of features, the error rate first decreases and then increases. The reason for this is that information contained in the additional features is not sensitive or useful for delamination detection. The different trends for linear and quadratic Bayesian classifiers may come from their properties: the linear classifier is not sensitive to the addition of "noisy" features. The optimal performance for the quadratic Bayesian classifier is achieved when the number of features is six and the error rate is 3.30% with an upper bound of 6.99%, both of which are lower than that of the linear Bayesian classifier.

5.2.3 Performance of Support Vector Machine

As mentioned at the end of Section 5.1.2, the kernel function may influence the performance of the SVM classifier. In this section, two types of kernel functions are used: a linear kernel function and a quadratic kernel function. Similar to the previous section,

the average error rate and the upper 95% CL of 100 trial runs are used as performance indices.

The performance of the linear kernel SVM is shown in Figure 5.9. The error rate decreases with an increase of the number of features. However, the improvement in performance is not significant when the number of features exceeds 5. The optimal performance for linear kernel SVM classifier is reached when the number of features is 12. The optimal error rate is 5.13% and the upper 95% CL is 8.42%.

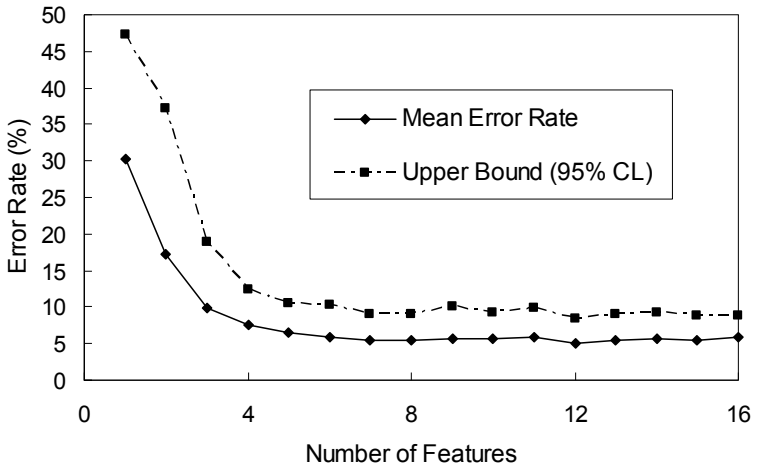


Figure 5.9 Performance of Linear Kernel SVM Classifier

The performance of the quadratic kernel SVM is shown in Figure 5.10. The relationship between the error rate and the number of features is similar to the linear kernel SVM. However, the performance of the quadratic kernel SVM is better than that of the linear kernel. The minimum error rate is only 2.55% and the upper 95% CL is 5.40%. The optimal number of features is 12, which is the same for the linear kernel.

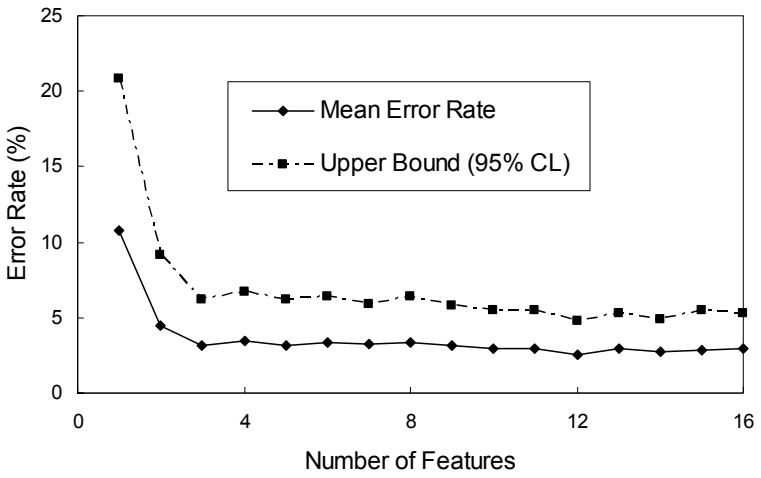


Figure 5.10 Performance of Quadratic Kernel SVM Classifier

5.2.4 Performance of Multi-Layer Perceptron

This section discusses the performance of the multi-layer perceptron (MLP). There are several factors that can affect the performance of an MLP classifier such as the number of hidden layers, the number of perceptrons in each hidden layer, and the choice of activation functions. When combining these factors, the MLP can have infinite number of architectural structures. There is no systematic way to find the optimal structure and it is impossible to evaluate all different structures to find the optimal structure by trial and error. Due to the relatively low dimension of the input (maximum dimension is 16), a 2-layer MLP is used and the number of perceptrons in each hidden layer is assumed to be the same. The activation function for all perceptron was chosen to be the log-sigmoid function shown in Figure 5.11. The performance of the MLP with different numbers of neurons for each hidden layer is plotted in Figure 5.12, in which "MLP22" refers to the case where the number perceptrons in the two hidden layers are 2 and 2, respectively.

MLP22 has the highest error rate. The performance of other MLP classifiers was similar but lower than MLP22. MLP44 was selected for its simple structure. Detailed information about the performance of MLP44 is plotted in Figure 5.13.

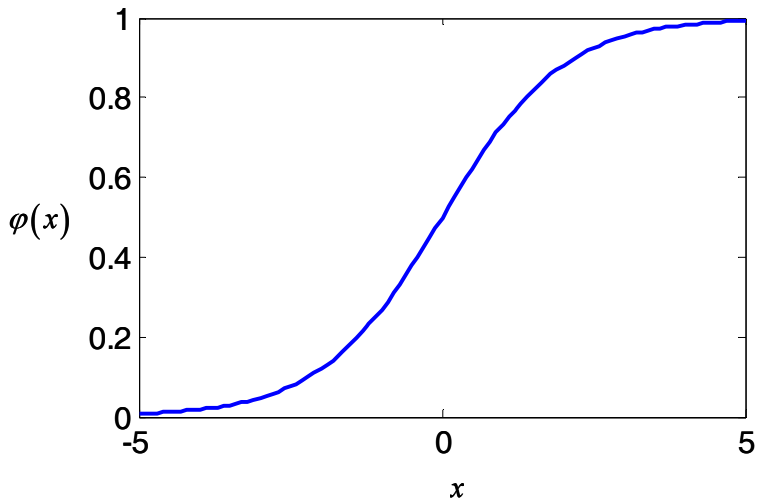


Figure 5.11 Log-Sigmoid Activation Function

The results indicate that the performance of the MLP is unsatisfactory when the number of features is too small or too large, indicating that the number of features must be carefully selected. The optimal number of features for MLP44 is 8 and the optimal mean error is 1.05% with an upper bound of 95% CL of 2.53%.

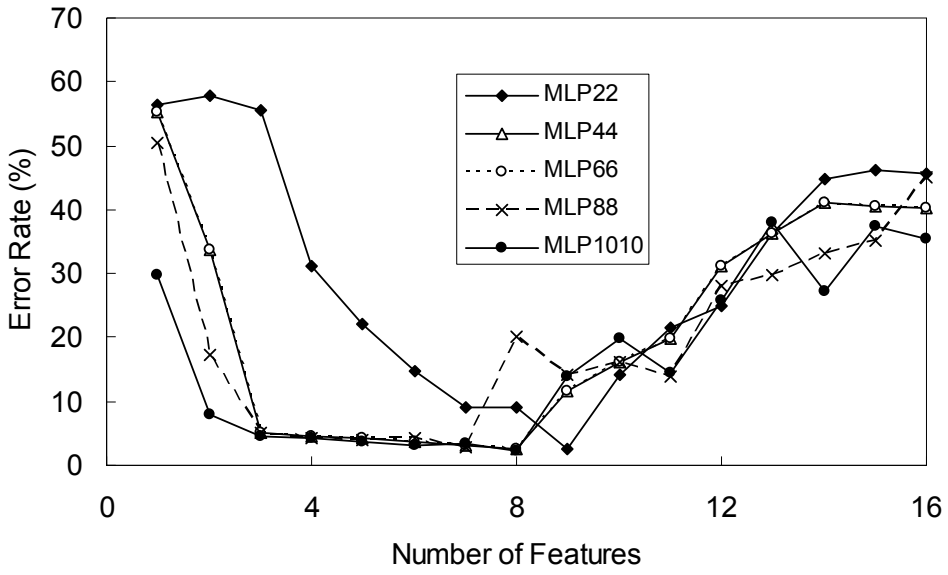


Figure 5.12 Performance of MLP with Different Structures

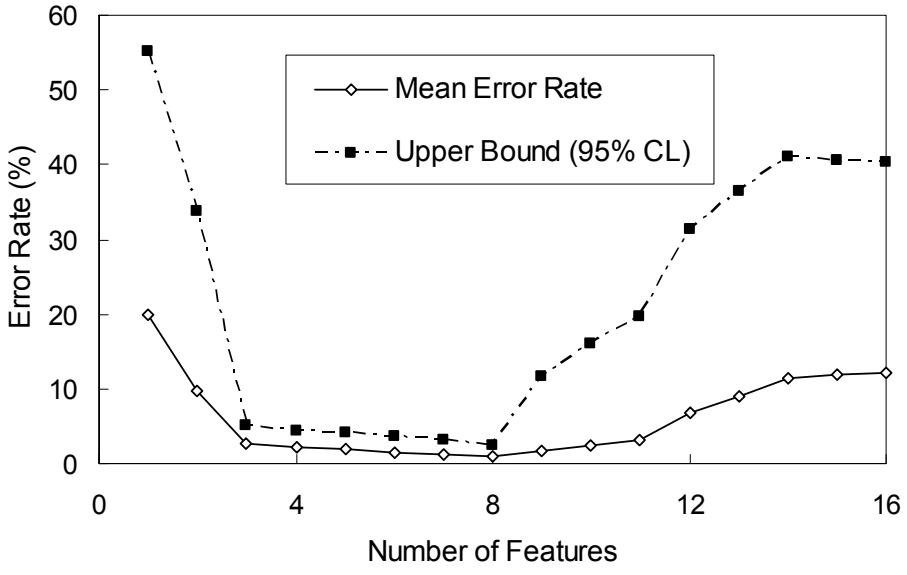


Figure 5.13 Performance of MLP44

5.2.5 Performance of Radial Basis Function

As described in Section 5.1.4, there are two factors that can affect the performance of the RBF classifier: the width of the RBF and the number of neurons. As with the MLP, there is no systematic way to find the optimal structure for the RBF. Combinations of limited number parameters were tried.

