Time-Frequency Based Adaptive Learning for Structural Health Management

by

Debejyo Chakraborty

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

August 2010

Time-Frequency Based Adaptive Learning for Structural Health Management

by

Debejyo Chakraborty

has been approved

July 2010

Graduate Supervisory Committee:

Antonia Papandreou-Suppappola, Chair
Aditi Chattopadhyay
Chaitali Chakrabarti
Bahar Jalali-Farahani

ACCEPTED BY THE GRADUATE COLLEGE

ABSTRACT

In the last few decades, several statistical signal processing techniques have been used to deal with uncertainty when detecting, identifying and classifying structural damage. One of the key challenges in integrating signal processing techniques in real-world structural health management systems is how to incorporate diversity in the damage state and variability in environmental and operational conditions. While conventional learning methods are adequate for characterizing the underlying mechanism of damage nucleation and evolution, they are of limited use in highly complex and rapidly changing environments. This is especially the case when the amount of available data is insufficient.

In this dissertation, time-frequency based methods and hidden Markov model based methods are used for the detection and classification of structural damage. Time-frequency techniques are also used to extract damage features; these techniques are chosen as they are well-matched to the time-varying spectral characteristics of waveforms measured using sensors on the structures. The proposed methodologies are adaptively learned by allowing the stochastic models to continuously evolve from experience with the time-varying environment. Damage-related features extracted from periodically-buffered structural data are modeled using Dirichlet process mixture models that provide for a growing number of mixture components or damage classes. Coupled with input from physics-based models in a Bayesian filtering framework, the adaptively-identified classes can be traced to different types of damage. An active data selection technique is used to optimize the adaptive identification of damage classes. The proposed adaptive learning methodology is baseline-free in the sense that it does not require any a priori damage training.

A novel information-transfer learning methodology is also proposed that reuses parameters that are learned from similar previous experiments. An inductive transfer mechanism is considered to aid damage classification when the available training data is statistically insufficient.

To

my parents, Sulagna & Debashis Chakraborty

for their wonderful parenting, guidance, support and care,

and

my wife, Bhavana Chakraborty

for her love and persistant encouragement.

ACKNOWLEDGMENTS

I extend my thanks and gratitude to my advisor, Prof. Antonia Papandreou-Suppappola for her continued guidance, support and mentoring. I am also very grateful for the mentoring of Dr. Narayan Kovvali and his invaluable collaboration.

I would like to thank Prof. Aditi Chattopadhyay, Prof. Pedro Peralta, Dr. Jun Wei, Dr. Sunilkumar Soni and Dr. Subhasish Mohanty from the School of Mechanical, Aerospace, Chemical and Materials Engineering at Arizona State University for providing me with Lamb wave data from bolted joint and lug samples, as well as the finite-element analysis model that I used in this work.

I would like to also thank Prof. James Spicer and Lindsey Channels from the Department of Materials Science and Engineering, at John Hopkins University for providing me with ultrasonic data and M. Derriso, M. DeSimio and S. Olson (AFRL, Wright Patterson AFB) for providing me with fastener failure data from a square aluminum plate.

I am thankful to my committee memebers, Prof. Chaitali Chakrabarti, Prof. Bahar Jalali-Farahani, Prof. Aditi Chattopadhyay and Prof. Andreas Spanias for their invaluable comments and constructive criticism.

I thank my family for their support and inspiration.

This work was supported by the Department of Defense Air Force Office of Scientific Research Grant FA95550-06-1-0309 (former program manager: Dr. Victor Giurgiutiu and current program manager: Dr. David S. Stargel).

TABLE OF CONTENTS

Page

CHAPTER Page

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

Introduction

1.1. Structural Health Monitoring

Structural health monitoring (SHM) is an important new technology for civil, mechanical and aerospace structures, that has been receiving significant attention in the scientific community in the past decade. In [1–3], SHM has been defined as the process of implementing a strategy for observing, detecting, identifying, processing, and predicting damage on a structure. SHM requires *in situ* sensing or non-destructive evaluation (NDE), together with detection, learning, identification, and classification of characteristic features to indicate the presence of structural damage or degradation. The presence of damage can adversely affect the current and future performance of a structural system [1]. Thus, the goal of SHM is to identify, locate, and quantify damage for state-awareness of key structural components in order to ensure the users safety and the system reliability.

In the past, SHM was confined to civil or mechanical/aerospace engineering, where qualitative visual inspection and time-based maintenance procedures were performed. Recently, interdisciplinary interest in SHM has let to improved statistical signal processing and identification techniques that enable damage quantification and structural residual life estimation. A considerable amount of literature, in the areas of mechanical, material, and structural engineering, and signal processing and statistical analysis [1, 2, 4], exists on methodologies for determining the presence, type, location, and intensity of structural damage. Some examples include statistical methods [5,6], time-series analysis [7], statistical pattern recognition [8], impedance-based methods [9], Fourier component pair analysis [10], Bayesian methods [11, 12], extreme value statistics [13], support vector machines [14], and other advanced time-frequency based signal processing methodologies [15–30].

Most of the aforementioned SHM methods rely on a training/testing paradigm of damage classification. Specifically, known types of damage are first studied on which the developed technique is trained. The training process can be expensive as it normally requires a substantial amount of data to properly address a damage with statistical significance. This is needed so that the algorithm is robust when used to identify damages for which it was trained in a real application. Note, also, that currently no methodologies have been established that can adapt an SHM algorithm to material variability, temperature, or other parameters that govern operating conditions. In addition, damage in a real scenario is unpredictable and there may not be a comprehensive training procedure available to account for all possible types of damage.

1.2. Processing of Signals for Structural Health Monitoring

Signal processing methods have always been an integral part of SHM. Many prevalent SHM techniques rely on vibration analysis [2]. The vibration signal acquired from a structure can be studied in the time domain, frequency domain, and also in the joint time-frequency domain to obtain information on the health condition of a structure.

In vibration analysis, the material under investigation is excited by an induced vibration sequence. The response of the material to such a vibration is a function of the material boundary condition, internal grain orientation and environment/operating conditions. This response could be altered when a damage is present or when there is a change in the operating environment. In addition, the vibrational response can be affected by ambient noise sources. For example, a rolling of the bearing shaft in a health monitoring system for bearings [31] can introduce significant noise.

The active sensor wave propagation technique [32] is a relatively new SHM method.

Elastic waves propagating through a structure can collect information on a damage present in the structure. The physical properties of the waves can facilitate monitoring over a large coverage area, thus a smart placement of a few sensors can monitor the entire structure under observation. The response of the waves received at different locations is different and governed by the relative position of the sensor. Figure 1.1 demonstrates this phenomenon.

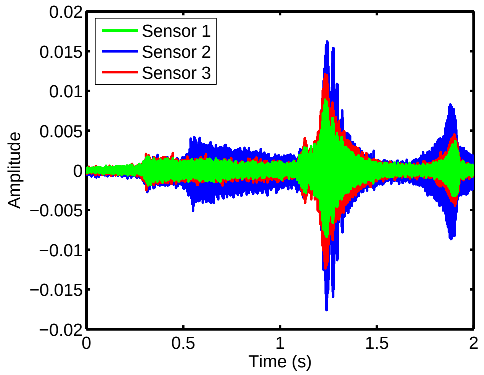


Fig. 1.1. Study of the response of a linear frequency-modulated chirp through a metal plate with a loose bolt as received by three different sensors from different parts of the plate.

This figure demonstrates the responses from three different sensors that are mounted on a metal plate and are recording the response to an excitation linear frequency-modulated chirp signal. The responses differ in envelope, amplitude and spectral content.

The approach of interpreting a wave signature for damage identification in [32, 33] was based on mathematical modeling. Specifically, the amplitude and time history of the elastic wave scattering and refraction were modeled to characterize different damage states. Though this technique provides *in situ* SHM, modeling for complex material structures is non-trivial. For example, a mathematical model of the dynamics of the plate needs to be developed and validated before the theoretically-predicted wave propagation could be used for damage identification. The authors had to thoroughly investigate different types of waves, including axial waves, shear waves, flexure waves, Raleigh waves, and Lamb waves, in order to obtain the theoretical solution for an elastic wave propagating through the material.

Instead of modeling the wave response, it is more feasible to conduct experiments in controlled conditions in order to study the response of elastic waves (like Lamb waves) under various damage scenarios. Specifically, if a response elastic wave signal could be decomposed into a smaller finite number of units or it could be represented using a finite set of features, then the decomposition could be used to reconstruct the original wave. More importantly, this finite set of units or features could be used to represent a specific type of damage.

Since structural materials are dispersive media, signals propagating through them exhibit time-dependent spectral variation. Joint time-frequency domain methods like the wavelet transform [15, 20, 34–38], short-time Fourier transform [39, 40], and quadratic time-frequency representations [21, 41] (TFRs) are some of the techniques that can be used to provide time-dependent spectral information. In addition, they can be used to study selective bands of frequencies. Wavelet analysis has been used for damage detection [15] since the late nineties. Note, however, that wavelet analysis does not yield high time or frequency resolution over all desirable regions in the time-frequency plane. Also, quadratic TFRs can yield interfering cross terms when the signal under analysis has multiple components [41].

Although these methods often provide reliable SHM, they still require training and testing. Thus, their performance is limited by the amount of data that is available and cannot be used to identify a type of damage that has not been encountered before. In fact, these methods work reliably only on the specific material on which they have been trained. In addition, there is no guarantee that they will perform reliably in changing environmental conditions or with varying test samples.

1.3. Proposed Matched Time-frequency Processing for Structural Health Monitoring

The matching pursuit decomposition (MPD) algorithm [42] and its corresponding MPD TFR are successful analysis tools that can be used to decompose a signal into a linear combination of Gaussian atom components that are time-shifted, frequency-shifted, and scaled to cover the entire time-frequency plane. These features can then be modeled and directly used for classification. Another option is to use the linear expansion in a two-dimensional (2-D) classification scheme in the time-frequency plane [43, 44]. Due to the time-varying nature of the signals, the 2-D classifier can be shown to outperform classical time-domain or frequency-domain correlation classifiers.

Using the MPD algorithm, we proposed a method to utilize joint time-frequency analysis to discriminate between data from various damage conditions [21–26, 45, 46]. Specifically, we considered not only Gaussian atoms for the decomposition but also time-frequency shifted versions of real measured data from sensors (modified MPD or MMPD) in order to better match different types of damage [21]. We also employed a Bayesian sensor decision fusion approach to help improve classification performance by combining the information collected by multiple distributed sensors. We demonstrated the success of our approach using real data from a fastener failure damage in an aluminum plate.

1.4. Proposed Adaptive Learning Processing for SHM

We considered a probabilistic classification framework using Gaussian mixture models to represent different types of damage or different environment conditions, where the mixture models are formed using MPD time-frequency features. We proposed an adaptive learning methodology, where stochastic Gaussian mixture models are allowed to continuously evolve from experience with the time-varying environment. The adaptive learning framework is based on the use of Dirichlet process (DP) mixture models [47–51] to provide the modeling with the machinery needed to self-adapt to structure within the data. DP priors can be used in applications that require modeling data with countably infinite mixtures and for estimating the mixture parameters [48–50, 52–58]. With the development of Markov chain Monte Carlo (MCMC) methods [59–61], sampling from the posterior of such countably infinite model posteriors is now computationally feasible. Note, however, that the use of MCMC makes it difficult to have an online, real time estimation procedure as it requires re-learning if the data changes and it can be computationally intensive.

In order to enable an online real time process, we developed an algorithm that combines Bayesian filtering with an iterative adaptive learning approach. This enables the use of the MCMC technique with DP priors to perform online damage classification. As a result, current environment/working conditions are concurrently monitored and integrated into the algorithm to reduce false alarms due to such changes. We have also integrated physics-based progressive damage models into this method in order to avoid training preprocessing and have a baseline-free algorithm. The notable advantages of this method are thus that no baseline training data is required and signals can be classified on the fly to new (previously unseen) damage classes, yielding an adaptive and effective approach for online SHM.

We have also proposed an information transfer mechanism over related SHM experiments through the formulation of translated transfer learning [62].

Depending on the type and availability of labeled data (e.g., from training or from other experiments), there are various types of transfer learning (TL). In multi-task inductive transfer learning (TL) [63–66], the source and target domains contain labeled data and the learning task is geared toward simultaneously improving the classification in both domains. In self taught inductive TL [67], the source domain data is not labeled. In translated inductive TL [68], there are labeled data in the source domain and statistically insufficient labeled data in the target domain. These frameworks can be used to solve regression, classification, and cross-domain learning tasks. In transductive TL [69] only source domain labeled data is available. It is used generally to solve tasks that involve domain adaptation and covariate shift, regression and classification. In unsupervised TL [70–72] no labeled data is available. It is used to solve clustering and dimensionality reduction problems.

In this dissertation, the translated learning framework is proposed for the problem of classification for damage classes in fatigue damage monitoring. Data is available from four sensors but training data is limited and sufficiently available from only one sensor (source domain). The other sensors (target domains) have some labeled data but that is statistically insufficient. Our results demonstrate the dramatic improvement in classification in the target domain sensors when TL is used, when compared to the performance without TL.

1.5. Dissertation Organization

The remainder of the dissertation is organized as follows. In Chapter 2, some signal processing methods are discussed that are prevalent in SHM applications. In Chapter 3, we review the MPD algorithm and we propose its use for damage classification in SHM. Bayesian sensor fusion is also formulated that combines decisions from multiple sensors, and classification performance is demonstrated using real data.

In Chapter 4, we investigate HMM damage classification, and we propose the use of the variational Bayes method to adaptively select the model complexity is used to determine the model complexity of the HMMs that would best describe the data. We test the classifier using real data and with controlled noise added to synthetic damage data.

In Chapter 5, we provide a background on adaptive learning methodologies, and we propose its application, combined with physics-based damage models, for progressive damage estimation in order to obtain an adaptive learning damage classification approach that does not require any a priori training.

In Chapter 6, we extend our study to a transfer learning mechanism which enables us to reuse information learned from similar and related training experiments. Our proposed work and future directions are summarized in Chapter 7.

1.6. List of Acronyms Used in Dissertation

The acronyms used throughout the dissertation are summarized next.

DP	Dirichlet process
DPMM	Dirichlet process mixture model
EM	expectation-maximization
FEA	finite element analysis
FEM	finite element model(ing)
FIFO	first-in-first-out
FT	Fourier transform
GMM	Gaussian mixture model
HMM	hidden Markov model

KLD	Kullback-Leibler distance
LFM	linear frequency-modulated
MAP	maximum a posteriori
MCMC	Markov chain Monte Carlo
ML	maximum-likelihood
MMPD	modified matching pursuit decomposition
MPD	matching pursuit decomposition
MPD-PDF	matching pursuit decomposition based probability density function
MPD-TFR	matching pursuit decomposition based time-frequency representation
pdf	probability density function
PSD	power spectral density
PZT	piezoelectric transducer
ROC	receiver operating characteristic
SEM	scanning electron microscopy
SHM	structural health monitoring
SNR	signal-to-noise ratio
STFT	short-time Fourier transform
TF	time-frequency
TFR	time-frequency representation
WD	Wigner distribution

VB	variational Bayes
VBEM	variational Bayes expectation-maximization

1.7. Notations

Listed next are the symbols used throughout the dissertation.