Figure 5.14 shows the effect of the number of neurons. In this evaluation, $\sigma = 100$ was assumed. A large value was assumed here to prevent the RBF classifier from capturing only the local effect. Better performance was achieved when N increased from 5 to 10. However, the increase in the performance was not significant when N increased beyond 10. By comparing the error rate for different cases, the optimal number of neurons was found to be 20.

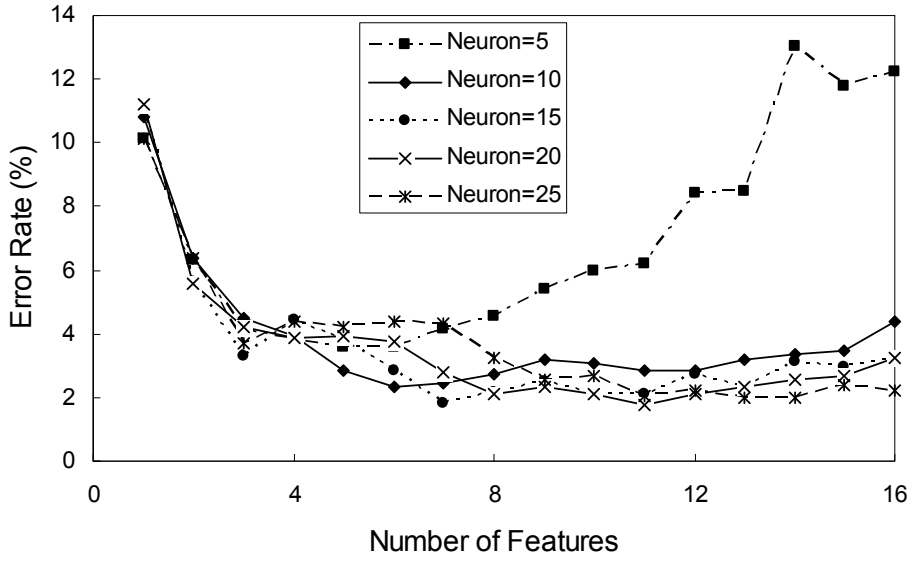


Figure 5.14 Effect of Number of Neurons on RBF

Figure 5.15 compares the error rate for different σ values. The optimal number of neurons of 20 was used. For the smaller variance, the performance became worse as the number of features increases. For the larger variance, the performance is better and more stable. Although the performance was sensitive when the variance is small but the effect of variance on the behavior of the RBF is not significant when it is greater than 10. Based on this analysis, the optimal variance was selected to be 10.

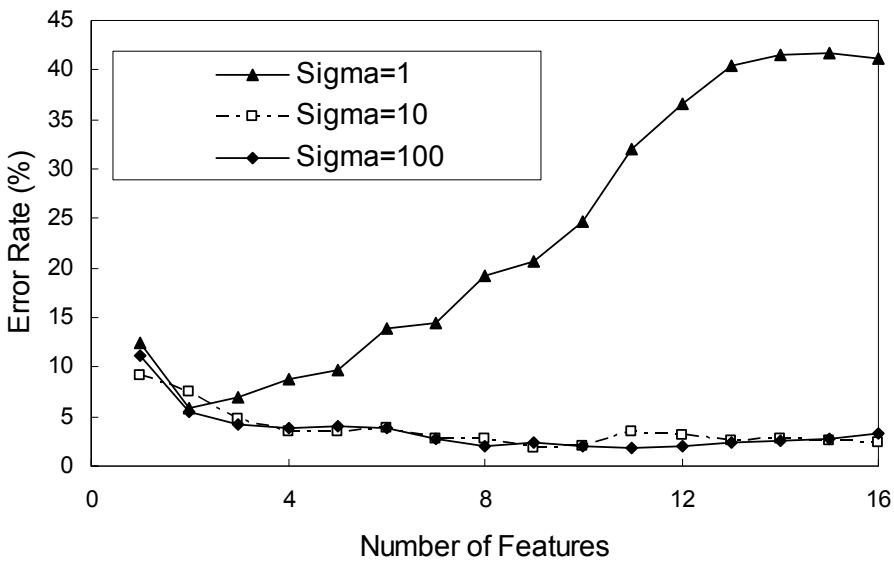


Figure 5.15 Effect of the Variance of RBF

Having optimized the number of neurons and the variance, the optimal RBF is a network with $\sigma = 10$ and $N = 20$. The performance of this classifier is shown below in Figure 5.16. The number of features yielding the best performance is 9, the lowest error rate is only 0.59% and the upper 95% CL is 1.80%.

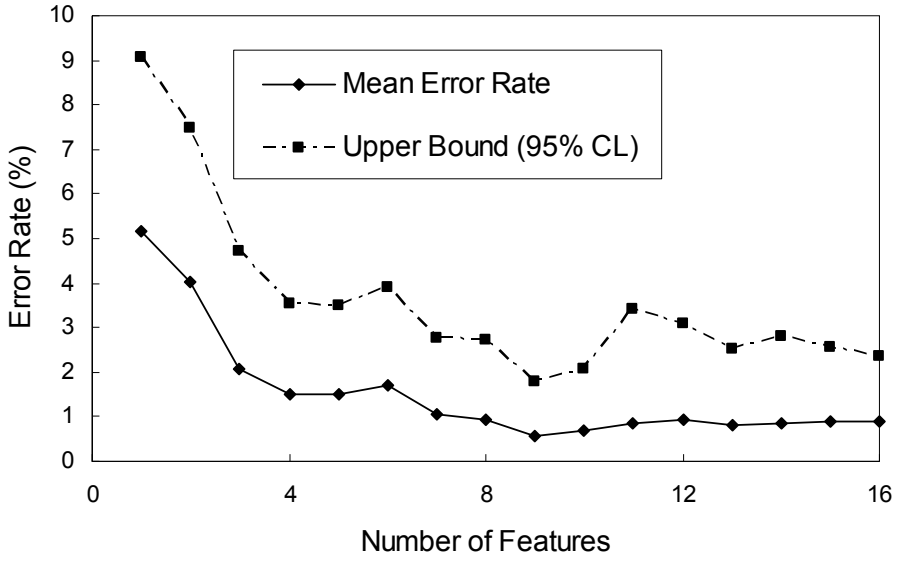


Figure 5.16 Performance of RBF Classifier

5.2.6 Selection of Detection Algorithm

In order to select the best classifier for delamination detection, the performance of different classifiers were evaluated in the previous sub-sections. Figure 5.17 compares the upper 95% CL of different classifiers. The RBF with $\sigma = 10$ and $N = 20$ has the smallest error rate and it was therefore selected to be used for delamination detection. The optimal number of features for the RBF with these parameters is nine.

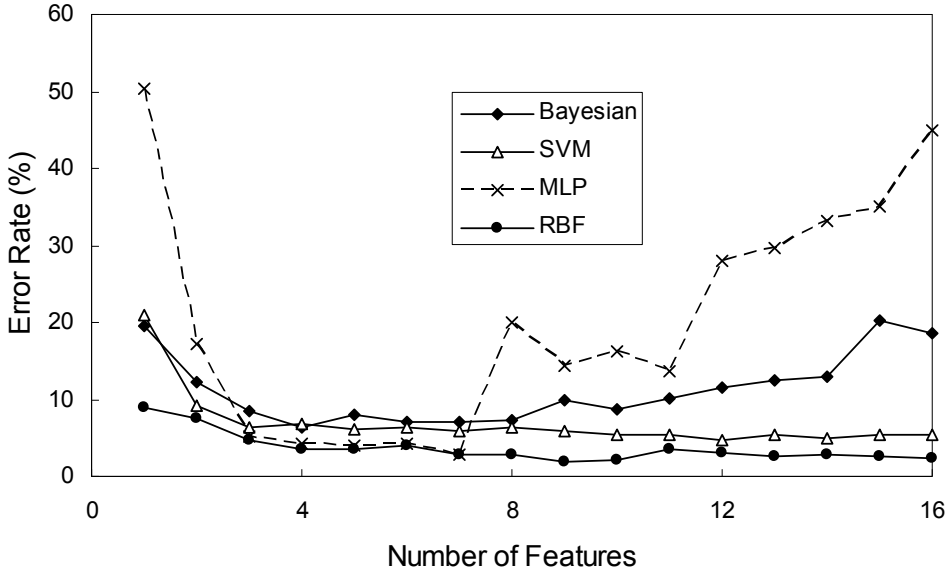


Figure 5.17 Comparison of Different Classifiers

5.2.7 Error Rate for Multiple Impacts

In field inspection, multiple impacts may be needed to increase the accuracy of the detection. In this case, the final decision can be made by comparing the number of impacts classified as solid with those that were classified as delaminated. If N is the number of impacts and N_1 and N_2 are the number of impacts classified as solid and delaminated, respectively, the final result of the class can be expressed as:

$$d = \begin{cases} solid & \text{if } N_1 \geq N_2 \\ delaminated & \text{if } N_1 < N_2 \end{cases} \tag{5.72}$$

In this case, an error occurs when the number of misclassification is greater than half of the total number of impacts. This compensates for the error due to variations in the impact signals and the error rate can be further reduced. Assuming that different impacts

are independent of each other and the error rate of an impact signal is ε_s, the error rate for multiple impacts will be:

$$\varepsilon = 1 - \sum_{i=0}^{\lfloor N/2 \rfloor} C_N^i \varepsilon_s^i \left(1 - \varepsilon_s\right)^{N-i} \tag{5.73}$$

where, $\lfloor N/2 \rfloor$ is the maximum integer smaller than $N/2$.

Figure 5.18 shows the envelope of the error rate for multiple impacts. The error rate drops as the number of impacts increases. If the error rate of an individual impact is 20%, the final error rate for 5 impacts is approximately 6%. If the single impact error is 10%, final error rate after 5 impacts is very small. Of course, increasing the number of impacts requires more processing time.

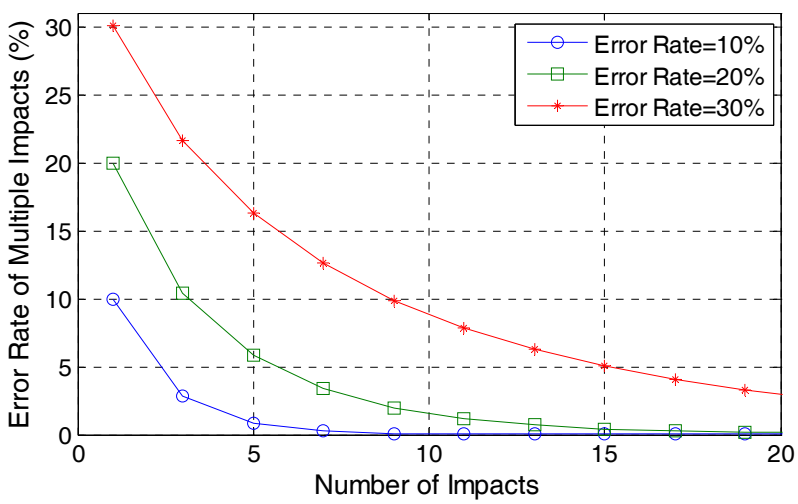


Figure 5.18 Error Rate of Multiple Impacts

ECR BRIDGE DECKS: DAMAGE DETECTION AND ASSESSMENT OF REMAINING SERVICE LIFE FOR VARIOUS OVERLAY REPAIR OPTIONS — PART I ..... **I**

TABLE OF CONTENTS ..... **I**

LIST OF FIGURES ..... **IV**

LIST OF TABLES ..... **VI**

CHAPTER 1 INTRODUCTION 1

1.1 MOTIVATION 1
1.2 PROBLEM STATEMENT 2
1.3 RESEARCH OBJECTIVES 3
1.4 ORGANIZATION OF THE REPORT 4

CHAPTER 2 LITERATURE REVIEW 7

2.1 DAMAGE IN CONCRETE 7
2.1.1 Crack 7
2.1.2 Honeycombing 8
2.1.3 Delamination 9
2.2 NON-DESTRUCTIVE EVALUATION (NDE) METHODS FOR CONCRETE 10
2.2.1 Impact Echo and Impulse Response 10
2.2.2 Ultrasonic Methods 14
2.2.3 Ground Penetrating Radar 17
2.2.4 Infrared Thermography 20
2.2.5 X-ray Imaging 22
2.2.6 Sounding Methods 23
2.3 SUMMARY 26

CHAPTER 3 NOISE CANCELLING ALGORITHMS 29

3.1 EVALUATION CRITERIA FOR NOISE CANCELLING ALGORITHMS 29
3.2 SPECTRAL SUBTRACTION 31
3.2.1 Theoretical Background 31
3.2.2 Performance Evaluation 33
3.3 ADAPTIVE FILTERS 33
3.3.1 Theoretical Background 33
3.3.2 Performance Evaluation 37
3.4 INDEPENDENT COMPONENT ANALYSIS 43
3.4.1 Theoretical Background 43
3.4.2 Performance Evaluation 48
3.5 MODIFIED ICA 51
3.5.1 Theory Background and Procedures 51
3.5.2 Performance Evaluation 55
3.6 SELECTION OF NOISE CANCELLING ALGORITHMS 58
3.7 SUMMARY 59

CHAPTER 4 FEATURE EXTRACTION 63

4.1 FEATURE EXTRACTION OF ACOUSTIC SIGNALS 63
4.1.1 Sub-band Energy 64
4.1.2 Energy of Wavelet Packet Tree 65
4.1.3 Psycho-Acoustic Features 68
4.1.4 Principal Component Analysis 71
4.1.5 Independent Component Analysis 73
4.2 PERFORMANCE OF DIFFERENT FEATURES 74
4.2.1 Criteria for Evaluation 74

4.2.2 Performance of Sub-band Energy *77*
4.2.3 Performance of the Wavelet Packet Tree *81*
4.2.4 Performance of MFCC *84*
4.2.5 Performance of Features Extracted by PCA *86*
4.2.6 Performance of Features Extracted by ICA *89*
4.2.7 Summary of the Section *92*
4.3 SELECTION OF THE FEATURE EXTRACTION ALGORITHM 92
4.3.1 Algorithm Selection Based on Weighted Rank *93*
4.3.2 Algorithm Selection Based on Error Rates *96*
4.4 SUMMARY 98

CHAPTER 5 PATTERN RECOGNITION AND DELAMINATION DETECTION 101

5.1 DETECTION ALGORITHMS 102
5.1.1 Bayesian-Based Classifier *102*
5.1.2 Support Vector Machine *105*
5.1.3 Multi-Layer Perceptron *108*
5.1.4 Radial Basis Function *113*
5.2 PERFORMANCE EVALUATION 115
5.2.2 Performance of Bayesian Classifier *117*
5.2.3 Performance of Support Vector Machine *119*
5.2.4 Performance of Multi-Layer Perceptron *121*
5.2.5 Performance of Radial Basis Function *124*
5.2.6 Selection of Detection Algorithm *126*
5.2.7 Error Rate for Multiple Impacts *127*
5.3 SUMMARY 131

CHAPTER 6 DELAMINATION DETECTION SYSTEM DEVELOPMENT AND ALGORITHM VERIFICATION 135

6.1 HARDWARE DEVELOPMENT 135
6.2 SOFTWARE DEVELOPMENT 136
6.2.1 Training Process *137*
6.2.2 Inspection Process *138*
6.2.3 Crank Noise Removal *140*
6.2.4 Implementation of the Algorithms *142*
6.3 VERIFICATION OF ALGORITHMS 144
6.3.2 Lab Experiments *146*
6.3.3 Field Inspection *149*
6.4 SUMMARY 153

CHAPTER 7 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK 155

7.1 SUMMARY OF THE STUDY 155
7.2 MAJOR CONCLUSIONS 157
7.3 RECOMMENDATIONS FOR FUTURE WORK 159

BIBLIOGRAPHY 161

5.3 Summary

This chapter evaluated and compared four types of commonly used classifiers: the Bayesian classifier, the support vector machine, the multi-layer perceptron network and the radial basis function network. The classifier for delamination detection was then selected based on their performance.

This chapter first briefly described the theoretical background of four classifiers.

1. The Bayesian classifier finds the decision surface by minimizing the probability of the misclassification. This classifier requires prior information about the underlying distribution of the input. If the underlying distribution is normal and if the covariance matrix is diagonal, the decision surface of the Bayesian classifier has a quadratic form. If the diagonal elements of the covariance matrix are equal, the decision surface is further reduced to a hyper-plane.

2. The support vector machine classifies different classes by finding a hyper-plane that maximizes the margin of separation and it does not require prior information about the underlying distribution of the input. The optimal hyper-plane can be found by using a Lagrange multiplier. For the case where the input data are not linearly separable, a non-linear kernel function must be used to transform the input data to a higher dimensional space where the classification becomes a linearly separable problem.

3. The Multi-layer perceptron network consists of several layers of perceptrons. By adaptively changing the synaptic weights that connects different perceptrons, the

mean square error between the desired output and the network output is minimized. This is equivalent to finding an optimal mapping between the inputs and the desired outputs.

4. The radial basis function network is another way to find the optimal mapping between the inputs and outputs. The difference is that the RBF consists of only one hidden layer and the mapping function is symmetric around the center. The classifier is trained by adjusting the centers of the RBF and the synaptic weights that connects the hidden layer and the output layer such that the error between the desired outputs and the actual outputs of the system is minimized.

The second part of the chapter evaluates the performance of the classifier because different classifiers use different optimization criteria and different approaches to find the optimal mapping. Also, the number of features has an important role on performance. Therefore, the error rate and the upper 95% confidence interval under different numbers of features were plotted and used to compare performance.

1. By comparing two types of Bayesian classifiers, it was found that the quadratic Bayesian classifier had a better overall performance than the linear Bayesian classifier. “redundant” features (when the number of features exceeds a certain number) had a negative effect on the quadratic Bayesian classifier, while the linear Bayesian classifier is not sensitive to redundant features.

2. For the SVM, both quadratic and linear kernel functions were compared. For both types of kernel functions, the SVM is not sensitive to the redundant features and

the SVM with quadratic kernels had a better performance than the SVM with linear kernel functions.

3. The performance of the MLP increased significantly when the number of perceptrons in each hidden layer increased from 2 to 4. However, further increase in the number of perceptrons in the hidden layer did not yield significant improvement in performance. The MLP network was also sensitive to redundant features.

4. The performance of the RBF network was poor for small values of N and σ, especially when the number of features was high. Increasing N and σ improved the performance. However, improvement in the performance was not significant when the values of N and σ exceeded certain values. By comparing the performance of different classifiers, it was found that a RBF with $\sigma = 10$ and $N = 20$ had the best performance. The optimal performance of this classifier was achieved when the number of features was 9.

Lastly, the chapter also discussed about the error rate when multiple impacts were performed at the same spot. The error rate dropped quickly with an increase in the number of impacts. Due to its superior performance, the RBF classifier will be used for the delamination detection.

(This page intentionally left blank.)

CHAPTER 6

DELAMINATION DETECTION SYSTEM DEVELOPMENT AND ALGORITHM VERIFICATION

Detailed information about the delamination detection algorithms were described in previous chapters. After selecting the algorithms, it was necessary to test the performance of the combined system under different conditions. This chapter briefly describes how the different components, i.e., noise cancellation, feature extraction and selection and pattern recognition are combined to form an automatic detection system, and how several minor practical implementation problems were solved. After the system was tuned, its performance was tested using experimental and field data.