A	state transition matrix for hidden Markov
A'	transpose of matrix A
B	state dependent observation density for HMM
C	confusion matrix
c	indicator variable (or class membership)
$c_i^{\text{tar,te}}$	class membership of ith test data from target domain
$c_i^{\text{sor,te}}$	class membership of ith test data from source domain
$\mathbf{c}_{\text{tar}}^{\text{tr}}$	set of class memberships of all training data in target domain
$\mathbf{c}_{\text{sor}}^{\text{tr}}$	set of class memberships of all training data in source domain
$\mathbf{c}_{\text{tar}}^{\text{te}}$	set of class memberships of all testing data in target domain
$\mathbf{c}_{\text{sor}}^{\text{te}}$	set of class memberships of all testing data in source domain
c_{comp}	computational cost
c_{pre}	pre-computational cost
D	dimensionality (eg: 1-D is one-dimensional)
$\mathbb{D}$	MPD dictionary
E_s	energy of time-domain signal $s(t)$

$\mathscr{E}_s(t, f)$ .... matching pursuit decomposition based time-frequency representation of signal $s(t)$

$\mathcal{E}$ .... 1-norm of error between a truncated DP model and a true infinite Dirichlet process model

$\mathcal{F}(\cdot)$ .... variational objective function

G_0 or $G_0(\Theta)$ .... base distribution for Dirichlet process

G or $G(\Theta)$ .... Dirichlet distribution draw from $DP(\alpha, G_0)$

$\mathcal{G}$ .... probability of data distributed according to a Gaussian mixture model

$\mathcal{H}$ .... state model for crack estimation in compact tension sample

$g(t)$ .... time domain representation of an matching pursuit decomposition dictionary atom

$\mathbb{H}$ .... representative subset of matching pursuit decomposition used in statistical measure of similarity computation

$\mathbb{I}$ .... acquired set of matching pursuit decomposition used in statistical measure of similarity computation

K .... number of sensors

K_{obs} .... number of (discrete) observations in a discrete hidden Markov model

L .... length of Markov chain

$L_{\mathbb{H}}$ .... number of elements in the set $\mathbb{H}$

L_B .... length of Markov chain burn-in

L_T .... length of observation sequence

M	DP truncation limit or number of Gaussian components in Gaussian mixture model
$\mathbb{M}$	set of models
$\mathbb{M}_{\mathbf{HMM}}$	set of hidden Markov models
M_c	number of damage classes
N	matching pursuit decomposition truncation limit, also, length of each observation sequence
N_p	number of processing cores
N_s	number of samples in (sampled) signal $s(t)$
N_{tr}	number of training signals
N_w	number of signals
N_Θ	the number of occurrences of Θ
N_x	number of discrete states
p	weights in a probability density mixture model
p^*	probability measure in the stick breaking prior
$\mathbf{p}^*$	a set of $\{p_1^*, \ldots, p_\infty^*\}$
$\mathbb{P}^{\text{ref}}$	set of reference matching pursuit decomposition based probability density functions
$\mathbb{P}^{\text{te}}$	set of test matching pursuit decomposition based probability density functions
$r(t)$	time representation of matching pursuit decomposition residue

$\mathfrak{r}$	inner product (projection)
$s(t)$, $u(t)$	continuous-time signals
$s(n)$	discrete-time series signals
S	Number of hidden states for the hidden Markov models
$\mathbb{S}$	set of signals
$\mathbb{S}^{\text{ref}}$	set of reference signals
T	total number of observed signals
$\hat{T}$	number of selected features
$\mathbb{U}$	probability space for a Dirichlet process
$u_\mu, u_\tau, u_\Lambda, u_\nu, u_a, u_b$	prior distribution hyperparameters
$\mathfrak{u}_\mu, \mathfrak{u}_\tau, \mathfrak{u}_\Lambda, \mathfrak{u}_\nu, \mathfrak{u}_a, \mathfrak{u}_b$	posterior distribution hyperparameters
$\mathbf{x}$	state (hidden) associated to an observation sequence
x	states
$x^{(k)}$	state at epoch k
$x^{(1:k)}$	state at all epochs from 1 to k
$\mathbf{y}$	observations vector
Y	observation sequence $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots]$
$Y^{\text{sor,tr}}$	source domain training data (observation sequence)
$Y^{\text{sor,te}}$	source domain training data (observation sequence)
$Y^{\text{tar,tr}}$	target domain training data (observation sequence)

$Y^{\text{tar,te}}$	target domain training data (observation sequence)
$\mathbb{Y}$	set of observation sequences, $\mathbb{Y} = \{Y_1, Y_2, \ldots, \}$
$\mathbb{Y}^{\text{tr}}_{\text{sor}}$	set of source domain training data (observation sequence)
$\mathbb{Y}^{\text{te}}_{\text{sor}}$	set of source domain testing data (observation sequence)
$\mathbb{Y}^{\text{tr}}_{\text{tar}}$	set of target domain training data (observation sequence)
$\mathbb{Y}^{\text{te}}_{\text{tar}}$	set of target domain testing data (observation sequence)
$\mathbb{Y}^{\text{FIFO}}$	set of features in the first-in-first-out buffer
$\mathbb{Y}^{\text{ref}}$	set of reference features selected from the first-in-first-out buffer
u_a, u_b, u_μ, u_τ	hyperparameters of normal-gamma distribution
α	matching pursuit decomposition expansion coefficient
χ	log-normal distributed random variable
$\Delta\chi$	discretization level in the distribution of χ
δ	Kronecker delta
ι	Dirichlet process innovation parameter
κ	time scaling
Λ_{N_s}	hidden Markov model with N_s states
λ	eigenvalue
μ	mean of a pdf
$\boldsymbol{\mu}$	set $\{\mu_1\, \mu_2\, \ldots\, \mu_T\}$
ν	frequency shift

Φ_k	environmental/operating conditions at kth epoch
$\boldsymbol{\Phi}_k$	environmental/operating conditions from 0, ..., kth epoch
$\boldsymbol{\pi}$	initial state distribution for hidden Markov model
ρ	probability of move in Metropolis-Hastings
σ	standard deviation probability density function
Θ	set of all model parameters that need to be estimated
$\boldsymbol{\theta}$	set $\{\Theta_1 \ldots \Theta_T\}$
$\boldsymbol{\theta}_{-i}$	set $\{\Theta_1 \ldots \Theta_{i-1}\, \Theta_{i+1} \ldots \Theta_T\}$
$\boldsymbol{\theta}^*$	set of of unique Θs: $\{\Theta_1^* \ldots \Theta_M^*\}$
$\boldsymbol{\theta}_m^{\mathrm{sor}}$	source domain parameter set for mth class
$\boldsymbol{\theta}_m^{\mathrm{tar}}$	target domain parameter set for mth class
τ	time shift
ζ	a probability density function
$Beta\,(\cdot,\cdot)$	Beta distribution
$Log\text{-}\mathcal{N}\,(\cdot,\cdot)$	log-normal distribution
$\overline{Log\text{-}\mathcal{N}}\,(\cdot,\cdot,\cdot)$	discrete approximation to log-normal distribution
$Mult\,(\cdot)$	a multinomial distribution
$Neg\text{-}Bin\,(\cdot,\cdot)$	negative binomial distribution
$\mathcal{N}\,(\cdot,\cdot)$	normal distribution
$DP\,(\iota, G_0)$	Dirichlet process with innovation parameter ι and base distribution G_0

$P\left(\cdot\right)$	continuous probability density function (or likelihood function)
$\mathrm{P}\left(\cdot\right)$	almost surely discrete probability density function (or likelihood function)
$\Pr\left(\cdot\right)$	discrete probability density function (or likelihood function)
$\mathrm{WD}_s(t,f)$	Wigner distribution of signal $s(t)$
$\left\Vert s\right\Vert_2$	L_2 norm of signal $s(t)$
$\langle\cdot,\cdot\rangle$	inner product (or projection)
$\vert i\vert$	absolute value of any number i.
$\vert\mathbb{Y}\vert$	cardinality of a set $\mathbb{Y}$
$\mathcal{O}(\cdot)$	computational complexity order
$\mathbf{E}\left[\cdot\right]$	expectation
$\mathbf{Var}\left[\cdot\right]$	variance
$\rightarrow$	shows progression
$\Leftrightarrow$	shows dependency/association

CHAPTER 2

Signal Processing Methods in Structural Health Monitoring

Different methods have been proposed in the literature to process signals that provide information on a structural damage. Next, we provide a short review of some of these methods with corresponding references. Note that this is not a representative review of all existing signal processing methods in SHM, and there is a multitude of references that we could not include due to space limitations. A comprehensive review of SHM methods can be found in [73, 74].

2.1. Time Domain Analysis

Using time domain analysis [7,75,76], the location of damage can be found by analyzing acceleration time histories recorded from the structure of interest. This is combined with autoregressive (AR) and autoregressive with exogenous input (ARX) parameters that are constructed from a selected reference signal. The AR coefficients of the observed signal are investigated and the closeness of these coefficients (using the Euclidean distance measure to one of the reference classes) classifies the data as belonging to that class.

2.2. Acoustic Emission

The acoustic emission (AE) technique is one of the oldest non-destructive evaluation techniques. Generally, two kinds of received signals are considered in practice: a burst (finite duration wave) and a continuous signal. A burst is the result of a local effect and is separable in time (has a starting and ending time) whereas a continuous signal is associated with mechanical and/or electrical disturbances; it can also be generated from multipath or multiple reflections from the geometric boundaries of the material. The representative signal parameters include [2]: peak amplitude, arrival time, duration time, rise time and ring down count. Electrical transducers like piezoelectric, electrodynamic, laser and capacitance sensors are used to measure these signals.

2.3. Ultrasonics

The pulse-echo technique uses ultrasound transducers [2,77–79]. Ultrasound waves are transmitted across the transverse section of the material and the echo is received. This inspection is performed using two modes: (a) beam inspection, which corresponds to the direct reflection of the wave at a normal angle of incidence, and (b) beam angle inspection, that introduces refracted shear waves. Only highly localized effects can be detected using the ultrasonic technique, and it requires prior knowledge of the expected damage location.

2.4. Kurtosis

The crest factor analysis or kurtosis [18, 80, 81] is a measure of the peakedness of a signal. The crest factor is defined as the ratio of the peak value to the root-mean-squared (RMS) value of a signal. For a discrete-time signal $s(n)$, $n = 1, \ldots, N_s$, the crest factor is defined as the ratio of the L_∞ norm to the L_2 norm of the signal. Thus, it is given by

$$\text{crest factor} \equiv \sqrt{N_s}\frac{\|s\|_\infty}{\|s\|_2},$$

where

$$\begin{aligned} \|s\|_2 &= \sum_{n=1}^{N_s} |s(n)|^2 \\ \|s\|_\infty &= \max_{n \in \{1,\ldots,N_s\}} |s(n)|. \end{aligned}$$

The crest factor tells us how much the signal has deviated from its maximum and it is a form of time domain analysis.

2.5. Cepstrum analysis

The Fourier transform of the decibel spectrum of a signal is called the cepstrum [2,82], and it provides the periodicity of the signal in the Fourier domain. For a signal $s(n)$, the cepstrum is defined as:

$$s_c(n) = |\mathbf{IFT}\{\log_{10} |\mathbf{FT}\{s(n)\}|^2\}|^2$$

where **FT** and **IFT** are the discrete Fourier and inverse discrete Fourier transforms [83], respectively. A healthy material is expected to echo the excitation sequence but a damaged material is expected to generate new frequency components unknown to the excitation sequence. This can disrupt the periodicity in the spectrum.

2.6. Guided Waves

Unlike bulk waves, guided waves [2,84] have infinite modes associated with their wave propagation. They have highly sensitive propagation properties that make them interact with material defects. There are various types of guided waves available including Rayleigh waves (surface waves) and Lamb waves.

Lamb waves are electric perturbations propagating in a solid plate with free boundaries for which the displacements correspond to different basic propagation modes with symmetric and antisymmetric vibrations [2,43]. For a given thickness of a plate t_h and frequency of wave propagation f_w, there exists a finite number of propagation modes (loci of phase velocity) given by the product $f_w t_h$. Joints, stiffeners, rivets, and materials with varying thickness affect the pattern of the Lamb waves and introduce more modes. By studying the pattern of the Lamb wave propagation, it is possible to detect the location of a defect.

2.7. Electromagnetic Impedance

The electromagnetic impedance technique is electrical in nature and thus the signal is not a vibration signal but it is an impedance signal [32, 33]. The piezoelectric transducer (PZT) crystal (eg. Lead Zirconate Titanate) is mounted on a circular plate and then its joint impedance characteristics are modeled. Excitation waves from frequency bands 10-40 kHz, 10-150 kHz, and 300-450 kHz are used. The PZT active sensor impedance is measured and the resonant peaks are studied. This is done for the healthy signals first to validate the model. In damaged materials, a shift in these resonant frequencies are observed that is inversely proportional to the distance of the damage from the PZT sensor.

2.8. Fourier Transform Component Pair

The Fourier transform has been extensively used for damage detection by studying the spectral components of the signal [2, 10]. Damage is indicated by the shift in resonance (natural) frequency of a material. In addition, a more pronounced nonlinearity could also be observed from spectral studies of damaged structures when compared to healthy ones. It was observed that analyzing the real and imaginary parts of the Fourier transform resulted in better detection than when the magnitude was used.

For time-varying signals, the study of the entire spectrum would not reveal any temporal changes. However, the short-time Fourier transform can provide the instantaneous time spectrum of a signal. In this approach, the signal is windowed such that the window segments appear to have a fixed frequency over the duration of the window. Then the Fourier transform of the segments can provide a joint time-frequency representation of the signal.

2.9. Wavelet Transform

The discrete wavelet transform can be considered as a time-frequency decomposition of a signal into a linear combination of scaled wavelet functions. Near the damage location, the wavelet coefficients exhibit a sharp change and the magnitude of their local extrema quantifies the damage. This analysis could also be used together with Kurtosis [18] to detect damage. In [15, 16, 36], the authors showed that signals from a damaged structure have a different decomposition than those corresponding to a healthy structure. The wavelet decomposition result could be used to classify damage due to its multiresolution property. As the wavelet transform can yield different levels of resolution for different frequencies, it is ideal for detecting abrupt signal changes [37, 38].

2.10. Correlation

Correlation is a mathematical operation used in classification to estimate how similar an observed signal is to a signal that belongs to a particular class. For example, to compute the level of similarity between two signals, $s^{(1)}(t)$ and $s^{(2)}(t)$, we correlate them using

$$\gamma(\tau) = \int_{\infty}^{\infty} s^{(1)}(t)s^{(2)}(t-\tau)dt$$

where τ is a time shift or lag parameter.

Various possibilities of correlation can be considered based on the information we are trying to obtain.

- *Real data with model data* — Correlation between real data and model data can be used to validate the model. Also, once the model is validated, data from the model can be correlated with the new real data to classify the new data as belonging to the model class.

- *Real data with training data* — Each class is defined by a set of training signals which are known to belong to that class based on prior knowledge. The new incoming data, called the test data, can be correlated to the data belonging to a class. Based on the correlation, the test data will be assigned to the class it most likely belongs to.

- *Real data with multiple sensor responses* — Correlation can be used for damage detection [44]. Correlating the responses from all the sensors can reduce the effect of noise and identify the signal corresponding to damage.