6.1 Hardware Development

The impact machine was designed and fabricated to automatically impact the concrete surface with constant energy. The impact was created by the free fall of the impactor from a constant height. The impactor was a #8 stainless steel bar with a ball-shaped head. The impactor was picked up by the pin on the flywheel and was lifted as the rotation of the flywheel. When the flywheel rotated to a certain location where the pin can no longer hold the handle on the impactor, the impactor will be released and fall freely from that height and impact the ground. A catching mechanism was also mounted on the cart to prevent the multiple impacts due to impactor rebound. The impact and ambient sounds were recorded by a condenser microphone. This microphone is directional and records the sound within a short distance, which helps limit extraneous

noise. There were two microphones mounted on the cart. The primary microphone was mounted under the base of the cart, pointing toward the impact point to record the impacting sound. Sound proofing curtain was mounted as a physical barrier to the traffic noise and a wind-isolator. The secondary microphone was mounted on the frame to measure the ambient noise. Figure 6.1 shows the proto-type of the impacting machine.

Figure 6.1 Proto-type of the Impacting Cart

6.2 Software Development

There are two major components in this automatic delamination detection system: classifier training and signal processing. For practical implementation, training will likely be conducted offline where selected signals from previous tests are used to train the

classifier and to find effective features. Once the classifier is trained and features are selected, the information can be saved as an external file for future inspection. In the inspection, a data acquisition system is used to record the sound. The recorded signal is first filtered through the modified ICA described in Chapter 3. The impact sound is extracted and the features, in this case MFCC, are calculated and then classified using the detector obtained in the training process. This section describes the training and detection process in detail.

6.2.1 Training Process

The training process was performed offline using existing data files where information including the concrete type, features and original signals were stored. The training process was as follows:

1. A certain number of training data files were selected. The selection of the training data needs to be representative of the structure to be inspected.

2. Concrete types solid and delaminated and MFCCs were directly read from the data file. The mutual information between the concrete type and each MFCC was calculated using Equation (3.19). The values of mutual information were compared and MFCCs with high values were selected and used as effective features to train the classifier.

3. The classifier (RBF neural network) was trained using the effective features of the training signals. The training process is basically an iterative one in which the coefficients of the neural network are adaptively updated such that the error

between the actual output and the target output is minimized. The training was performed using the Artificial Neural Network tool box in MATLAB.

4. Once the training was completed, the classifier and the indexes of the effective features were saved to a classifier file that could be used for future inspection.

The flow chart of the training algorithm is shown in Figure 6.2.

6.2.2 Inspection Process

The inspection process was performed at a bridge site and the analysis (including filtering and detection) was completed in a semi-real-time manner. The signal was first recorded and then processed by the computer. After the process at one spot was completed, the computer can be used to process new data. The estimated time needed to perform the analysis (filtering, feature extraction and detection) for 3 seconds of signal sampled at 10 kHz is about 6 seconds on a laptop computer (1.8Ghz CPU and 3Gb RAM).

The inspection process comprised of the following:

1. The impact signal was recorded by two microphones and digitalized by a data acquisition card to be processed in a computer. The primary microphone was placed toward the impact point to record the impact sound and a small portion of ambient noise. The secondary microphone was pointed away from the impact point to record ambient noise and a fraction of the impact sound.

2. The fly-wheel and mechanism that lifted the impact rod made a “crank” sound. The crank sound in the recording was removed using the “cross-correlation” method described later in this chapter.

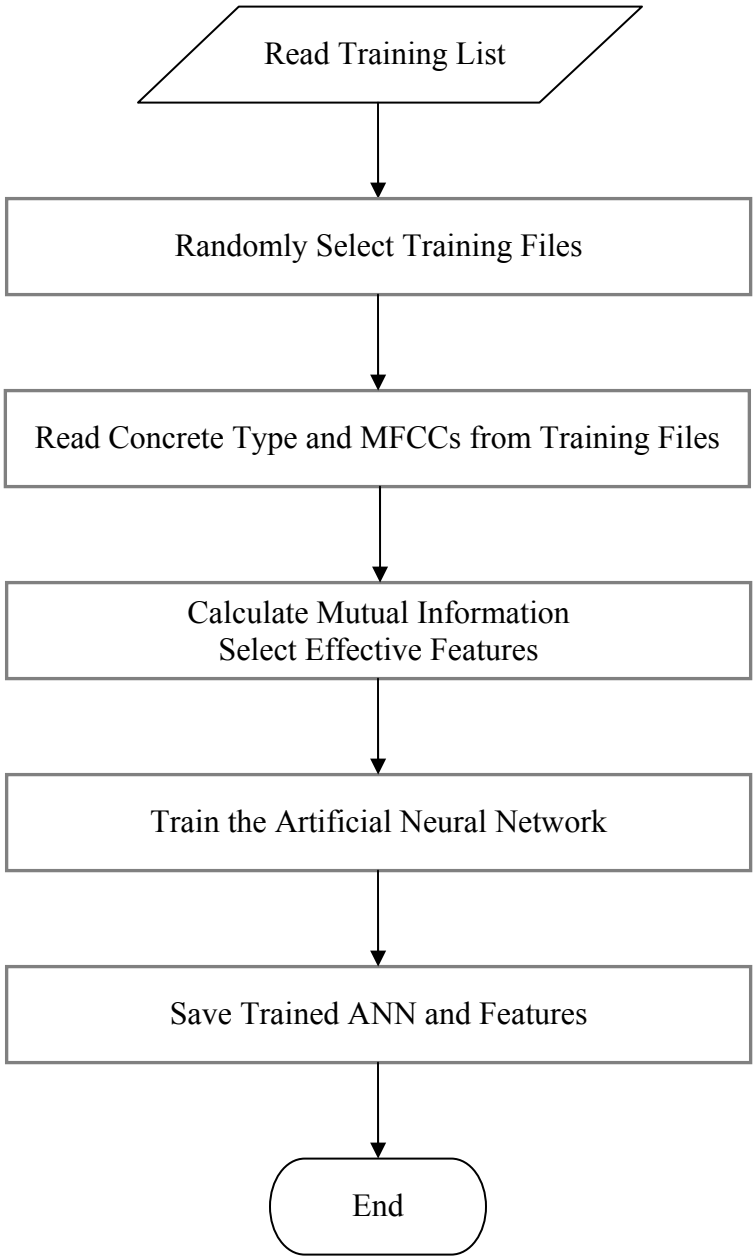


Figure 6.2 Flow Chart of the Training Process

3. The impact sound and the noise were separated from the recordings using the modified ICA described in Chapter 3

4. The filtered impact signal was obtained using the results in Steps 2 and 3. The MFCCs of each impact were calculated.

5. The MFCCs obtained in Step 4 were used by the classifier to determine whether the concrete was delaminated or solid.

6. Information about the recording, such as the concrete condition, calculated features and the original recording etc. was saved as a data file for future use.

7. Steps 1 to 6 were repeated until the inspection was completed.

The flow chart of the inspection process is shown in Figure 6.3.

6.2.3 Crank Noise Removal

In addition to the impact sound, there was a crank sound produced by the impacting machine when the pin on the flywheel hits picks the handle on the impactor. The existence of this sound had a negative effect on the delamination detection. However, removal of the crank sound could be achieved in the time domain using the simple cross-correlation method described below.