CHAPTER 3

Time-Frequency Damage Classification

3.1. Matching Pursuit Decomposition based Feature Extraction

The matching pursuit decomposition (MPD) algorithm [42,85] is an iterative processing method that expands a signal into a weighted linear combination of elementary basis functions or "atoms" chosen from a complete dictionary. The resulting expansion of a finite energy signal $s(t)$ is given by

$$s(t) = \sum_{i=0}^{\infty} \alpha_i \, g_i(t),$$

where $g_i(t)$ is the basis function selected from the MPD dictionary $\mathbb{D}$ at the ith MPD iteration and α_i is the corresponding expansion coefficient. The expansion is such that the energy E_s of the signal is preserved. In particular, if the basis functions $g_i(t)$ are normalized to unit energy, then it can be shown that

$$E_s = (\|s\|_2)^2 \triangleq \int_{-\infty}^{\infty} s(t)\, s^*(t) dt = \sum_{i=0}^{\infty} |\alpha_i|^2 \,.$$

After N iterations, the resulting expansion can be shown to converge in the L_2 sense, i.e.,

$$\lim_{N \to \infty} \left\| s(t) - \sum_{i=0}^{N-1} \alpha_i \, g_i(t) \right\|_2 = 0,$$

In practical applications, the MPD signal expansion is thus given by

$$s(t) = \sum_{i=0}^{N-1} \alpha_i \, g_i(t) + r_N(t) \,, \qquad (3.1)$$

where $r_N(t)$ is the residual signal after N MPD iterations such that

$$(\|s\|_2)^2 = \left(\sum_{i=0}^{N-1} |\alpha_i|^2 \right) + (\|r_N\|_2)^2 \,.$$

The steps of the MPD iterative algorithm are given as follows [42,85]. With $r_0(t) = s(t)$, at the ith iteration, $i = 0, 1, \ldots, N-1$, the projection of the residue $r_i(t)$ onto every dictionary element $g^{(d)}(t) \in \mathbb{D}$ is computed to obtain

$$\mathfrak{r}_i^{(d)} = \langle r_i, g^{(d)} \rangle \triangleq \int_{-\infty}^{\infty} r_i(t)\, g^{(d)*}(t)\, dt\,.$$

The selected dictionary atom $g_i(t)$ is the one that maximizes the magnitude of the projection,

$$g_i(t) = \underset{g^{(d)}(t) \in \mathbb{D}}{\operatorname{argmax}} \left| \mathfrak{r}_i^{(d)} \right| . \tag{3.2}$$

The corresponding expansion coefficient is

$$\alpha_i = \langle r_i, g_i \rangle = \int_{-\infty}^{\infty} r_i(t)\, g_i(t)\, dt\,. \tag{3.3}$$

The residues at the ith and $(i+1)$th iterations are related as $r_{i+1}(t) = r_i(t) - \alpha_i\, g_i(t)$. Thus, after N MPD iterations, the residue is given by

$$r_N(t) = r_{N-1}(t) - \alpha_{N-1}\, g_{N-1}(t) = s(t) - \sum_{i=0}^{N-1} \alpha_i\, g_i(t)\,.$$

It can be shown that the residue energy progressively decreases as the number of iterations increases, i.e., $(\|s\|_2)^2 = (\|r_0\|_2)^2 \geq (\|r_1\|_2)^2 \geq \ldots \geq (\|r_N\|_2)^2 \geq 0$.

The truncation limit N is usually chosen such that the energy of the residue after N iterations is smaller than some pre-defined value. Note that, by design, the MPD yields a compact representation of a signal in terms of selected basis functions in the dictionary. In addition, it effectively filters out unwanted signal components such as noise because the noise subspace is typically orthogonal to that spanned by the dictionary elements [42]. The MPD algorithm is summarized in the flowchart in Figure 3.1.

The MPD dictionary consists of a redundant set of atoms which, in general, do not have to form an orthonormal set; the dictionary is, however, required to be complete [42]. The dictionary is formed by time-frequency shifting and scaling a basic atom. In [42], the

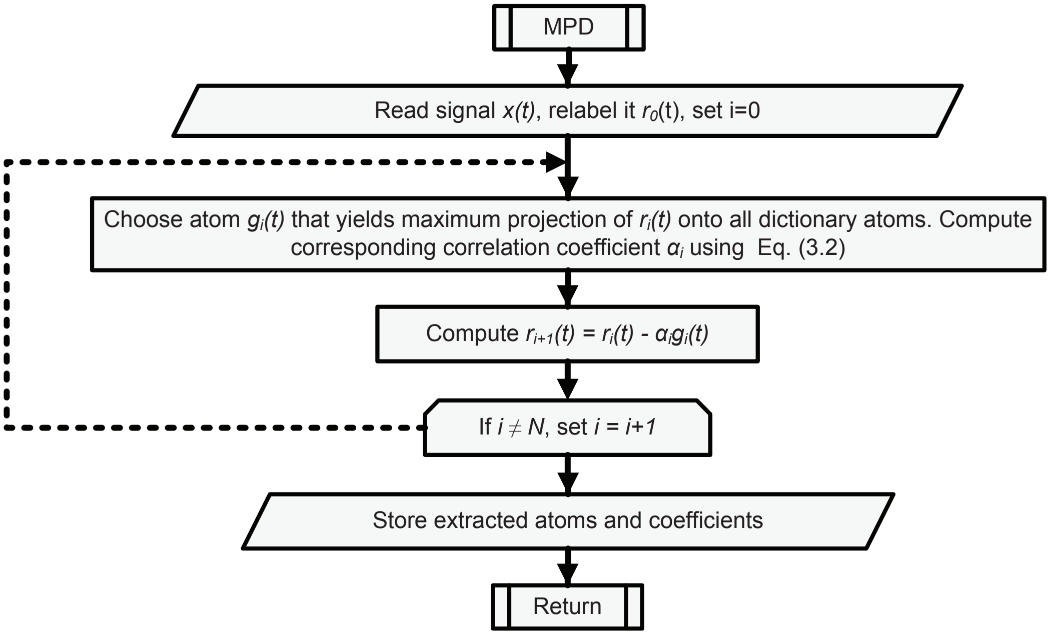

MPD
Read signal $x(t)$, relabel it r_0(t), set i=0
Choose atom $g_i(t)$ that yields maximum projection of $r_i(t)$ onto all dictionary atoms. Compute corresponding correlation coefficient α_i using Eq. (3.2)
Compute $r_{i+1}(t) = r_i(t) - \alpha_i g_i(t)$
If $i \neq N$, set $i = i+1$
Store extracted atoms and coefficients
Return

Fig. 3.1. Flowchart summarizing the steps of the MPD algorithm.

basic atom is chosen to be a Gaussian signal, given by $g(t) = (2/\pi)^{1/4}\, e^{-t^2}$. This basic atom has unit energy and is concentrated at the origin $(t, f) = (0, 0)$ in the time-frequency plane. The shifting transformations result in moving the atom in the time-frequency plane, and the scaling transformation changes the variance (or spread in time) of the Gaussian function. Specifically, the transformed basis functions that form the dictionary $\mathbb{D}$ are given by

$$g^{(d)}(t) = (2\kappa_l/\pi)^{1/4}\, e^{-\kappa_l\,(t-\tau_n)^2}\, e^{j2\pi\nu_m t}\,, \tag{3.4}$$

where $d = \{\tau_n,\, \nu_m,\, \kappa_l\}$ is the set of all time-shifted (by τ_n, $n = 1, \ldots, N_d$), frequency-shifted (by ν_m, $m = 1, \ldots, M_d$), and time-scaled (by $\kappa_l > 0$, $l = 1, \ldots, L_d$) atoms that are normalized to have unit energy. If $s(t)$ is real, then the dictionary can be formed using real unit energy Gaussian atoms, given by

$$g^{(d)}(t) = (8\kappa_l/\pi)^{1/4}\, e^{-\kappa_l\,(t-\tau_n)^2}\, \cos(2\pi\nu_m t). \tag{3.5}$$

The computational complexity of the MPD algorithm is $\mathcal{O}(N_s \log N_s)$ [42], where N_s

is the number of available samples of the signal $s(t)$. This computational cost is the result of a few facts. First, each projection of the residue signal onto the time-shifted atoms at every MPD iteration can be interpreted as a correlation that can be efficiently implemented using the fast Fourier transform (FFT) with $\mathcal{O}(N_s \log N_s)$ computational complexity. In addition, the cost of the FFT of the Gaussian atoms, needed to obtain the frequency shift in (3.4), can be reduced by analytically computing the transforms. Specifically, the Fourier transform of a frequency-shifted Gaussian atom $e^{-\kappa t^2}\, e^{j2\pi\nu t}$ is a Gaussian function in the frequency domain given by $\sqrt{\pi/\kappa}\, e^{\left(-(f-\nu)^2/(4\kappa)\right)}$. The computational cost can be further reduced by storing the frequency shifts of the Gaussian atoms ahead of time to use only when needed .

3.2. Matching Pursuit Decomposition based Time-Frequency Representation

The choice of dictionary basis functions used in the MPD algorithm depends on the application. In the present problem, the analysis signals propagate through dispersive media, and as a result, they exhibit time-varying spectral characteristics as their frequency content can change with time. Thus, they are best analyzed using time-frequency techniques that are designed to depict time-varying spectra in the time-frequency plane [86–88]. The WD is a popular TFR as it does not exhibit windowing effects when analyzing signals, thus providing high time-frequency resolution properties. For a signal $s(t)$, the WD is defined as

$$\mathrm{WD}_s(t,f) \equiv \int_{-\infty}^{\infty} s\left(t+\frac{\tau}{2}\right)\, s^*\left(t-\frac{\tau}{2}\right)\, e^{-j2\pi f\tau}\, d\tau\,.$$

The WD is well-matched to the MPD framework as the WD preserves time-frequency shifts and scale changes as in Equation (3.4). Specifically, the WD satisfies the following three properties:

$$u(t) = s(t-\tau) \quad \Rightarrow \quad \mathrm{WD}_u(t,f) = \mathrm{WD}_s(t-\tau,f)$$
$$u(t) = s(t)\, e^{j2\pi\nu t} \quad \Rightarrow \quad \mathrm{WD}_u(t,f) = \mathrm{WD}_s(t,f-\nu)$$
$$u(t) = \sqrt{|\kappa|}\, s(\kappa\, t) \quad \Rightarrow \quad \mathrm{WD}_u(t,f) = \mathrm{WD}_s(\kappa\, t, f/\kappa)\,.$$

Using these properties, the WD of the Gaussian function $g(t)$ is related to the WD of the transformed Gaussian atom $g^{(d)}(t)$ as $\mathrm{WD}_{g^{(d)}}(t,f) = \mathrm{WD}_g\left(\kappa_l(t-\tau_n),(f-\nu_m)/\kappa_l\right)$. Also, the WD of the Gaussian atoms $g^{(d)}(t)$ in Equation (3.4) can be computed analytically [42] as

$$\mathrm{WD}_{g^{(d)}}(t,f) = 2\exp\left(-2\kappa_l(t-\tau_n)^2\right)\exp\left(-\frac{2\pi^2(f-\nu_m)^2}{\kappa_l}\right). \tag{3.6}$$

As Gaussian signals are the most concentrated signals in both time and frequency according to the uncertainty principle [87], their WD is highly localized, in addition to being easy to compute using the closed-form expression in Equation (3.6).

The main drawback of the WD TFR is the presence of cross-terms when used to analyze signals with multiple time-frequency components due to its quadratic nature. Essentially, if a signal consists of multiple components, the WD of the signal is not only the sum of the WDs of the components but also the cross WD between any two components [87]. In order to avoid cross-terms, the MPD-TFR is defined as the sum of the weighted WDs of each of the selected atoms [42]

$$\mathscr{E}_s(t,f) \equiv \sum_{i=0}^{N-1} |\alpha_i|^2\, \mathrm{WD}_{g_i}(t,f)\ . \tag{3.7}$$

Since $\mathrm{WD}_{g_i}(t,f)$ is the WD of the selected atom $g_i(t)$ in the ith MPD iteration, and this atom consists of a single Gaussian, the MPD-TFR in Equation (3.7) is free of cross-terms.

An example of the sum of the WD of $N = 4$ time-frequency transformed Gaussian atoms is shown in Figure 3.2. The corresponding time-frequency shifts and scale changes of the four atoms are marked on the time-frequency plot in Figure 3.2.

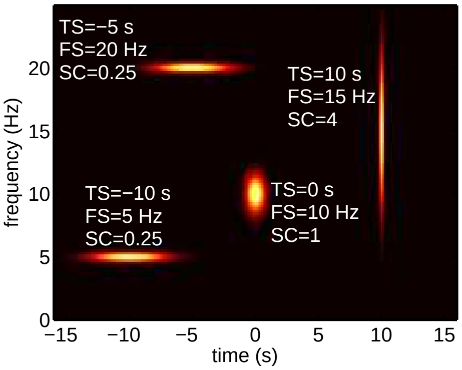


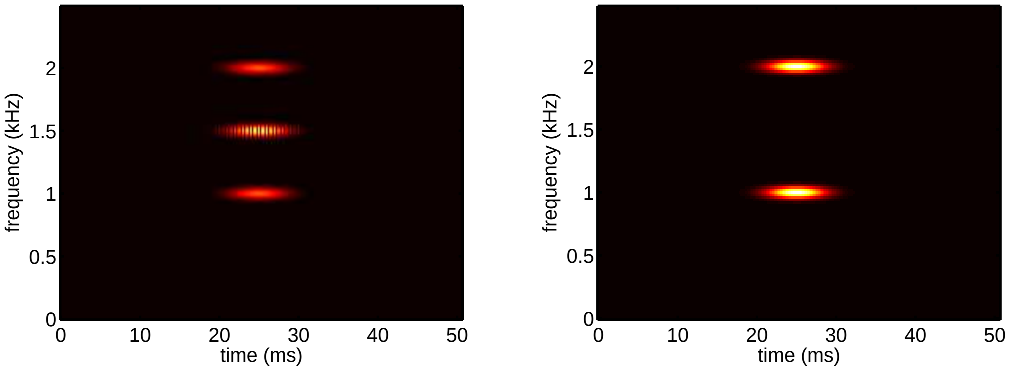
Fig. 3.2. Sum of the WDs of $N = 4$ time-frequency shifted and scaled Gaussian atoms from an MPD dictionary. Here, TS denotes time shift, FS is frequency shift and SC is scale change.

3.2.1. Demonstration of the MPD algorithm and MPD-TFR

The use of the MPD algorithm in decomposing a signal and the corresponding MPD-TFR of the signal is demonstrated with a simple example. A simulated signal is considered that consists of the sum of two Gaussian functions

$$s(t) = e^{-40000\,(t-0.025)^2} e^{j2\pi 1000t} + e^{-40000\,(t-0.025)^2} e^{j2\pi 2000t}\,. \tag{3.8}$$

The first Gaussian function is time-shifted by $\tau = 25$ ms, frequency-shifted by $\nu = 1$ kHz, and scaled by $\kappa = 40,000$. The second Gaussian function is time-shifted and scaled by the same amounts as the first Gaussian, but frequency-shifted by $\nu = 2$ kHz. Figure 3.3(a) shows the WD of $x(t)$, where the cross-term between the two auto WDs (corresponding to the WD of each single Gaussian component) can be seen halfway between the two auto terms at $(t, f) = (25$ ms, 1.5 kHz$)$. When the signal in Equation (3.8) is decomposed using the MPD, as the MPD dictionary does contain these two Gaussian atoms, the algorithm converges after only $N = 2$ iterations. The extracted atoms correspond to exactly the two signal components and the residue is identically zero. Note that, in general, this will not be true for most real signals; it was designed to be so here, in order to emphasize the high time-frequency localization and no-cross-term properties of the MPD-TFR. The resulting MPD-TFR in Equation (3.7) is shown in Figure 3.3(b) without any cross-terms.