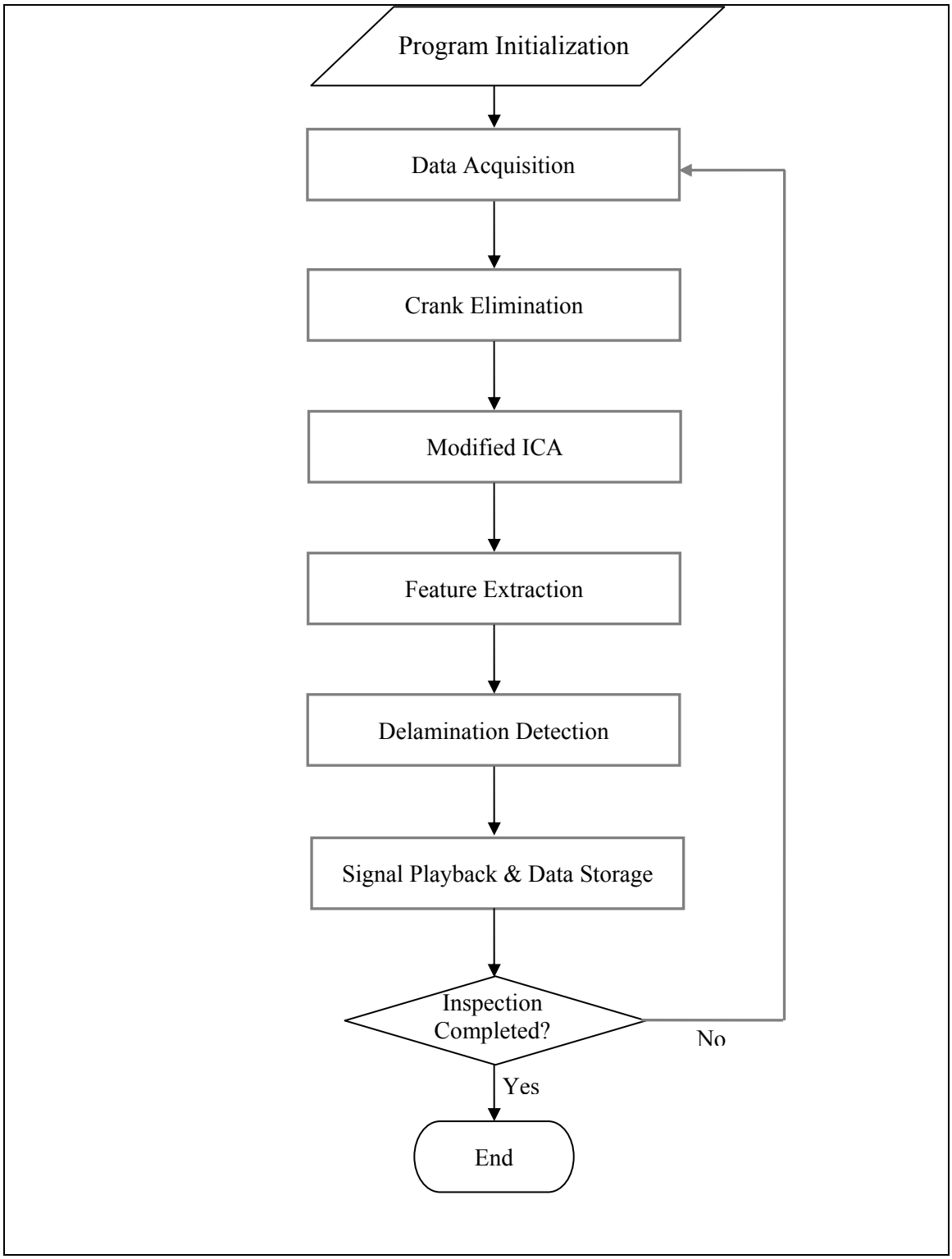


Figure 6.3 Flow Chart of the Inspection Process

Figure 6.4 (a) shows a typical waveform of the recording. The relative location between the impact and the crank was fixed (the impact sound always occurred after the crank). Therefore, the impact sound could be extracted by locating and removing the crank. Since the crank sound comes from the impacting machine and is consistent for a given machine, the recording containing the crank sound will have a high correlation with the crank signal from the same machine. The crank sound was removed as follows:

1. A sample of the crank sound was obtained by running the machine in a quiet environment where the crank sound could be easily identified by detecting the peak of the waveform.

2. The cross-correlation between the crank sample and the actual recording was calculated. Figure 6.4 (b) shows a typical cross-correlation function. The peak value of the cross-correlation function matches well with the location of the crank.

Once the crank sound was identified, it was eliminated by zeroing the recording during the crank period.

6.2.4 Implementation of the Algorithms

Detailed information about the algorithms for training and inspection were described in previous chapters. One more step was needed to develop a practical tool that can be used in field inspection. This was to implement the algorithms into an executable program with proper user interface.

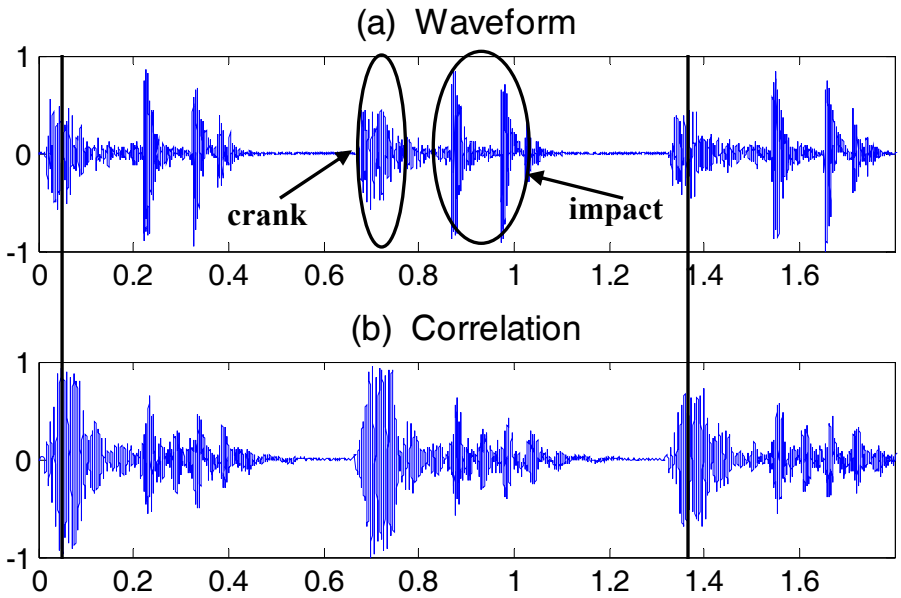


Figure 6.4 Crank Removal

Combined programming in MATLAB, C++ and LabVIEW was used to implement the algorithm. LabVIEW provides a simple interface between the computer and the data acquisition hardware. It includes many built-in libraries and instruments drivers, and no programming at the hardware level is needed. Last, but not the least, it has a graphical programming environment. That makes it very easy to create a graphical user interface (GUI). MATLAB was selected because it provides a very convenient programming language and has powerful toolboxes and built-in functions. MATLAB is also capable of converting scripts into executable (.exe) files or dynamic link library (.dll) files, both of which can be run independent of the MATLAB environment. The problem associated with combining LabVIEW and MATLAB is that it is difficult to communicate data directly between them. Therefore, C++ program were used as a wrapper or bridge to enable the data communication between LabVIEW and MATLAB. The data communication inside the developed system is shown in Figure 6.5.

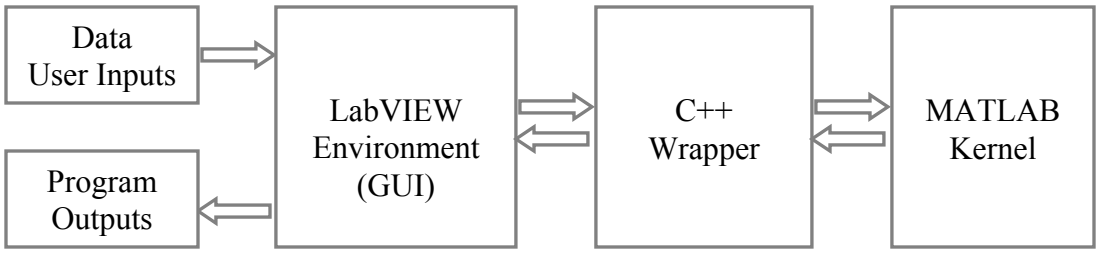


Figure 6.5 Data Communication

Figure 6.6 and Figure 6.7 show the GUI for the training module and the inspection module, respectively.

Detailed information about the algorithms and the wrapper as well as the LabVIEW flow chart are included in Appendix A.

6.3 Verification of Algorithms

The performance of the proposed algorithms was tested using two types of data: experimental data and field data. Experimental data was used to check the performance of the algorithm under quiet conditions as well as the influence of the noise cancelling algorithm on detection accuracy. Field data was used to test the performance of the system under real environments. The results of the tests are described in this section.