(a) Wigner distribution.

(b) MPD-TFR.

Fig. 3.3. Comparison of the Wigner distribution and the MPD-TFR of the signal in Equation (3.8).

3.3. Matching Pursuit Decomposition based Damage Classification

3.3.1. Classification Using MPD

The SHM problem of analyzing and classifying measured signals from structures at different levels of damage conditions (referred to as damage classes) is considered. As the MPD algorithm decomposes a signal into highly-localized basis functions, the MPD-TFR can be used for damage classification by exploiting the time-frequency features extracted by the MPD [89, 90]. There are two main steps in this classification algorithm. The first step is to train the MPD classifier using a representative set of training signals from damaged classes of interest. The second step is to classify the MPD extracted features of a test signal using the MPD extracted features of the training signals in the time-frequency plane.

Specifically, the MPDs of training signals from different damage classes are first computed, and the corresponding MPD-TFRs of the training signals are used to construct template TFRs that represent the time-frequency characteristics of each damage class. Given M_c classes corresponding to signals from M different types of damages, a training signal for class m, $m = 1, \ldots, M_c$, is denoted as $u_{l,m}(t)$, where $l = 1, \ldots, N_{\mathrm{tr},m}$ and $N_{\mathrm{tr},m}$ is the number of training signals used for class m. The MPD-TFR for each training signal is computed as

$$\mathscr{E}_{u_{l,m}}(t,f) = \sum_{i=0}^{N-1} |\alpha_{i,m,l}|^2 \, \mathrm{WD}_{g_{i,m,l}}(t,f), \tag{3.9}$$

where $g_{i,m,l}(t)$ and $\alpha_{i,m,l}$ are the selected Gaussian atom and expansion coefficient in the ith MPD iteration for the lth training signal in the mth class. The template TFRs for each class are then computed by averaging the MPD-TFRs of all the training signals in that class as

$$\mathscr{E}_{u_m}(t,f) = \frac{1}{N_{\mathrm{tr},m}} \sum_{l=1}^{N_{\mathrm{tr},m}} \mathscr{E}_{u_{l,m}}(t,f)\,. \tag{3.10}$$

Classification is performed using two-dimensional (2-D) correlations between template TFRs and the MPD-TFR of the test signal. The strength of the correlations in the time-frequency plane is used to quantify how similar or dissimilar the given signal is to known members of each class. Specifically, given a test signal $s(t)$, its MPD-TFR, $\mathscr{E}_s(t,f)$, is first obtained. Then, $s(t)$ is assigned to Class m^* if

$$m^* = \underset{m=1,\ldots,M_c}{\operatorname{argmax}} \left\{ \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathscr{E}_s(t,f) \mathscr{E}^*_{u_m}(t,f)\, dt\, df \right\}. \tag{3.11}$$

Note that the MPD-TFRs are normalized to unit 1-norm before computing the correlations (equivalent to an 2-norm signal normalization [42]). The flowchart in Figure 3.4 summarizes the MPD classification technique.

Note that the larger the number of measurements available for training, the more statistically accurate are the template TFRs, and consequently, the more robust the classifier. In practice, as many training signals should be obtained as possible. Also, in addition to training on multiple signals from the same experiment, it is also desirable to train on measurements from different experiments for the same type of damage and boundary conditions in order to make the classifier more robust. The computational complexity of the MPD classifier can be reduced by writing the relevant MPD-TFRs in terms of their component WDs, and then using the fact that the 2-D Gaussian functions (corresponding to the WDs of each selected Gaussian atom) are uncorrelated in time and frequency (see Equation (3.6)). The 2-D integrals in Equation (3.11) can therefore be factored into one-dimensional time and frequency integrals of Gaussian functions, which can be computed efficiently.

3.3.2. Modified MPD Classifier

In the MPD damage classification algorithm discussed in Section 3.3.1, the dictionary is composed of time-frequency shifted and scaled versions of an elementary Gaussian atom. Since the WD of a Gaussian signal is known in closed-form (see Equation (3.6)), the use of this dictionary can reduce the MPD computational cost. If a signal, however, is not sparse in the time-frequency plane, then a very large number of Gaussian functions will be required in order to accurately represent it.

In the modified MPD (MMPD) algorithm [89], the dictionary is composed of time-frequency shifted signals from real sensor measurements. The MMPD-based damage classi-

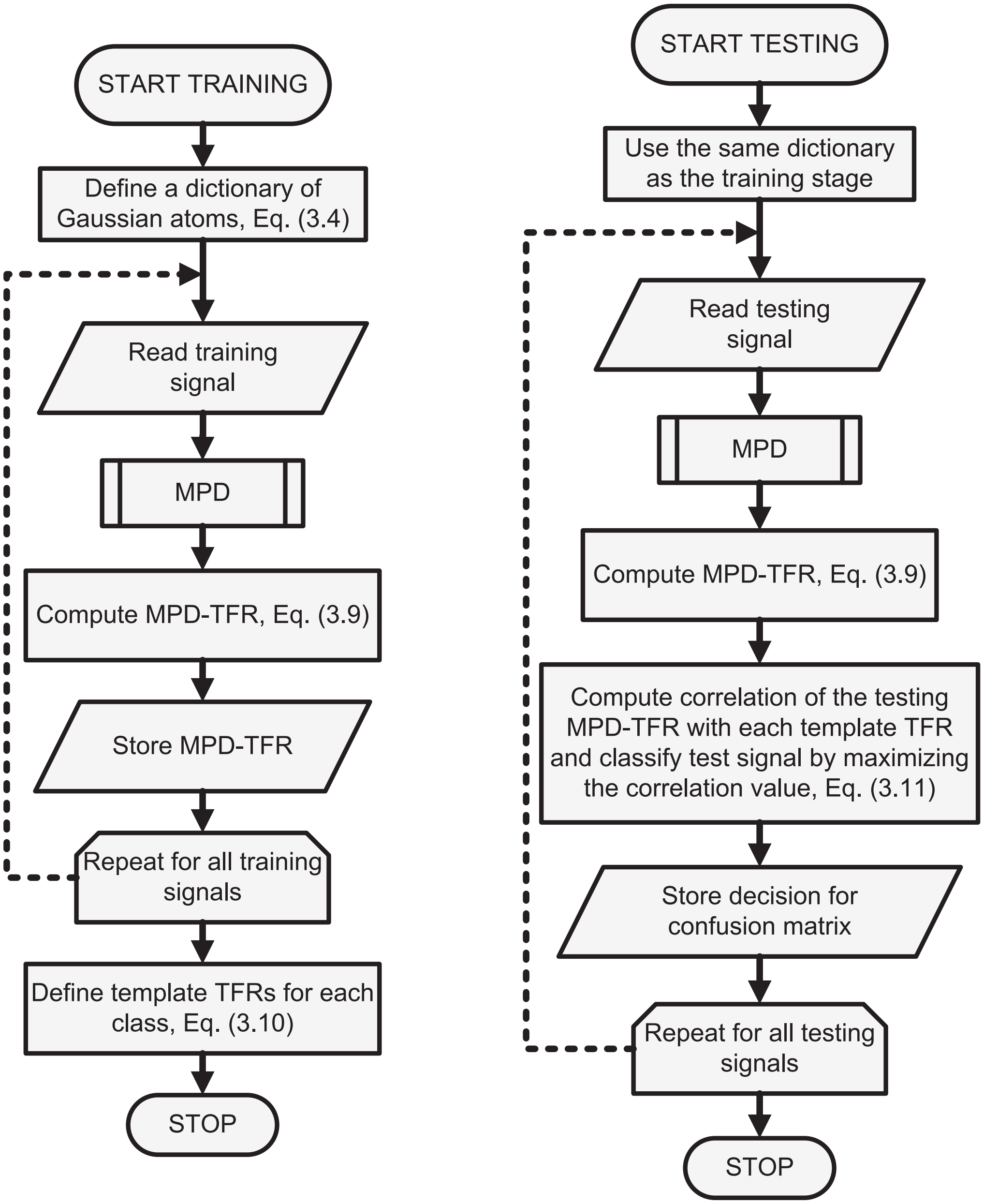

START TRAINING
Define a dictionary of Gaussian atoms, Eq. (3.4)
Read training signal
MPD
Compute MPD-TFR, Eq. (3.9)
Store MPD-TFR
Repeat for all training signals
Define template TFRs for each class, Eq. (3.10)
STOP
START TESTING
Use the same dictionary as the training stage
Read testing signal
MPD
Compute MPD-TFR, Eq. (3.9)
Compute correlation of the testing MPD-TFR with each template TFR and classify test signal by maximizing the correlation value, Eq. (3.11)
Store decision for confusion matrix
Repeat for all testing signals
STOP

Fig. 3.4. Flowchart summarizing a classification algorithm based on MPD features.

fier uses the MPD with sensor data obtained directly from structural damage experiments as the dictionary atoms [90]. As this dictionary is matched specifically to the signals of interest in an application, it has the advantage of yielding highly parsimonious representations.

The MMPD is performed using the MPD steps described in Section 3.1 but with the dictionary atoms composed of multiple real signals from each of the M_c damage classes, with each signal time and frequency shifted to cover the entire time-frequency plane. The dictionary atoms are now denoted as $g_m^{(d)}(t)$, $m = 1, \ldots, M_c$, to emphasize that time-frequency shifted versions of representative signals from each class need to be included in the dictionary. At the ith MPD iteration, $i = 0, 1, \ldots, N-1$, for a signal $s(t)$, the projection of the residue $r_i(t)$ (with $r_0(t) = s(t)$) onto each dictionary element is computed as

$$\mathfrak{r}_{m,i}^{(d)} = \langle r_i, g_m^{(d)} \rangle \,.$$

The atom that maximizes the projection is denoted as

$$g_{m,i}(t) = \underset{g_m^{(d)}(t) \in \mathbb{D}}{\operatorname{argmax}} \left\{ |\mathfrak{r}_{m,i}^{(d)}| \right\} \,,$$

and is known to belong to class m with corresponding expansion coefficient $\alpha_{m,i}$. The decomposed signal can be approximated as

$$s(t) \approx \sum_{i=0}^{N-1} \alpha_{\bar{m},i} \, g_{\bar{m},i}(t) \,, \tag{3.12}$$

where $\bar{m} \in \{1, \ldots, M_c\}$ could be any one of the M_c possible classes.

With the MMPD, the MPD-TFR is not adequate for classification. This is because the WD of an extracted atom could suffer from inner interference terms as the atom (corresponding to a real signal) may have nonlinear time-frequency characteristics [88]. In this framework, the classification of a given test signal $s(t)$ is not performed in the time-frequency plane. Instead, the magnitude of the projection of the signal onto each damage class becomes the class deciding factor. The projection $\mathfrak{r}^{(m)}_{\text{MMPD},s}$ of the signal $s(t)$ onto the mth damage class, $m = 1, \ldots M_c$, is computed as

$$\begin{aligned}
\mathfrak{r}^{(m)}_{\text{MMPD},s} &= \sum_{l=1}^{N_{\text{tr},m}} \left[\langle \sum_{i=0}^{N-1} \alpha_{\bar{m},i}\, g_{\bar{m},i}\,,\, g^{(d)}_{m,l} \rangle \right] = \sum_{l=1}^{N_{\text{tr},m}} \left[\sum_{i=0}^{N-1} \langle \alpha_{\bar{m},i}\, g_{\bar{m},i},\; g^{(d)}_{m,l} \rangle \right] \\
&= \sum_{l=1}^{N_{\text{tr},m}} \left[\sum_{i=0}^{N-1} \alpha_{\bar{m},i} \langle g_{\bar{m},i},\; g^{(d)}_{m,l} \rangle \right] , \qquad (3.13)
\end{aligned}$$

where $\bar{m} \in \{1,\, 2,\, \ldots,\, M\}$ from the decomposition of $s(t)$ in Equation (3.12) and $g^{(d)}_{m,l}(t)$ is the dictionary atom corresponding to the time-frequency shifted version of the lth training signal in the mth class. Furthermore, it can be shown that

$$\langle g_{\bar{m},i}\,,\, g^{(d)}_{m,l} \rangle = \begin{cases} \alpha_{m,i}, & \bar{m} = m \\ \beta_{m,i}, & \bar{m} \neq m \end{cases} , \qquad (3.14)$$

where $\beta_{m,i}$ is any number that is less than $\alpha_{m,i}$. Equation Equation (3.14) follows from the fact that signals within a class tend to be more correlated. In order to save computation, $\beta_{m,i}$ in Equation (3.14) can be set to zero. Thus, using this result, Equation (3.13) simplifies to

$$\mathfrak{r}^{(m)}_{\text{MMPD},s} = \sum_{i=0}^{N-1} |\alpha_{\bar{m},i}|^2\; \delta[\bar{m} - m] \,, \qquad (3.15)$$

where $\delta[\bar{m} - m] = 1$ if $\bar{m} = m$ and 0 if $\bar{m} \neq m$.

The resulting classifier assigns test signal $s(t)$ to class m^* if

$$m^* = \underset{m=1,\ldots,M_c}{\text{argmax}} \left\{ \mathfrak{r}^{(m)}_{\text{MMPD},s} \right\} . \qquad (3.16)$$

The MMPD based classification scheme is summarized in the flowchart in Figure 3.5.

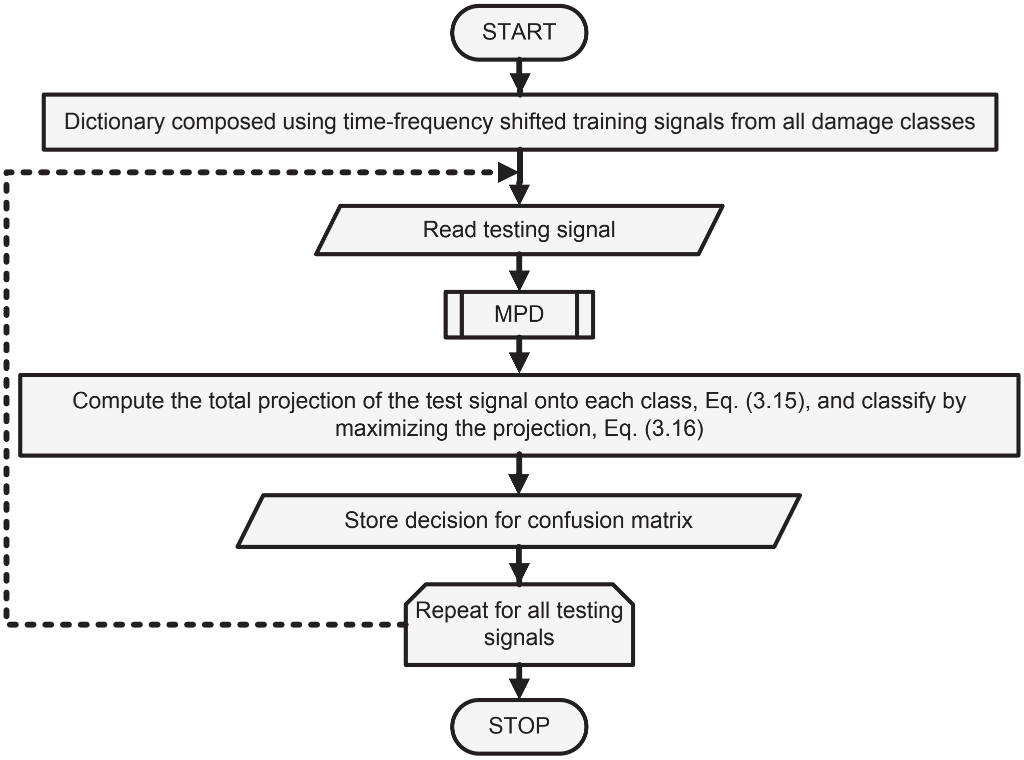

START
Dictionary composed using time-frequency shifted training signals from all damage classes
Read testing signal
MPD
Compute the total projection of the test signal onto each class, Eq. (3.15), and classify by maximizing the projection, Eq. (3.16)
Store decision for confusion matrix
Repeat for all testing signals
STOP

Fig. 3.5. Flowchart summarizing a classification algorithm based on MMPD features.