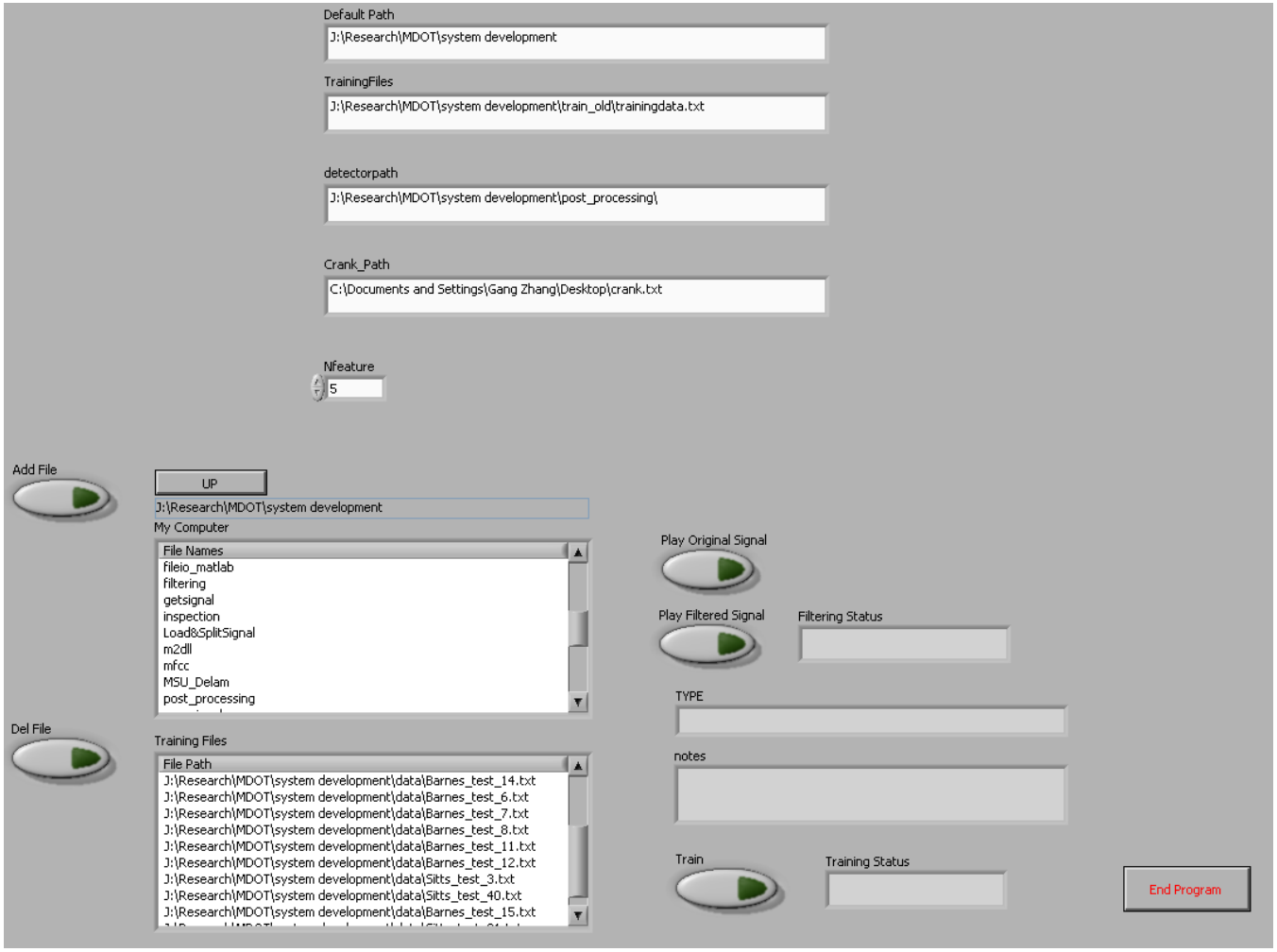


Figure 6.6 GUI forTraining Module

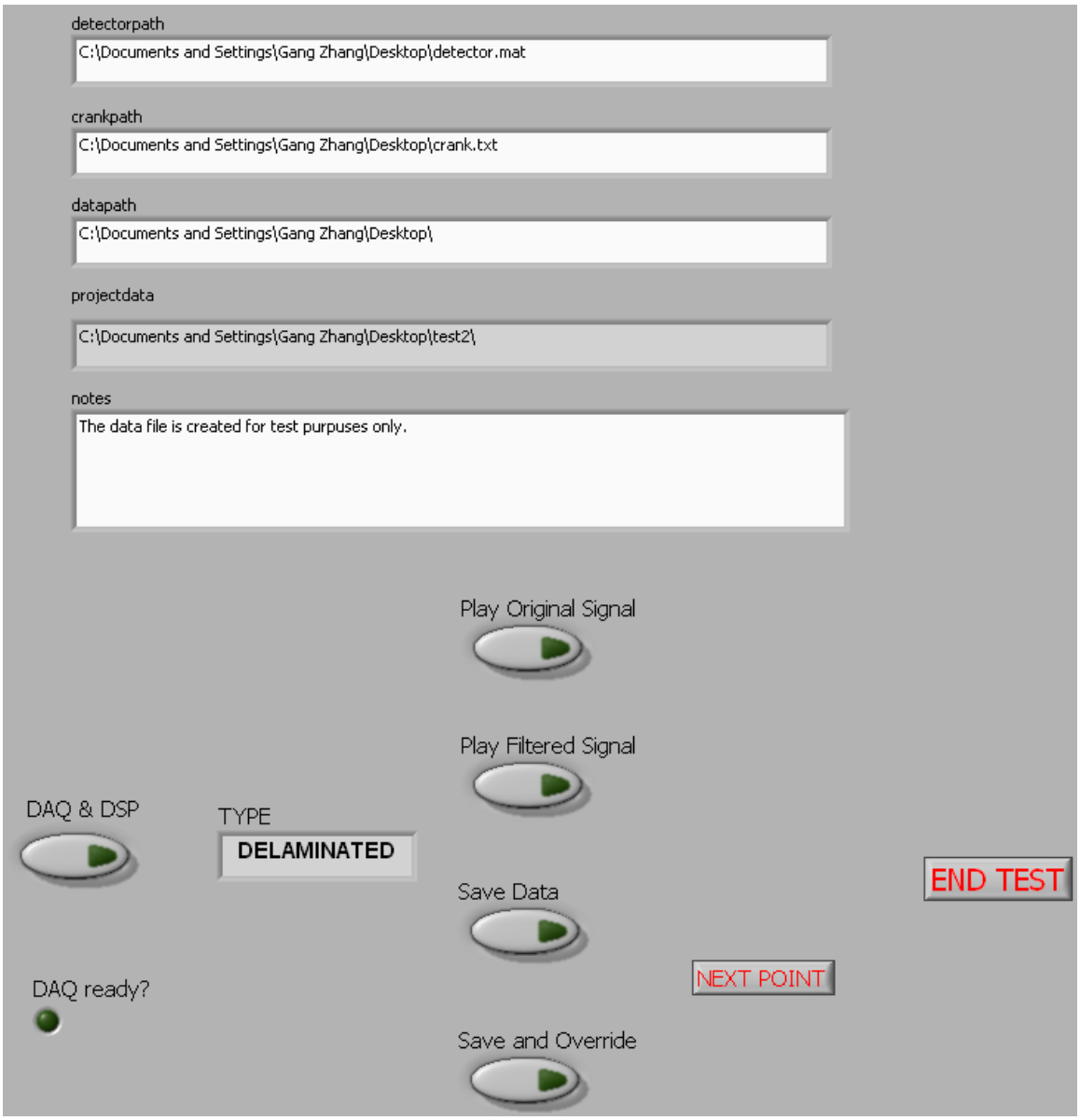


Figure 6.7 GUI for Inspection Module

6.3.2 Lab Experiments

To verify and check the performance of the detection algorithm, impact testing was carried out on a slab constructed in the laboratory. Figure 6.8 and Figure 6.9 show a photograph and an elevation-view of the slab with artificial delamination. The strength of the concrete was 4 ksi (27.6 MPa), which is typical for concrete used in bridge decks.

The delamination was simulated by different thicknesses of the top layer. The thickness of the slab was 9 inches and the thicknesses of the two “delaminated” parts are 3 inches and 6 inches, respectively to simulate different delamination depths. The test results showed that the sound produced from the 6-inch delamination was very similar to that produced by the solid concrete. This is because the energy of the impact was not large enough to excite the mode where the difference between the 6-inch delamination and the solid concrete (9 inches in thickness) could be clearly observed. Therefore, in the analysis of the result, signals from these two cases were combined and labeled as “solid”.

Figure 6.8 Slab for Lab Experiment

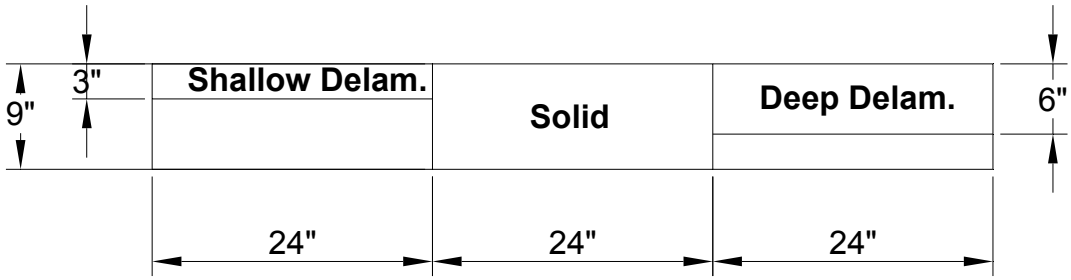


Figure 6.9 Side View of the Slab

In the lab environment, the noise level was low and the recordings from the primary microphone were clean enough to perform the analysis. To evaluate the performance of

the algorithm with noisy input, recordings with different signal-to noise ratios (SNRs) were simulated by mixing recorded traffic noise signals with the impact signal obtained in a quiet laboratory environment. Different noise levels were obtained by mixing the scaled impact signal with noise signals as shown in Equation (3.19) except that the impact sound and the noise were convoluted versions. Four noise levels were considered: quiet condition ($\alpha = \infty$), low noise level (α =10), medium noise level (α = 1), and high noise level (α = 0.1). The modified ICA was used to perform noise elimination and then the MFCCs of both noisy recordings and the filtered signals were computed for comparison.

A total of 228 impacts were recorded on two different days. 120 impacts were obtained from solid concrete and 108 impacts were obtained from delaminated concrete. 40 randomly selected impacts were used for feature extraction and classifier training. The remaining signals were classified by the trained classifier. The average error rates under different conditions were calculated using the same method described in Chapter 5. The results are given in Table 6.1.

The results show that MFCCs performed well in a quiet environment (large SNR) yield an error rate of only 2.3%, but the accuracy of the algorithm drops (error rate increases) with increase in the noise level if the signals are not pre-processed with the modified ICA noise cancelling algorithm. When the signals are filtered with the modified ICA algorithms, the detection algorithm becomes much more noise robust and the error rate remained constant (around 5%) for all noise levels considered.

Table 6.1 Error Rate under Different Noise Levels

SNR (α)	Measurements	Error Rate (%)	
		Filtered Signals	**Noisy Signals**
∞	$m = s$	2.31	N/A
10	$m = 10s + n$	5.17	8.48
1	$m = s + n$	5.14	12.39
0.1	$m = 0.1s + n$	5.85	26.15

6.3.3 Field Inspection

To test the detection algorithms under field conditions, tests were performed on two bridges near Mason, Michigan. Bridge 1 is located on Barnes Road over US 127 (shown in Figure 6.10) and Bridge 2 is on Sitts Road over US 127 (shown in Figure 6.11). Both bridges had concrete decks with delaminations. The concrete condition at several spots was first identified through traditional bar tapping (i.e., impacting the bridge deck using a steel bar and listening to the sound). The impact machine described in the previous section was then used to test these spots and the sound signals were collected using the data acquisition card.

Figure 6.10 Barnes Road over US127 (Bridge 1)

Figure 6.11Sitts Road over US127 (Bridge 2)

The analysis of the signal was performed offline to investigate the factors that influenced the performance of the algorithms. Table 6.2 compares the error rate difference between the original signals and the filtered signals. The error rate was calculated as described in Chapter 5: training signals were randomly selected from the training pool, and then the selected features and trained classifiers were used to classify the data in the testing pool. Due to the randomness in the selection of the training samples, the error rate could be different. To consider this, the error rate was calculated based on

the average of 100 simulations. It can be seen that both the original signals and filtered signals gave very good results. The advantages of the filtered signals were not significant. This is because both bridges are not on a busy street and the noise level was low when the inspection was performed. This is consistent with the results of the lab experiments, indicating limited performance improvement due to filtering if the noise level is low.

Table 6.2 Error Rates of Original Signals and Filtered Signals

Signal Type	**Signals from Bridge 1**	**Signals from Bridge 2**	**Signals from both Bridge 1 and Bridge 2**
Original Signals	0.23%	0.74%	1.06%
Filtered Signals	0.30%	0.71%	0.69%

In the previous analysis, the training signals were randomly selected from the data pool. However, in real situations, the training signals can only be obtained from existing recordings. Due to the limited amount of field data, data obtained from Bridge 1 was divided into two groups: the first half (labeled as group A) and the second half (labeled as group B), similarly, the data from Bridge 2 were divided into group C and group D. The training data and testing data were randomly selected from the recordings in the training pool and testing pool. The number of training data and testing data were 150 and 100, respectively, and were fixed for all cases. Table 6.3 shows the error rates under different training sets. Comparing Table 6.2 and Table 6.3, it is clear that the error rates listed in Table 6.3 are higher than those in Table 6.2. This is because in Table 6.2, the training set contains all the information of the testing set and the classifier was tuned to this particular type of data. However, the inspector does not have prior information about the bridge to be inspected and therefore the error rate will increase. For the same testing pool, different training pools would give different error rates and sometimes the error rates can

be quite high. However, this does not invalidate the proposed algorithms. As can be seen from Table 6.3, if the number of groups in the training pool is increased, the average error rate drops for most groups (except for group D, possibly due to variance in the data). If sufficient different cases are selected, the error rate can be lowered to a satisfactory level (around 15% error rate for single impact). If multiple impacts are collected at the same spot, the error rate drops exponentially as shown in Figure 5.16. Even though the error rate of a single impact can be as high as 17.67%, the error rate can be dropped to less than 5% if 5 impacts are recorded. Further, as more bridges are inspected and more data becomes available, the performance of single impact detection will also improve.

Table 6.3Error Rates under Different Training Sets

Groups in Training Pool	Groups in Testing Pool	Error Rate (%)	Average Error Rate (%)	Groups in Training Pool	Groups in Testing Pool	Error Rate (%)	Average Error Rate (%)
B	A	4.66		A	C	23.59	
C	A	18.42	9.18	B	C	26.86	23.73
D	A	4.46		D	C	20.73	
BC	A	5.61		AB	C	23.20	
BD	A	2.05	5.28	AD	C	17.92	20.14
CD	A	8.17		BD	C	19.29	
BCD	A	3.20	3.20	ABD	C	17.67	17.67
A	B	9.38		A	D	6.31	
C	B	20.33	13.20	B	D	11.29	9.25
D	B	9.89		C	D	10.14	
AC	B	8.38		AB	D	9.88	
CD	B	13.98	9.91	BC	D	15.69	13.29
AD	B	7.38		AC	D	14.30	
ACD	B	8.79	8.79	ABC	D	12.44	12.44

6.4 Summary

This chapter described the development of inspection and training systems and then the performance of the system was verified using both experimental data and field data.

The first section of the chapter focused on the development of the system. In the training process, a certain number of recordings were selected from the existing data files as training data. The features and the concrete condition were read directly from the file. The mutual information of each feature was calculated and compared. The features with a high value of mutual information were selected and used to train the RBF neural network classifier. The index of the selected features and the trained neural network were saved as a file for future use. In the inspection system, the data were first collected by data acquisition system and the crank sound from the impacting was eliminated by a cross-correlation based algorithm. In this algorithm, the cross-correlation between the recorded signal and the crank signal was calculated and was used to identify the locations of the crank sound. The crank sound was eliminated by zeroing the crank period identified by the algorithm. After removal of the crank sound, the signal was processed by the modified ICA to separate the impact signal and the ambient noise. The features (MFCCs) of the filtered signal were calculated and the concrete condition was determined by the classifier obtained through the training process. The algorithms were incorporated into a LabVIEW program with MATLAB and C++ components. A LabVIEW project was created to provide a graphical user interface (GUI) and perform data acquisition. The inspection process took around 6 seconds to process a 3 seconds length of recording with a sampling rate of 10 kHz.

The performance of the algorithms was verified using both experimental and field data. The results from the experimental data showed that the algorithms worked well under quiet conditions. However, the error rate increased with increasing the noise level. The introduction of the noise cancelling algorithm made the system noise robust and produced better results. The field data indicated that the selection of the training data has an important effect on performance. If the training sets are not representative of the test set, the error rate can be quite high. However, the error rate drops if sufficient number of different training sets were used. Also, multi-impact on the same location will further increase the accuracy. By recording five impacts on each location, the error rate can be maintained to less than 5% even in the worst scenario shown in Table 6.3. Therefore, the proposed system is fast and accurate enough to be used in field inspection.

CHAPTER 7

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

This chapter first summarizes the research efforts to improve the sounding using an impact rod. The conclusions obtained during the investigation are also included. The last section of the chapter provided several directions for further investigation and potential areas in which the research in this study is beneficial.

7.1 Summary of the Study

Even though sounding methods are simple, fast and inexpensive for detecting delamination in concrete bridge decks, their performance can be undermined by traffic noise in adjacent lanes and the subjectivity of the operator. To improve the performance of the traditional sounding methods, this study addresses the two factors that reduce their performance. The sounding method used in this work was restricted to impacts by a rod because it produced cleaner signals than a chain drag. Several noise cancelling algorithms were investigated including spectrum subtraction, adaptive filtering, traditional independent component analysis (ICA) and modified ICA. The performance of the algorithms was evaluated based on the numerical criterion signal to distortion ratio (SDR). The results showed that the modified ICA has the best performance among all the candidate algorithms compared in this study and it was selected as the noise cancelling algorithm in this work.

After the noise signals and the impact signals were successfully separated, the features of filtered signal were extracted. Different feature extraction algorithms were used to extract features of the signals: energy of sub-band using FFT, energy of the wavelet packet tree, mel-frequency cepstral coefficient (MFCC), principal component analysis (PCA), and independent component analysis (ICA). The performance of different algorithms was evaluated using repeatability, seperability and mutual information measures. The extracted features were further reduced based on the criterion of mutual information to select those features that best separated the solid sound and the delamination sound. Based on a weighted rank and the error rate, the MFCCs were selected as the best features.

Delamination detection was posed as a classification problem and several candidate classifiers including the linear and non-linear Bayesian classifier, the support vector machine (SVM), the multi-layer perceptron (MLP) neural network and the radial basis function (RBF) neural network were considered. Selected features of the signals in a training set were used to train the classifiers. The selected features and trained classifiers were used to classify the signals in the test set. The performance of the different classifiers was evaluated using the error rate. The results showed that the RBF had the lowest error rate and hence it was selected as the classifier for delamination detection.

The selected noise cancelling and delamination detection algorithms were implemented in LabVIEW, MATLAB and C/C++ routines for use by general operators. The performance of the system was verified using experimental data obtained in the laboratory and field data obtained from two bridges. The results showed that delaminations could be accurately detected by the proposed algorithms.

7.2 Major Conclusions

This study improved the performance of sounding using an impact rod by including noise cancellation, feature extraction and selection, and pattern recognition. The improvement in the performance was verified using data from laboratory experiments and field tests. The conclusions obtained from the investigation are summarized below.

Noise Cancelling Algorithms

1. Spectrum subtraction is very simple to implement, but it requires that the noise signal is short-term stationary which is not guaranteed for traffic noise.

2. The recursive least square (RLS) adaptive filter can adaptively cancel the noise in the reference recording from the primary recording. However, because the source is also in the reference recording in real situations, it will partially cancel the source and lead to distortion.

3. Independent component analysis (ICA) can separate linear mixtures without any prior information about the sources. However, recordings in real situations are convolutive mixtures and cannot be separated by traditional ICA.

4. A modified ICA is the only algorithm that worked with real signals and was selected to cancel the traffic noise in this study. The pre-defined delay required in this method can be estimated through simple calculations.

5. The results also showed that SDR provides an adequate comparison among different algorithms.

Feature Extraction and Feature Selection

1. Five different feature extraction algorithms were evaluated against three criteria: repeatability, separability and mutual information. The repeatability and separability measures did not provide a consistent comparison due to an ill-conditioning problem. However, mutual information provided a better indication of separability.

2. Different algorithms extract different features of the signals and features extracted by the same algorithm perform differently. Existence of features with poor separability may have a negative effect on the classification and mutual information was used as a criterion to eliminated unwanted features.

3. The weighted rank based on the repeatability and separability measures and the error rate based on a linear Bayesian classifier were used test the performance of features. The results from the weighted rank and the error rate agreed well and MFCC was selected as the feature extraction algorithm for this research.

Pattern Recognition and Delamination Detection

1. Quadratic Bayesian classifiers have better performance than the linear Bayesian classifier but are sensitive to “redundant” features.

2. The quadratic support vector machine (SVM) has a slightly better performance than the linear SVM. Both types of SVM are not sensitive to redundant features.

3. The performance of the multi-layer perceptron (MLP) network increases with the number of perceptrons in the hidden layers but the increase is not significant if the

number of perceptrons is greater than four. The MLP is sensitive to the presence of redundant features.

4. The performance of the radial-basis-function (RBF) network increases with the increase of the number of neurons and the spread of the activation functions of each neuron. But the increase becomes insignificant after the number of neurons and the spread have reached a certain value.

5. An RBF with $\sigma = 10$ and $N = 20$ had the best performance amongst all the classifiers and was used for delamination detection.

6. The detection accuracy can be further improved by performing up to five impacts on the same spot.

Algorithm Verification

1. The performance of filtered signals and original signals are both satisfactory under low noise levels but only the filtered signals can provide good results under noisy conditions.

2. To achieve good performance, the training data must be representative of the test data. In real situations, as the amount of training data increases the performance will improve.

7.3 Recommendations for Future Work

Even though the performance of the system is satisfactory for both experimental data and field data, the following enhancements may be possible:

1. Data needs to be obtained from different types of bridges so that a representative training set is available for more accurate detection. In general, the more data, the better the detection.

2. Investigation is needed on how to select the optimal training set based on different conditions or bridges. Rather than using all available data, it may be more optimal to use a smaller subset of data based on some criteria.

3. The development of faster noise cancelling and pattern recognition algorithms could improve the speed of detection/operation of the AIDD system.

BIBLIOGRAPHY

[1] http://www.nationalconcretebridge.org/advantage.html; Accessed on Feb. 7., 2010.