3.4. Time-frequency based Classification of Structural Data using Lamb Wave Sensing

3.4.1. Experimental Setup and Data Collection

The type of damage addressed in this paper is fastener failure in a square aluminum plate. The experiments for data collection were conducted at the Advanced Structural Concepts Branch, Air force Research Laboratory. The test article is a 12 inch by 12 inch by 0.204 inch aluminum plate [44], fastened at the corners with four bolts as shown by the schematic plot in Figure 3.6. There are four surface mounted PZTs of 0.25 inch diameter. PZT1 in Figure 3.6 is the actuator; PZT2, PZT3 and PZT4 are used as receivers. The orientation is symmetric. Any change in the structural condition (damage) reflects upon the vibrational response of the plate and is encoded by the sensor responses.

The excitation signal used in this experiment was a 0-1.5 kHz linear frequency-

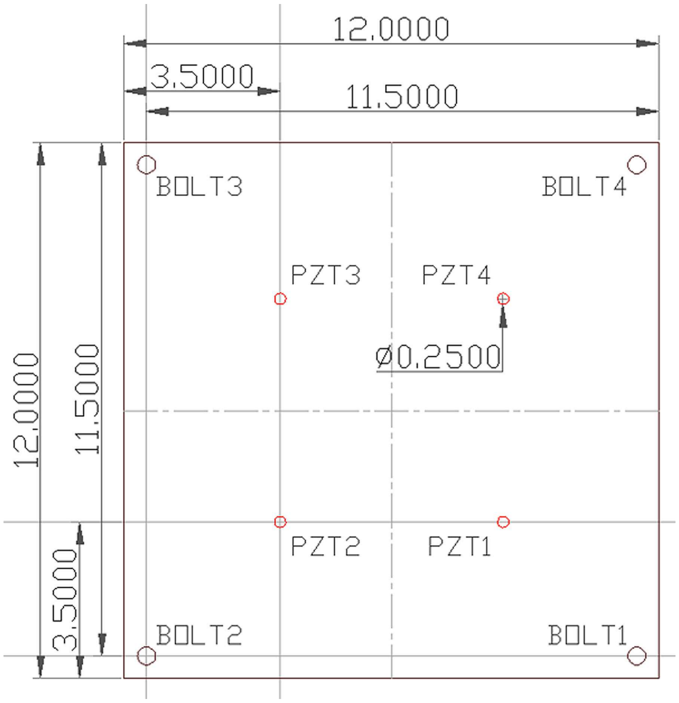


Fig. 3.6. Experimental setup for fastener damage in a square aluminum plate [44] (all dimensions are in inches).

modulated (LFM) chirp that induced normal structural vibrations. The MPD and MMPD based classifiers are not restricted to this type of excitation [91, 92]. Broadband excitation was not used because of experimental and computational difficulties. Specifically, the PZT sensors used had a narrowband response, and high sampling rates and large dictionaries would be needed in the MPD to accommodate broadband signals. Five different classes of structural damage are defined. Classes 1-4 of damage correspond to bolts 1-4 at 25% torque (30 inch-pound), respectively, and Class 5 (the undamaged class) has all the bolts at 100% torque (120 inch-pound). Responses were collected from the PZTs from multiple measurements, and 400 signals were collected from each PZT and for each damage class. Class 5 had 1,600 signals collected at each PZT. A total of 9,600 signals were used for classification from all three PZT sensors. The sampling frequency at the transmitter and receiver was 5 kHz. Figure 3.7 shows example time-domain plots of the signal transmitted by PZT1 and the signals received by PZT3 for the five classes.

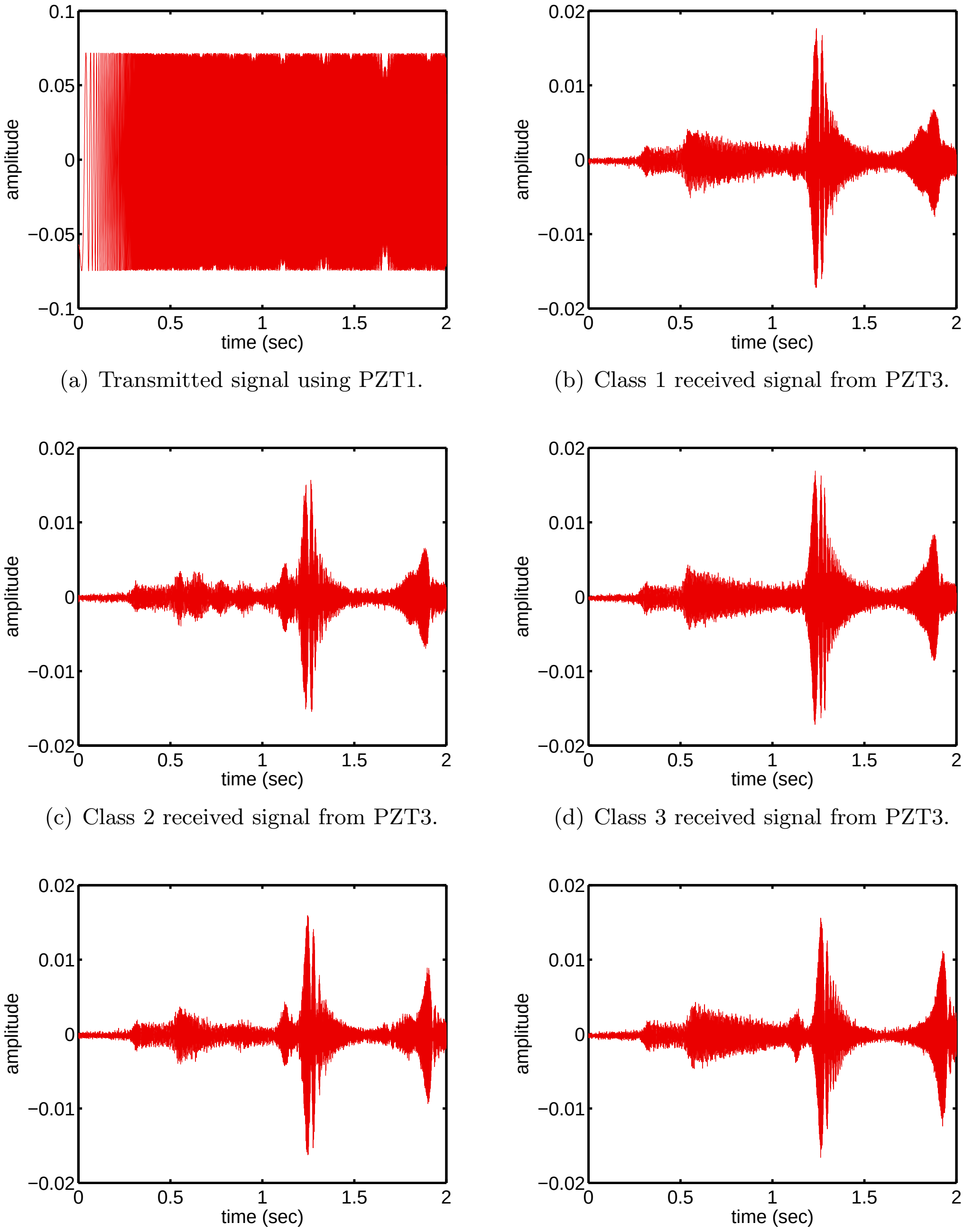
3.4.2. Data Preprocessing and MPD

Preprocessing of the received signals and feature extraction helps obtain better classification results. In some cases, however, some loss of useful information is unavoidable. The measured signals were filtered using a (unit-energy) lowpass Butterworth filter that was shifted to the signal mean value to remove the 0 Hz component. This ensures that all the signals have the same energy for correct damage classification.

The MPD was performed with $N = 60$ iterations using a dictionary of real unit-energy Gaussian atoms that are time-shifted, frequency-shifted, and scaled versions of an elementary Gaussian atom. This choice of N corresponds to a residue energy of less than 10% of the original signal energy. Note that it is not necessary to use the same value of N for all signals. However, N was fixed in order for each signal to have in an equal number of feature vectors (corresponding to expansion coefficients, time-frequency shift values and scale change values).

Using time shifts of 2 s duration and frequency shifts of 1.5 kHz bandwidth, the dictionary consisted of about 42 million atoms. Figure 3.8(b) shows an example time-domain Class 1 signal (from PZT3) after its decomposition using $N = 60$ MPD iterations. Figure 3.8(a) shows the energy of the residue at each MPD iteration. The residue energy decreases at each successive iteration and reaches about 5% of the original signal energy in 60 iterations.

The MPD-TFRs of signals in Figure 3.7(b)-(f) are shown in Figures 3.9(b)-(f). It can be seen that the locations of the atoms with high energy correspond to positions of high



(a) Transmitted signal using PZT1.

(b) Class 1 received signal from PZT3.

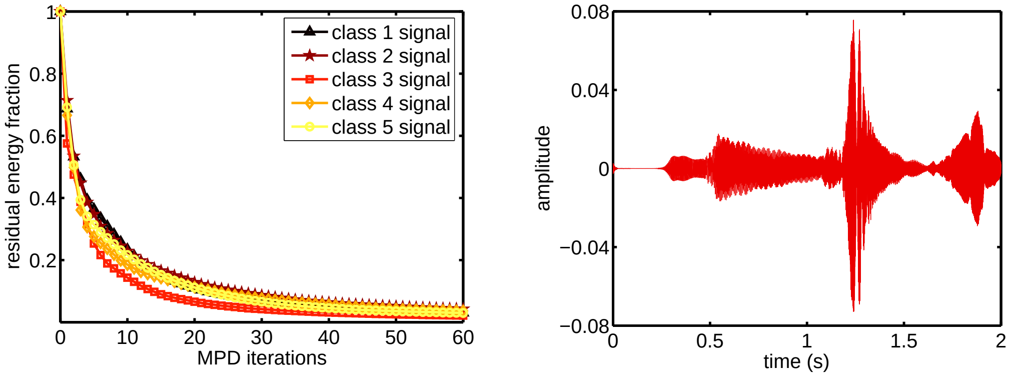
(c) Class 2 received signal from PZT3.

(d) Class 3 received signal from PZT3.

(e) Class 4 received signal from PZT3.

(f) Class 5 received signal from PZT3.

Fig. 3.7. Example time-domain plots showing the signal transmitted by PZT1 and the signals received by PZT3 for the five different classes.



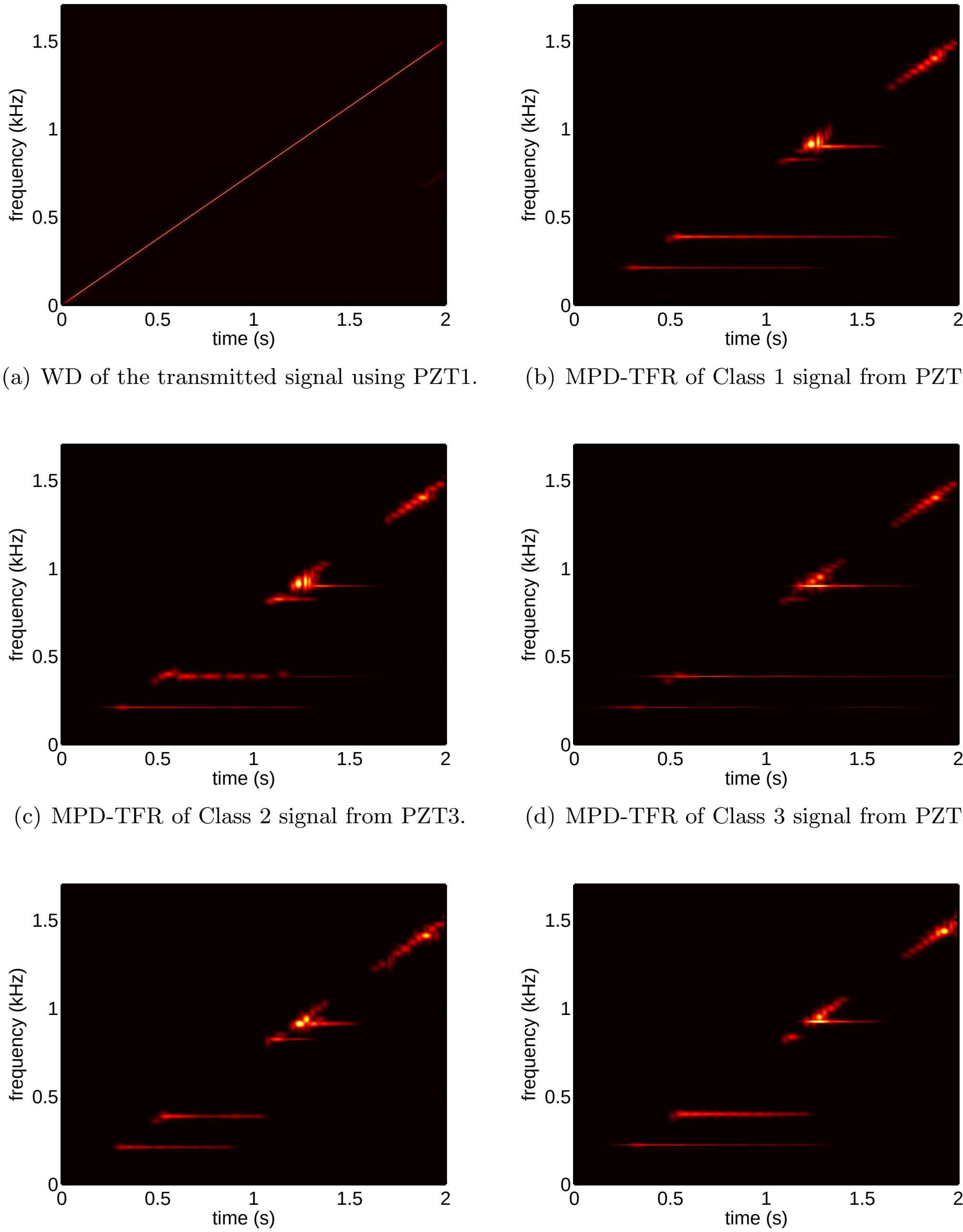
(a) Residue energy versus MPD iteration number.

(b) Class 1 signal from PZT3.

Fig. 3.8. (a) Residue energy versus MPD iteration number. (b) Time-domain PZT3 Class 1 signal after preprocessing and decomposition using the MPD with $N = 60$ iterations. This is the processed version of the signal in Figure 3.7(b).

amplitude in the signals in Figures 3.7(b)-(f).