[2] http://www.asce.org/reportcard/2009/grades.cfm; Accessed on Feb. 7, 2010.

[3] Rens, K.L. and Transue, D.J., *Recent Trends in Nondestructive Inspections in State Highway Agencies.* Journal of Performance of Constructed Facilities, 1998. **12**(2): pp. 94-97.

[4] Rhazi, J., *NDT in Civil Engineering: The Case of Concrete Bridge Decks.* CSNDT Journal, 2000. **21**(5): pp. 18-22.

[5] Yehia, S., et al., *Detection of Common Defects in Concrete Bridge Decks using Nondestructive Evaluation Techniques.* Journal of Bridge Engineering, 2007. **12**(2): pp. 215-224.

[6] Carino, N.J. *The Impact-Echo Method: An Overview*. Proceedings of *the 2001 Structures Congress & Exposition*. 2001. Washington, DC.

[7] Mindess, S., *Acoustic Emission and Ultrasonic Pulse Velocity of Concrete.* International Journal of Cement Composites and Lightweight Concrete, 1982. **4**(3): pp. 173-179.

[8] Sansalone, M. and Carino, N.J., *Impact-Echo Method: Detecting Honeycombing, the Depth of Surface-Opening Cracks, and Ungrouted Ducts.* Concrete International, 1988. **10**(4): pp. 38-46.

[9] Bungey, J.H., *Sub-surface Radar Testing of Concrete: A Review.* Construction and Building Materials, 2004. **18**(1): pp. 1-8.

[10] Li, C.Q., et al., *Concrete Delamination Caused by Steel Reinforcement Corrosion.* Journal of Materials in Civil Engineering, 2007. **19**(7): pp. 591-600.

[11] ASTM-C4580, *Standard Practice for Measuring Delaminations in Concrete Bridge Decks by Sounding*. 2003, ASTM Internatinal, West Conshohocker, PA.

[12] Sansalone, M. and Carino, N.J., *Detecting Delaminations in Concrete Slabs with and without Overlays using the Impact-Echo Method.* ACI Materials Journal, 1989. **86**(2): pp. 175-184.

[13] Warhus, J.P., Mast, J.E., and Nelson, S.D. *Imaging Radar for Bridge Deck Inspection*. Proceedings of *SPIE*. 1995.

[14] Sansalone, M., *Impact-Echo: The Complete Story.* ACI Structural Journal, 1997. **94**(6): pp. 777-786.

[15] Ata, N., Mihara, S., and Ohtsu, M., *Imaging of Ungrouted Tendon Ducts in Prestressed Concrete by Improved SIBIE.* NDT and E International, 2007. **40**(3): pp. 258-264.

[16] Yehia;, S., et al., *Detection of Common Defects in Concrete Bridge Decks using Nondestructive Evaluation Techiniques.* Journal of Bridge Engineering, 2007. **12**(2): pp. 215-224.

[17] McCann, D.M. and Forde, M.C., *Review of NDT Methods in the Assessment of Concrete and Masonry Structures.* NDT & E International, 2001. **34**(2): pp. 71-84.

[18] Zhu, J., *Non-contact NDT of Concrete Structures using Air-Coupled Sensors*, in *Department of Civil and Environmental Engineering*. 2006, University of Illinois at Urbana-Champaign: Urbana-Champaign, IL.

[19] Ottosen, N.S., Ristinmaa, M., and Davis, A.G., *Theoretical Interpretation of Impulse Response Tests of Embedded Concrete Structures.* Journal of Engineering Mechanics, 2004. **130**(9): pp. 1062-1071.

[20] Martin, J., et al., *Ultrasonic Tomography of Grouted Duct Post-Tensioned Reinforced Concrete Bridge Beams.* NDT & E International, 2001. **34**(2): pp. 107-113.

[21] Malhotra, V.M. and Carino, N.J., *Handbook on Nondestructive Testing of Concrete*. 2nd ed. 1991, CRC Press, Boca Raton, FL.

[22] Bungey, J.H. and Millard, S.G., *Radar Inspection of Structures.* ICE PROCEEDINGS, STRUCTURES AND BUILDINGS, 1993. **99**(2): pp. 173-186.

[23] Clemena, G.G. and McKeel, W.T., *Detection of Delamination in Bridge Decks with Infrared Thermography.* Transportation Research Record, 1978(664): pp. 180-182.

[24] Brink, A., et al. *Application of Quantitative Impulse Thermography for Structural Evaluation in Civil Engineering - Comparison of Experimental Results and Numerical Simulations*. Proceedings of *Quantitative Infrared Thermography*. 2002. Croatia.

[25] Clark, M.R., McCann, D.M., and Forde, M.C., *Application of Infrared Thermography to the Non-Destructive Testing of Concrete and Masonry Bridges.* NDT and E International, 2003. **36**(4): pp. 265-275.

[26] Masad, E., et al., *Computations of Particle Surface Characteristics using Optical and X-Ray CT Images.* Computational Materials Science, 2005. **34**: pp. 406–424.

[27] Jandhyala, V.K. and Dasgupta, N., *Characterization of Air Void Distribution in Asphalt Mixes using X-Ray Computed Tomography.* Journal of materials in civil engineering, 2002. **14**: pp. 122.

[28] Zelelew, H.M., Papagiannakis, A.T., and Masad, E. *Application of Digital Image Processing Techniques for Asphalt Concrete Mixture Images*. Proceedings of *the 12th International Association for Computer Methods and Advances in Geomechanics*. 2008. Goa, India.

[29] MDOT, *Instruction Manual for Bridge Deck Delamination Detector*. 1977, Michigan Department of Transportation: Lansing, MI.

[30] Henderson, M.E., Dion, G.N., and Costley, R.D. *Acoustic Inspection of Concrete Bridge Decks*. Proceedings of *SPIE*. 1999. Newport Beach, CA.

[31] Vincent, E., Gribonval, R., and Fevotte, C., *Performance Measurement in Blind Audio Source Separation.* IEEE Transactions on Audio, Speech, and Language Processing, 2006. **14**(4): pp. 1462-1469.

[32] Boll, S., *Suppression of Acoustic Noise in Speech using Spectral Subtraction.* IEEE Transactions on Acoustics, Speech and Signal Processing, 1979. **27**(2): pp. 113-120.

[33] Widrow, B. and Hoff, M., *Adaptive Switching Circuits.* IRE WESCON Convention Record, 1960: pp. 96-104.

[34] Comon, P., *Independent Component Analysis: A New Concept?* Signal Processing, 1994. **36**: pp. 287-314.

[35] Haykin, S., *Neural Networks: A Comprehensive Foundation*. 2nd ed. 1999, Prentice Hall, Englewood Cliffs, NJ.

[36] Amari, S., Cichocki, A., and Yang, H.H., *A New Learning Algorithm for Blind Signal Separation.* Advances in neural information processing systems, 1996.

[37] Koldovsky, Z., Tichavsky, P., and Oja, E., *Efficient Variant of Algorithm FastICA for Independent Component Analysis Attaining the Cramér-Rao Lower Bound.* IEEE Transactions on Neural Networks, 2006. **17**(5): pp. 1265-1277.

[38] Kay, S.M., *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. 1993, Prentice Hall.

[39] Koldovsk, Z. and Tichavsk, P. *Time-Domain Blind Audio Source Separation using Advanced ICA Methods*. Proceedings of *The 8th Annual Conference of the International Speech Communication Association*. 2007.

[40] Thomas, J., Deville, Y., and Hosseini, S., *Time-Domain Fast Fixed-Point Algorithms for Convolutive ICA.* IEEE Signal Processing Letters, 2006. **13**(4): pp. 228 - 231

[41] Cherkassky, V. and Mulier, F.M., *Learning from Data: Concepts, Theory and Methods*. 1998, John Wiley & Sons Inc., Hoboken, NJ.

[42] John, G.H., Kohavi, R., and Pfleger, K. *Irrelevant Features and the Subset Selection Problem*. Proceedings of *11th International Confrence on Machine Learning*. 1994.

[43] Debnath, L., *Wavelet Transforms and Their Applications*. 2002, Springer.

[44] MATLAB. 2009, The MathWorks Inc. , Natick, MA.

[45] http://en.wikipedia.org/wiki/Wavelet_packet_decomposition; Accessed on Jan. 20, 2010.

[46] Wicker, E.Z. and Fastl, H., *Psychoacoustic: Facts and Models*. 1990, Springer Berlin, Germany.

[47] Zheng, F., Zhang, G., and Song, Z., *Comparison of Different Implementations of MFCC.* Journal of Computer Science and Technology, 2001. **16**(6): pp. 582-589.

[48] Lee, S.M., et al., *Improved MFCC Feature Extraction by PCA-Optimized Filter Bank for Speech Recognition.* IEEE workshop on Automatic Speech Recognition and Understanding, 2001: pp. 49-52.

[49] Lee, J.H., et al. *Speech Feature Extraction using Independent Component Analysis*. Proceedings of *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2000.

[50] Torkkola, K., *Feature Extraction by Non Parametric Mutual Information Maximization.* The Journal of Machine Learning Research, 2003. **3**(7): pp. 1415-1438.

[51] Kay, S.M., *Fundamentals of Statistical Signal Processing, Volumn II: Detection theory*. 1998, Prentice-Hall, Englewood Cliffs, NJ.

[52] Rice, J.A., *Mathematical Statistics and Data Analysis*. 2007, Duxbury Press.

[53] Moon, T.K., *The Expectation-Maximization Algorithm.* IEEE Signal Processing Magazine, 1996. **13**(6): pp. 47-60.

[54] Theodoridis, S. and Koutroumbas, K., *Pattern Recognition*. 2nd ed. 2003, Academic Press, San Diego, CA.

[55] Bertsekas, D.P., et al., *Nonlinear Programming*. 1995, Athena Scientific, Belmont, MA.

[56] Tikhonov, A.N., *Solution of Incorrectly Formulated Problems and the Regularization Method.* Soviet Math., 1963. **4**: pp. 1035-1038.