Performing MPD with a dictionary of 42 million atoms for 10,800 signals, each of which was 10,000 samples long, is a challenging task. In order to reduce computational time, `C` programming language was used and the algorithm was implemented in parallel on a computing cluster using Message Passing Interface (MPI). The cluster consisted of seven 3 GHz dual-core Intel Pentium D processors running on 32 bit Linux (with 1 GB RAM per core). The parallelization efficiency of the MPD implementation is shown in Figure 3.10. The run-time can be modeled as a function of the number of processors using the model $c_{\text{pre}} + c_{\text{comp}}(N_w/N_p)$, where c_{pre} is the pre-computation cost common to all processors, c_{comp} is the computational cost (per signal) of the parallelizable part of the code, N_w is the total number of signals, and N_p is the number of processing cores. The small discrepancy between the actual run-times and those predicted by the model is due to the fact that this model does not account for communication costs and possible network delays.



(a) WD of the transmitted signal using PZT1.

(b) MPD-TFR of Class 1 signal from PZT3.

(c) MPD-TFR of Class 2 signal from PZT3.

(d) MPD-TFR of Class 3 signal from PZT3.

(e) MPD-TFR of Class 4 signal from PZT3.

(f) MPD-TFR of Class 5 signal from PZT3.

Fig. 3.9. Example time-frequency plots showing: (a) WD (magnitude) of the transmitted signal, and (b)-(f) the MPD-TFRs of the PZT3 signals from the five classes after preprocessing and $N = 60$ MPD iterations.

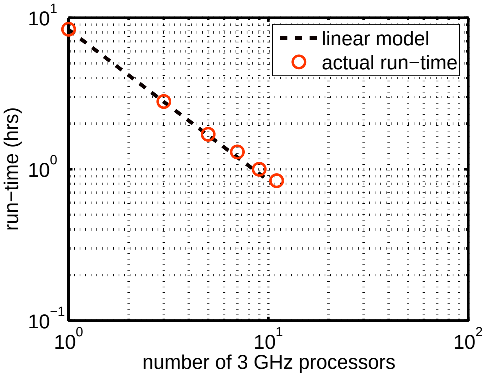


Fig. 3.10. Parallelization efficiency of the MPD implementation.

3.4.3. Optimizing the MPD-TFR Classifier

When physics-based models on structural damage are incorporated into the damage analysis and classification algorithms, the resulting performance can improve. This is demonstrated next, where prior knowledge using finite element analysis helps to improve the MPD classification performance.

Specifically, the MPD feature vectors from the various damage classes were observed to have a potential problem. The first few atoms extracted by the MPD have similar time-frequency shift and scale parameters, regardless of the actual signal classification. Since the first few MPD extracted atoms are also the highest in energy, they are the strongest contributors to the correlations used for classification. This is a problem as the classification is then effectively based on features that convey little or no discriminatory information. This is demonstrated in Figure 3.11 by superimposing the power spectral density (PSD) plots of multiple signals (from different damage classes) collected from the PZT3 sensor (solid lines) and the PSD plot of the MPD representation of one of the signals with only $N = 3$ extracted atoms (dotted line). As it can be seen, the peaks lie in the same spectral regions, showing that there is significant frequency content overlap between signals from different damage classes. Similar analysis of data from PZT2 and PZT4 sensors, which are geometrically positioned further from the actuator than the PZT3 sensor as shown in Figure 3.6, indicates that the first 6 to 7 atoms in the MPD representation are similar for signals from all five classes.

In trying to explain this phenomenon, finite element analysis (FEA) is conducted to study the effects of damage on material properties and wave propagation. This analysis, provided in Appendix A, demonstrates that the common frequencies correspond to the resonant frequencies of the plate which lie in the excitation range 0-1.5 kHz. Since the resonant frequencies affect signals from different classes in a similar way, the MPD analysis and classification benefits from simply disregarding the first few high-energy MPD extracted atoms.

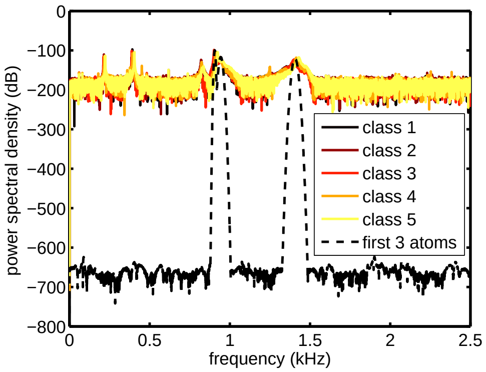


Fig. 3.11. Superimposed power spectral densities of multiple PZT3 signals from each of the five classes and of one PZT3 MPD represented signal after $N = 3$ iterations.

The identification of the MPD atoms that need to be discarded before classification can be carried out using cross-validation. This method finds the best combination of atoms to be discarded by validating the performance of the classifier on a separate data set, known as the validation set. However, this technique requires additional experimental data to form the validation set and can be expensive when the number of possible combinations of atoms is large. Instead of doing cross-validation, the finite element modeling (FEM) can be used to determine the excitation resonant frequencies of the structure under investigation. These frequencies can then be compared with the parameters of the extracted MPD atoms in order to determine which ones are common to all the classes. While finite element modeling and simulation of complex structural systems can be difficult, the method provides a reliable way of removing the redundant atoms. Details of the three-dimensional finite element modeling analysis of the aluminum plate are given in Appendix A.

3.4.4. MMPD and Dictionary Size

As discussed in Section 3.3.2, the MMPD is a parsimonious decomposition as it uses real signals to form the dictionary. In this application, only $N = 2$ MMPD iterations were needed to reduce the residue energy to 5-10% of the signal energy. This is demonstrated in Figure 3.12 that shows the fast convergence of the MMPD. As the dictionary is composed of time-frequency shifted versions of the real data, the signals to be decomposed are well-matched to the dictionary. Also, the time-frequency area of interest is more localized as it depends on the time-frequency content of the data. For example, frequency shifts from 0 to 25 Hz, in steps of 0.25 Hz, were found to be sufficient for this dictionary. For Gaussian dictionary MPDs, 42 million atoms were required in the dictionary in order to reduce the residue energy; for the real data dictionary MPD, similar energy reduction results were obtained with only 6 million dictionary atoms. The computational overhead due to lack of closed form expressions for the atoms was more than offset by the benefits of a smaller dictionary. In addition, the far fewer MMPD iterations needed to decompose the signal accurately also added to the computational advantage of the MMPD over the MPD.

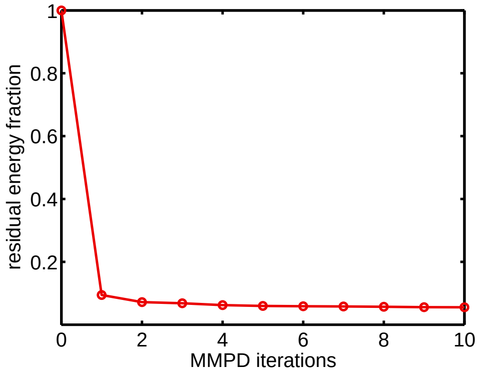


Fig. 3.12. Residue energy fraction versus MMPD iterations.

3.4.5. Classification Results

The data recorded by the PZT sensors was divided into three sets. The first set was used for training, the second set was used for cross-validation, and the third set was used for testing. The signals from PZT2, PZT3, and and PZT4 were represented using $N = 60$ MPD iterations, corresponding to an average energy residue of about 5%. In the MPD-TFR based damage classifier, the first 6, 3, and 7 MPD atoms were discarded from signals from PZT2, PZT3, and PZT4, respectively (see Section 3.4.3).

The performance of the algorithm is demonstrated using a confusion matrix, as shown on Table 3.1(a) for PZT2 data classification. The confusion matrix element $C_{i,k}$, ith row and kth column, estimates the probability that a signal actually from damage Class i is classified as belonging to damage Class k. Here, i and k correspond to the five classes of fastener damage defined earlier. For example, in Table 3.1(a), $C_{5,5} = 0.98$ because 98% of the signals from Class 5 were (correctly) classified to Class 5. Note that the entries in each row of a confusion matrix add up to unity. The mean $\frac{1}{M_c}\sum_{i=1}^{M_c} C_{i,i}$ of the diagonal elements in the confusion matrix can be used as a measure of the average correct classification rate, where M is the total number of classes.

Table 3.1 shows the confusion matrices for MPD-TFR based damage classification performed using data from PZT2, PZT3, and PZT4, without disregarding any atoms from the MPD decomposition. The average correct classification rates observed for data from PZT2, PZT3, and PZT4 are 78.4%, 60%, and 59.8%, respectively. The classification performance is thus poor, as explained by the finite element modeling analysis conducted on the test sample. Specifically, the principal modal frequencies are not altered by the induced damage, and the most dominant MPD atoms are common to all the classes. When those atoms are discarded, the classification performance improves significantly, as shown in Table 3.2. The average correct classification rates observed for data from PZT2, PZT3, and PZT4 are 87.6%, 94%, and 93.2%, respectively.

In order to demonstrate the improvement afforded by the time-frequency analysis, the classification results obtained from the MPD-TFR algorithm are compared with those from a baseline time-domain method. In this method, classification is performed directly using time-domain correlations. A representative template signal is computed for each damage class by averaging the training signals available from that class. A testing signal is classified by maximizing the time-domain correlation value of the testing signal with the template signals for each damage class. The time-domain correlation based damage classifier was tested on data from PZT4 and the results are shown in the confusion matrix shown in Table 3.3. While the correct classification rates are good for classes 1, 3, 4, and 5, Class 2 signals are completely misclassified. The average correct classification rate is 77.2%. This is in contrast to the MPD-TFR classification results on the same PZT4 data (Table 3.2(c)), where the average correct classification rate is 93.2%. This illustrates the unreliability of the time-domain classifier, and confirms the robustness advantage offered by the MPD-TFR

Class	1	2	3	4	5
1	**0.94**	0.00	0.06	0.00	0.00
2	0.58	**0.00**	0.38	0.04	0.00
3	0.00	0.00	**1.00**	0.00	0.00
4	0.00	0.00	0.00	**1.00**	0.00
5	0.00	0.00	0.00	0.02	**0.98**

(a) MPD classification results for data from PZT2.

Class	1	2	3	4	5
1	**0.00**	0.00	0.38	0.62	0.00
2	0.00	**0.00**	0.13	0.87	0.00
3	0.00	0.00	**1.00**	0.00	0.00
4	0.00	0.00	0.00	**1.00**	0.00
5	0.00	0.00	0.00	0.00	**1.00**

(b) MPD classification results for data from PZT3.

Class	1	2	3	4	5
1	**1.00**	0.00	0.00	0.00	0.00
2	0.96	**0.00**	0.00	0.04	0.00
3	1.00	0.00	**0.00**	0.00	0.00
4	0.00	0.00	0.00	**1.00**	0.00
5	0.00	0.00	0.00	0.01	**0.99**

(c) MPD classification results for data from PZT4.

Table 3.1. MPD classification results without optimization.

Class	1	2	3	4	5
1	**0.89**	0.11	0.00	0.00	0.00
2	0.10	**0.87**	0.00	0.03	0.00
3	0.00	0.38	**0.62**	0.00	0.00
4	0.00	0.00	0.00	**1.00**	0.00
5	0.00	0.00	0.00	0.00	**1.00**

(a) MPD classification results for data from PZT2.

Class	1	2	3	4	5
1	**0.77**	0.00	0.23	0.00	0.00
2	0.01	**0.96**	0.00	0.03	0.00
3	0.00	0.00	**1.00**	0.00	0.00
4	0.00	0.02	0.00	**0.97**	0.01
5	0.00	0.00	0.00	0.00	**1.00**

(b) MPD classification results for data from PZT3.

Class	1	2	3	4	5
1	**0.82**	0.09	0.06	0.03	0.00
2	0.03	**0.94**	0.00	0.03	0.00
3	0.00	0.09	**0.91**	0.00	0.00
4	0.01	0.00	0.00	**0.99**	0.00
5	0.00	0.00	0.00	0.00	**1.00**

(c) MPD classification results for data from PZT4.

Table 3.2. MPD classification results with optimization.

damage classifier.

Class	1	2	3	4	5
1	**0.96**	0.00	0.02	0.02	0.00
2	0.51	**0.00**	0.46	0.03	0.00
3	0.08	0.00	**0.91**	0.01	0.00
4	0.00	0.00	0.01	**0.99**	0.00
5	0.00	0.00	0.00	0.00	**1.00**

Table 3.3. Time-domain correlation based classification results for data from PZT4.

When the MMPD based damage classifier is used for classification, training signals from all classes are used to create the MMPD dictionary. As a result, there is no one atom that is common to all the classes of damage. Misclassification occurs in this method only if the time-frequency signature of the signals from one class is very similar to that of another class. Table 3.4 shows the classification results for the MMPD algorithm, using data from PZT3. The average correct classification rate is 98.8%, where the corresponding rate using the MPD-TFR classifier with PZT3 signals is 94%. In addition, the MMPD used a dictionary consisting of 6 million atoms as opposed to the 42 million atom dictionary used for the MPD-TFR classifier. The MMPD also needed only 1 or 2 iterations to achieve a residue energy of about 5% of the original signal whereas 60 iterations were used by the MPD-TFR classifier. Note, however, that the MMPD dictionary is very application specific and is limited by the acquired training data. Also, there is no analytic form of the dictionary components, and this may add to the processing time. Nevertheless, due to the smaller dictionary size and reduced number of required iterations, the computation time of the MMPD classification was less than that of the MPD-TFR based classification.

These findings have been published in [21].

Class	1	2	3	4	5
1	**0.94**	0.00	0.06	0.00	0.00
2	0.00	**1.00**	0.00	0.00	0.00
3	0.00	0.00	**1.00**	0.00	0.00
4	0.00	0.00	0.00	**1.00**	0.00
5	0.00	0.00	0.00	0.00	**1.00**

Table 3.4. MMPD classification results for data from PZT3.

3.5. Bayesian Sensor Fusion

In SHM, data is often available from multiple sensors located at different physical positions. The goal of sensor fusion is to combine the information received from multiple sensing devices in a coherent manner so as to increase the overall damage detection performance.

If $s^{(k)}(t)$, $k = 1, \ldots, K$, is the data from the kth sensor, then an optimal Bayesian data fusion scheme relies on the *joint* density function, $P\left(s^{(1)}(t), \ldots, s^{(K)}(t)|m\right)$, $m = 1, \ldots, M_c$ of the data under class assumption m. Specifically, the data is assigned to class m^* if

$$m^* = \underset{m=1,\ldots,M_c}{\operatorname{argmax}} \left\{P\left(s^{(1)}(t), \ldots, x^{(K)}(t)|m\right)\right\} ,$$

where a uniform prior density $\Pr(m)$ is assumed for simplicity. As the joint density is often difficult to model, the marginals $P\left(s^{(k)}(t)|m\right)$, $k = 1, \ldots, K$, can be used for the classification. Thus, the data from sensor k is assigned to class m^* if

$$m^* = \underset{m=1,\ldots,M_c}{\operatorname{argmax}} \left\{P\left(x^{(k)}(t)|m\right)\right\} .$$

The sensor data fusion problem is now transformed (sub-optimally) into a decision fusion problem, where the local decisions at each sensor are combined. Defining the discrete indicator variables $c^{(k)} = l$ if the data at the kth sensor is classified to Class l, the joint density function of the local decisions $P\left(c^{(1)}, \ldots, c^{(K)}|m\right)$ is used in the Bayesian decision

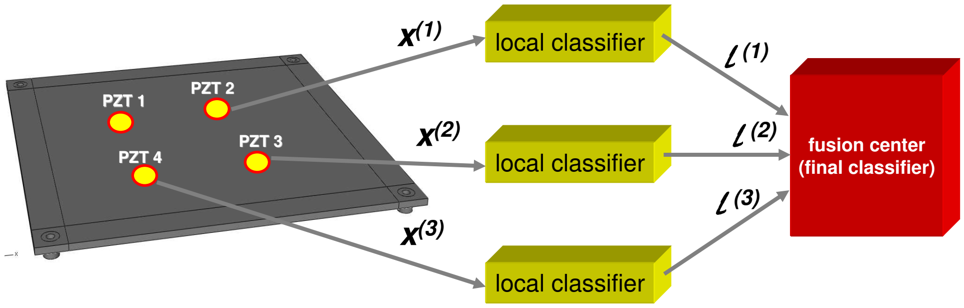


Fig. 3.13. Sensor fusion approach.

fusion to assign the data to Class m^* if

$$m^* = \underset{m=1,\ldots,M_c}{\operatorname{argmax}} \left\{ P\left(c^{(1)},\ldots,c^{(K)}|m\right)\right\} . \tag{3.17}$$

In practice, it is often convenient to assume statistical independence of the local decisions, so that $P\left(c^{(1)},\ldots,c^{(K)}|m\right) = \prod_{k=1}^{K} P\left(c^{(k)}|m\right)$ in Equation (3.17). The marginal density functions $P\left(c^{(k)}|m\right)$, $k = 1,\ldots,K$, can generally be estimated more reliably than the full joint density, and are available directly from the local confusion matrices as $P\left(c^{(k)} = j|m\right) = C^k_{m,l}$, where $C^k_{m,l}$ is the mth row and lth column element of the confusion matrix at sensor k. Note that this Bayesian decision fusion approach can also be used to integrate information flow from multiple sensing modalities and even multiple local detection algorithms. This kind of fusion is especially advantageous in situations where some sensors and algorithms might be in a better position to classify damage from certain classes than others. A schematic of the sensor fusion scheme used here is shown in Figure 3.13.

The average correct classification rates obtained from the MPD-TFR classifier are based on using data from individual sensors only. When the sensor fusion algorithm is used with the MPD-TFR classifier to combine the information received from all the sensors, then the corresponding results are shown in Table 3.5. As it can be seen, the overall performance

Class	1	2	3	4	5
1	**0.92**	0.00	0.08	0.00	0.00
2	0.01	**0.97**	0.00	0.02	0.00
3	0.01	0.00	**0.99**	0.00	0.00
4	0.00	0.01	0.00	**0.99**	0.00
5	0.00	0.00	0.00	0.00	**1.00**

Table 3.5. MPD classification results with sensor fusion.

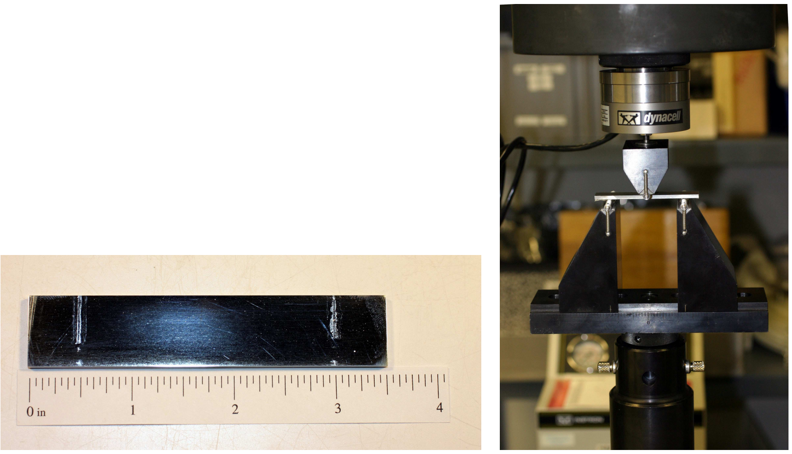
of the system improved significantly as the average correct classification rate with sensor fusion is 97.4%, as compared to 87.6%, 94%, and 93.2% when PZT2 sensor, PZT3 sensor, and PZT4 sensor were used alone, respectively (sensor fusion was not applied with the MMPD classifier because the classification performance using data from individual sensors was already very good).

These findings have been published in [21].

3.6. Ultrasonic Measurement and MMPD Classifier

3.6.1. Experimental setup and data collection

The samples used for these experiments are aluminum 6061-T6 plates, with a mirror finish and cut into 3.5″ x 0.7″ x 0.125″ bars. Aluminum 2024-T351 plates (Figure 3.14(a)) were also used, which were cut into the same bar size and polished to a mirror finish on one side. The mirror finish is necessary for the interferometric detection and also eliminates surface defects that have the potential to be locations for crack initiation during the cyclic loading. The samples were fatigued on a three-point bend fixture (see Figure 3.14(b)), with a 75 lb load at 5 Hz. This results in a maximum bending stress of approximately 245 MPa, which is about 75% of the yield stress. Every 10,000 cycles, the fatigue process was interrupted and samples were evaluated with the laser-ultrasonic system in order to monitor any changes that might occur within the specimen's lifetime. The damage itself is at or below the grain level (about 100 μm), and is not visible to naked eye for low fatigue levels such as 40,000 cycles. For higher fatigue levels like 70,000-100,000 cycles, surface damage was visible in the form of fatigue-induced surface roughness.



(a) Al-2024 plate

(b) Experimental setup for three-point bend fatiguing

Fig. 3.14. Three-point bend fatiguing of the aluminum plates.

The sensing experimental apparatus (Figure 3.15) consists of a pulsed Nd:YAG laser (∼10 ns pulse FWHM, 1064 nm) with an energy of 2.4 mJ, and is used to generate ultrasonic waves in our samples. The pulsed laser allows for non-contact ultrasonic acquisition as well as the ability to control the ultrasonic amplitude without material ablation. As a result, the bulk of the material undergoes no physical changes. The ultrasonic wave surface displacement is measured in transmission mode using a path-stabilized, Michelson-type interferometer with a flat frequency response from 1-60 MHz. This instrument is very sensitive to room vibrations. To control this, a vibration isolated work station is used to maximize the sensitivity of the interferometer to surface displacement. This isolation also fulfills our need to record consistent signals.

To collect ultrasonic information localized to regions that are suspected to contain fa-

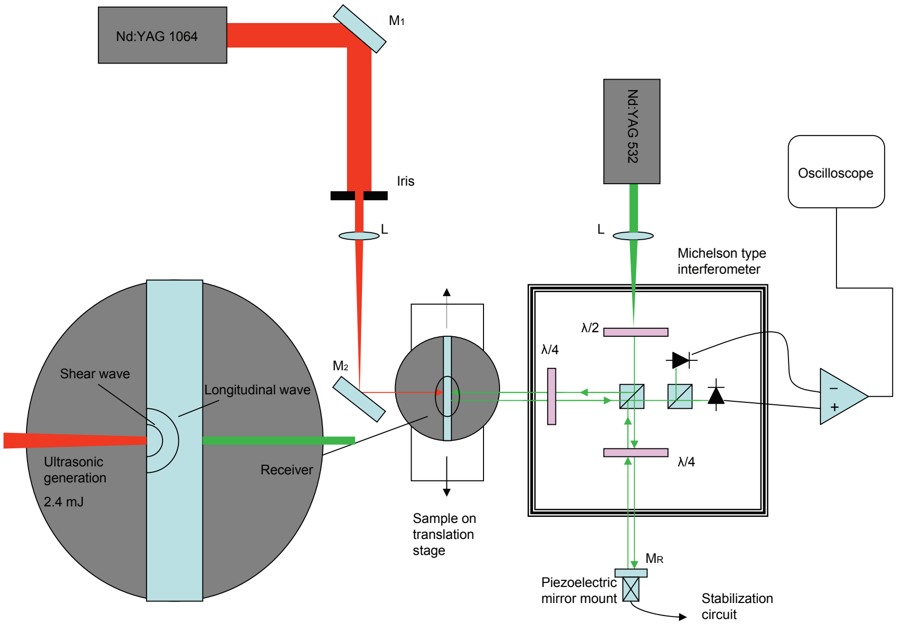


Fig. 3.15. Schematic showing the experimental setup for ultrasonic based detection of fatigue damage in aluminum plates.

tigue damage in the sample, the source and receiver lasers are focused (0.1 mm diameters) in an epicentral configuration (source directly opposite to receiver) allowing for precise determination of ultrasonic properties. The sample is manually translated in 0.5 mm steps while the source and receiver generate and gather ultrasonic information as a function of position. After irradiating the sample, the interferometer output is recorded by a digital oscilloscope. The signal is digitized at a sampling rate of 10 GHz providing significant over-sampling and ensuring complete characterization of noise characteristics. For a reasonable signal-to-noise ratio (SNR), 30 signals are averaged to eliminate the noise generated by the pulsed laser.

A typical signal is shown in Figure 3.16(a), with t_{L} and t_{S} indicating the arrival times for the longitudinal and shear waves, respectively. The step displacements of the longitudinal and shear waves are A_{LS} and A_{SS}, and the noise at time zero is light from the generation laser entering the interferometer photodetectors - this helps establish time zero. The epicentral ultrasonic displacement essentially follows previously known results and established models for laser generated ultrasound in metals [93].

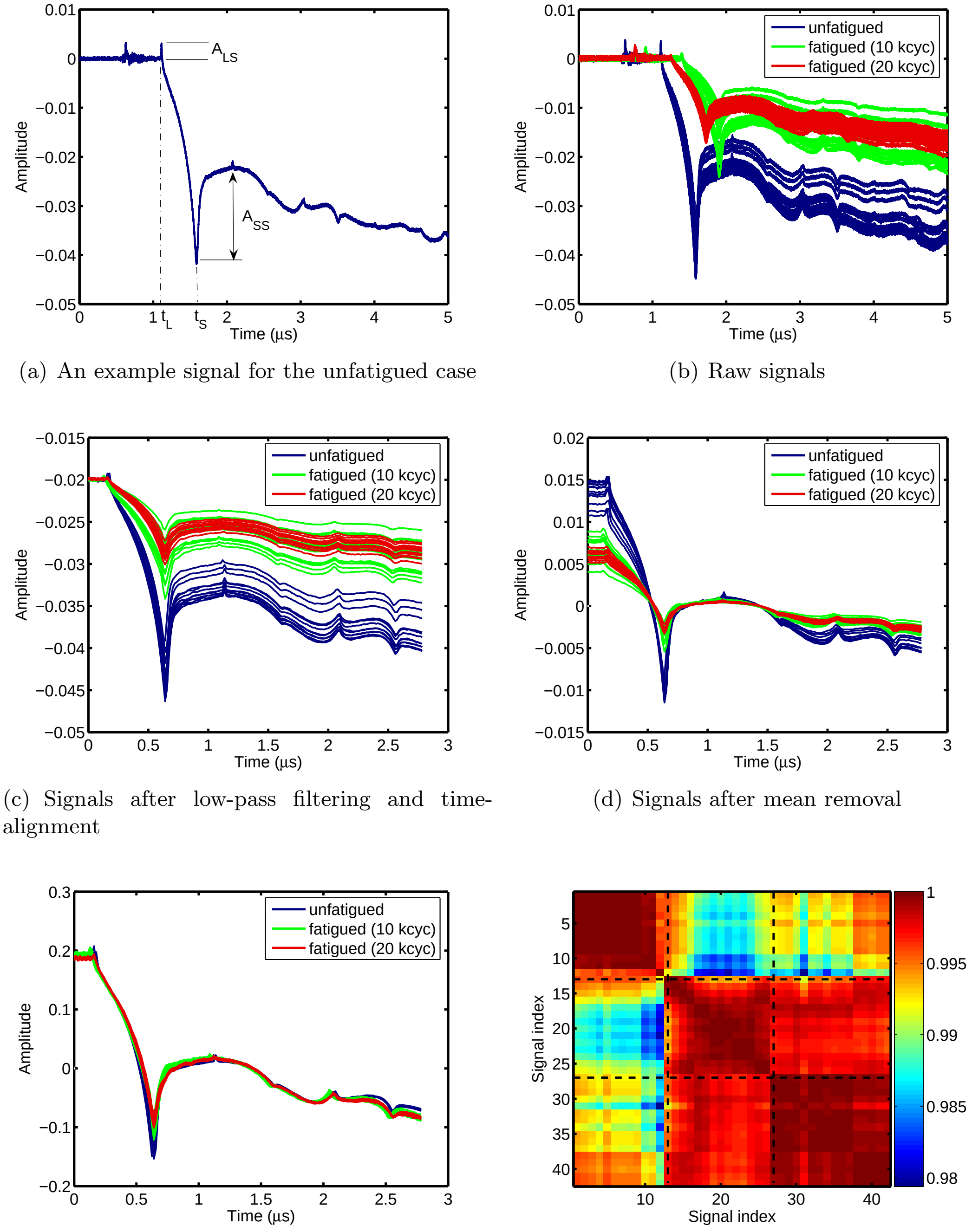
Data was collected for three different structural conditions (classes): unfatigued sample, sample fatigued to 10,000 cycles, and sample fatigued to 20,000 cycles. About 15 signals were available for each class.

3.6.2. Preprocessing

The measured signals were first lowpass filtered and down-sampled. This was followed by time-alignment, mean removal, and normalization. The preprocessing steps are illustrated in Figures 3.16(b)-(e). The plots show the signals from the three classes. Figure 3.16(f) shows the time-domain correlations between the signals. While the block diagonal structure of the correlation plot is quite evident (signals from the same class correlate better than those from different classes), we see that the correlation values themselves are not very different. A classifier based simply on time-domain correlations is therefore not expected to be very robust to noise.

3.6.3. MMPD Classifier and Choice of Parameters

Figure 3.17 shows plots of the signals both in the time-domain and time-frequency plane. The latter is useful for analyzing the time-varying spectral content of the signals. The time-frequency representation (TFR) used here for this purpose is the Short-Time Fourier Transform (STFT) [88]. We employ the STFT with a Gaussian window (known for its good time-frequency localization properties), so that good resolution is retained both in time and frequency. From the TFR plots we see that the shear waves are more attenuated in the



(a) An example signal for the unfatigued case

(b) Raw signals

(c) Signals after low-pass filtering and time-alignment

(d) Signals after mean removal

(e) Signals after normalization

(f) Signal correlations

Fig. 3.16. Data and preprocessing.

Damage	0Kcyc	10 kcyc	20 kcyc
0 kcyc	6/6	0/6	0/6
10 kcyc	0/7	7/7	0/7
20 kcyc	0/8	0/8	8/8

Table 3.6. MMPD classification on ultrasonic data.

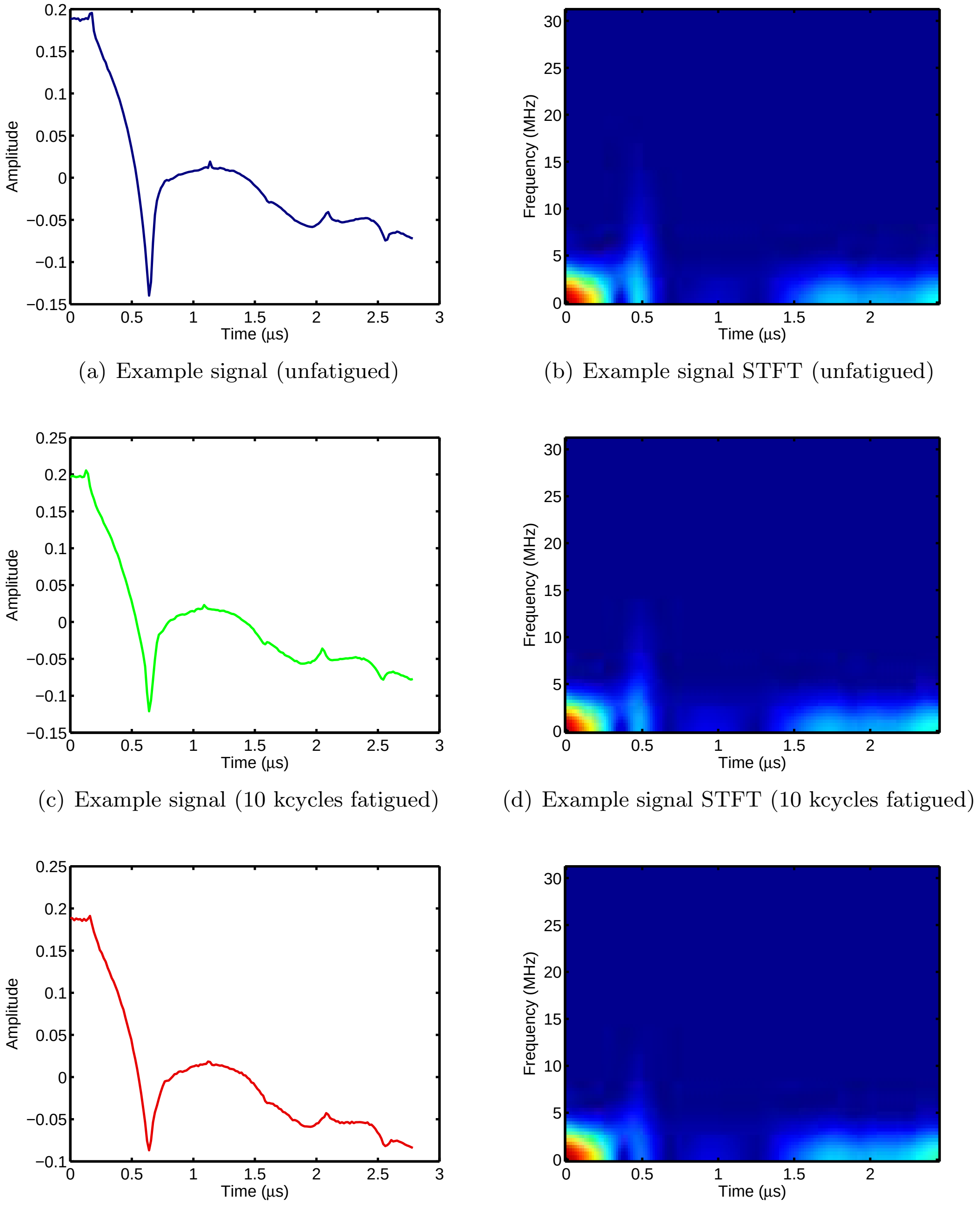
fatigued samples as compared to the unfatigued case, especially at the higher frequencies. The MMPD time-frequency based technique performs damage classification by exploiting these differences between the signals for the various structural conditions.

Half of the available data was used for constructing the MMPD dictionary (training data), and the remaining half was used for testing the classifier performance. Only one MMPD iteration was performed to decompose the test signals, because this was found to be sufficient to reduce the residue energy to less than 20%.

3.6.4. Classification Results

We present classification results here in the form of a confusion matrix. Essentially, the (i, k)th entry of this matrix indicates the probability that data actually from class i is classified as belonging to class k. In the present problem there are three classes, and we obtain the 3×3 confusion matrix shown in Table 3.6.

Upon testing, our MMPD classifier resulted in correct classification for all the tested data and this is quite promising. However, the size of the data set was too small to make any significant claims. These findings have been published in [25].



(a) Example signal (unfatigued)

(b) Example signal STFT (unfatigued)

(c) Example signal (10 kcycles fatigued)

(d) Example signal STFT (10 kcycles fatigued)

(e) Example signal (20 kcycles fatigued)

(f) Example signal STFT (20 kcycles fatigued)

Fig. 3.17. Example plots showing representative (preprocessed) signals from the three classes and the corresponding short-time Fourier Transforms (STFTs).

CHAPTER 4

Hidden Markov Model based Damage Classification

4.1. Hidden Markov Models

Given a set of observation sequence data $\mathbb{Y}$, a hidden Markov model (HMM) with a set of parameters $\boldsymbol{\theta}$ can be estimated that best describes the set. The HMM [94] is a probabilistic method used for modeling sequential data [94]. For a length-L_T observation sequence $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_{L_T}\}$, $Y \in \mathbb{Y}$, the HMM defines a probability distribution over Y by invoking a sequence of unobserved (hidden) discrete states $\mathbf{x} = \{x_1, \ldots, x_{L_T}\}$. The model imposes Markov dynamics on the sequence of hidden states, and it ensures independence of the observations $\mathbf{y}_n$ from all other variables given x_n. Suppose that the number of distinct states is S, with the state variables x_n assuming values from the alphabet $\{1, \ldots, S\}$. The model is then parameterized by the $S \times 1$ initial state distribution vector $\boldsymbol{\pi}$ whose ith element is the probability that x_1 equals i (denoted as $\Pr(x_1 = i)$), the $S \times S$ state-transition matrix A whose (i,k)th element is $a_{x_n, x_{n+1}} = a_{i,k} = \Pr(x_{n+1} = k | x_n = i)$, and the state-dependent observation density B whose kth element is $b_k(\mathbf{y}_n) = P(\mathbf{y}_n | x_n = k)$. The HMM parameters are denoted by $\boldsymbol{\theta} = \{\boldsymbol{\pi}, A, B\}$.

If the observations $\mathbf{y}_n$ are discrete and restricted to the alphabet $\mathbb{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_{K_{\text{obs}}}\}$, where K_{obs} are the total number of possible discrete levels, then B reduces to a $S \times K_{\text{obs}}$ matrix whose (i,k)th element is $b_{ik} = \Pr(\mathbf{y}_n = \mathbf{v}_k | x_n = i)$ and the model is known as a *discrete* HMM. In a *continuous* HMM, the observations are continuous and B is often modeled using a Gaussian mixture model (GMM) with M_G components:

$$b_k(\mathbf{y}_n) = \sum_{m=1}^{M_G} p_{km} \mathcal{N}(\mathbf{y}_n, \boldsymbol{\mu}_{km}, \Sigma_{km}), \tag{4.1}$$

where $\mathcal{N}(\cdot)$ represents the Gaussian distribution, and p_{km}, $\boldsymbol{\mu}_{km}$, and Σ_{km} are the coefficients, mean, and covariance matrices, respectively, of the mth mixture component.

Given a set of 'training' observation sequences $\mathbb{Y}^{\mathrm{tr}}$, and an S-state HMM assumption $\Lambda_S \in \mathbb{M}_{\mathbf{HMM}}(S)$ (where $\mathbb{M}_{\mathbf{HMM}}(S)$ is the set of all S state HMMs), one first computes a maximum-likelihood (ML) estimate of the parameters $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P\left(\mathbb{Y}^{\mathrm{tr}} | \boldsymbol{\theta}, \Lambda_S\right) \tag{4.2}$$

using the Baum-Welch algorithm [94], a special case of the expectation-maximization (EM) algorithm [95] which iteratively maximizes the likelihood of the training data. At the lth iteration,

$$\boldsymbol{\theta}^{(l+1)} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{\mathbf{x}} \Pr\left(\mathbf{x} | \mathbb{Y}^{\mathrm{tr}}, \boldsymbol{\theta}^{(l)}, \Lambda_S\right) \log P\left(\mathbf{x}, \mathbb{Y}^{\mathrm{tr}} | \boldsymbol{\theta}, \Lambda_S\right) \tag{4.3}$$

where the sum is over all possible state sequences. The algorithm is guaranteed to converge to a local maximum of the likelihood function [96].

The predictive likelihood of a 'test' observation sequence Y^{te} can then be computed as

$$\begin{aligned} P\left(Y^{\mathrm{te}} | \boldsymbol{\theta}_{\mathrm{ML}}, \Lambda_S\right) &= \sum_{\mathbf{x}} P\left(\mathbf{x}, Y^{\mathrm{te}} | \boldsymbol{\theta}_{\mathrm{ML}}, \Lambda_S\right) \\ &= \sum_{\mathbf{x}} \boldsymbol{\pi}_{x_1} \prod_{n=1}^{N_T - 1} a_{x_n, x_{n+1}} \prod_{n=1}^{N_T} b_{x_n}(\mathbf{y}_n^{\mathrm{te}}), \end{aligned} \tag{4.4}$$

where $a_{x_n,x_{n+1}}$ and b_{x_n} are elements of the matrices A and B. Note that information on how to reduce the complexity of the algorithm can be found in [94].

4.2. Model Selection and Variational Bayesian Learning

ML learning, as in Equation (4.2), provides no information about the uncertainty of the parameters estimated. Moreover, it does not account for model complexity and is susceptible to over-fitting the data [96]. The likelihood function is unbounded and it is possible to increase the likelihood of the data by using models of increasing complexity.

Using ML learning with unnecessarily complex models is therefore not recommended as it has the disadvantages of increased data requirements and computational burdens.

Conventional methods for dealing with this problem include limiting the number of model parameters (such as the number of HMM states S), constraining the form and information-flow in the model (for example, by constraining the connectivity of the state transition matrix A), using maximum a posteriori (MAP) [96] learning with regularizing priors on the parameters, or using cross-validation. These techniques work well in some applications, but often they become computationally expensive.

In the Bayesian setting, the task of model selection comprises calculating the posterior distribution over a set of models $\mathbb{M}$ given some dataset $\mathbb{Y}$ and a priori knowledge $P(\mathbb{M})$, where $P(\cdot)$ is a continuous pdf. From Bayes' rule, the posterior is

$$P(\mathbb{M}|\mathbb{Y}) = \frac{P(\mathbb{Y}|\mathbb{M})\ P(\mathbb{M})}{P(\mathbb{Y})} \propto P(\mathbb{Y}|\mathbb{M})\ P(\mathbb{M}). \tag{4.5}$$

Assuming a uniform prior $P(\mathbb{M})$, the key quantity for Bayesian model selection is the evidence for model $\mathbb{M}$:

$$P(\mathbb{Y}|\mathbb{M}) = \int P(\mathbb{Y}|\boldsymbol{\theta}, \mathbb{M})\ P(\boldsymbol{\theta}|\mathbb{M})\ d\boldsymbol{\theta}, \tag{4.6}$$

where $\boldsymbol{\theta}$ represents the set of parameters. Models are ranked by evaluating the evidence, which embodies the principle of Occam's razor since it automatically penalizes complex models with more parameters [96]. The Bayesian integration in Equation (4.6) is, however, not easy to compute directly.

This problem is solved using variational Bayesian (VB) learning [96,97], where we lower bound the evidence $P\left(\mathbb{Y}|\mathbb{M}\right)$ using an approximate probability distribution $P\left(\mathbf{x},\boldsymbol{\theta}\right)$ as [97]

$$\begin{aligned}
\log P\left(\mathbb{Y}|\mathbb{M}\right) &= \log \int \sum_{\mathbf{x}} P\left(\mathbf{x},\boldsymbol{\theta}\right) \frac{P\left(\mathbf{x},\mathbb{Y},\boldsymbol{\theta}|\mathbb{M}\right)}{P\left(\mathbf{x},\boldsymbol{\theta}\right)} d\boldsymbol{\theta} \\
&\geq \int \sum_{\mathbf{x}} P\left(\mathbf{x},\boldsymbol{\theta}\right) \log \frac{P\left(\mathbf{x},\mathbb{Y}|\boldsymbol{\theta},\mathbb{M}\right)\ P\left(\boldsymbol{\theta}|\mathbb{M}\right)}{P\left(\mathbf{x},\boldsymbol{\theta}\right)} d\boldsymbol{\theta} \\
&\equiv \mathcal{F}_{\mathbb{M}}(P\left(\mathbf{x},\boldsymbol{\theta}\right)). \qquad (4.7)
\end{aligned}$$

For many models and approximating distributions, the variational objective function $\mathcal{F}_{\mathbb{M}}$ and its derivatives with respect to the approximating distributions' parameters can be evaluated. Since

$$\begin{aligned}
\mathcal{F}_{\mathbb{M}}(P\left(\mathbf{x},\boldsymbol{\theta}\right)) &= \int \sum_{\mathbf{x}} P\left(\mathbf{x},\boldsymbol{\theta}\right) \log \frac{P\left(\mathbb{Y}|\mathbb{M}\right)\ P\left(\mathbf{x},\boldsymbol{\theta}|\mathbb{Y},\mathbb{M}\right)}{P\left(\mathbf{x},\boldsymbol{\theta}\right)} d\boldsymbol{\theta} \\
&= \log P\left(\mathbb{Y}|\mathbb{M}\right) - \int \sum_{\mathbf{x}} P\left(\mathbf{x},\boldsymbol{\theta}\right) \log \frac{P\left(\mathbf{x},\boldsymbol{\theta}\right)}{P\left(\mathbf{x},\boldsymbol{\theta}|\mathbb{Y},\mathbb{M}\right)} d\boldsymbol{\theta} \\
&= \log P\left(\mathbb{Y}|\mathbb{M}\right) - \mathcal{D}_{\mathrm{KL}}\left[P\left(\mathbf{x},\boldsymbol{\theta}\right)||P\left(\mathbf{x},\boldsymbol{\theta}|\mathbb{Y},\mathbb{M}\right)\right],
\end{aligned}$$

where $\mathcal{D}_{\mathrm{KL}}$ denotes the Kullback-Leibler distance [96] (KLD) measure. The KLD is always non-negative $\mathcal{D}_{\mathrm{KL}}\left[P\left(\mathbf{x},\boldsymbol{\theta}\right)||P\left(\mathbf{x},\boldsymbol{\theta}|\mathbb{Y},\mathbb{M}\right)\right] \geq 0$, where the lower bound is maximized (and achieves equality) when $P\left(\mathbf{x},\boldsymbol{\theta}\right) = P\left(\mathbf{x},\boldsymbol{\theta}|\mathbb{Y},\mathbb{M}\right)$.

Using the variational Bayesian EM (VBEM) algorithm [97], the bound $\mathcal{F}_{\mathbb{M}}$ is iteratively maximized with respect to $P\left(\mathbf{x},\boldsymbol{\theta}\right)$, and convergence is guaranteed to a local maximum of $\mathcal{F}_{\mathbb{M}}$. The optimized variational posterior $P\left(\mathbf{x},\boldsymbol{\theta}\right)$ simultaneously yields an approximation for the true posterior $P\left(\mathbf{x},\boldsymbol{\theta}|\mathbb{Y},\mathbb{M}\right)$. In contrast to conventional ML learning where we attempt to infer a single set of parameters $\boldsymbol{\theta}$, VB learning optimizes a whole ensemble over $\boldsymbol{\theta}$.

The relevant update equations, specialized to the case of hidden Markov models, can be found in [97–99]. It is important to note that the computational complexity of variational Bayesian learning is not very different from that of the standard (ML) Baum-Welch algorithm [97, 98].

4.3. HMM Based Damage Classifier

The number of states S to use in the HMMs can be estimated empirically by examination of the time-frequency (TF) representation of the data. We make use of the cross-term free TF representation given in [100] that can be computed directly from the signal's MPD. We choose S based on the number of stationarities in the data. An example of a 3-state definition is shown in Figure 4.1.

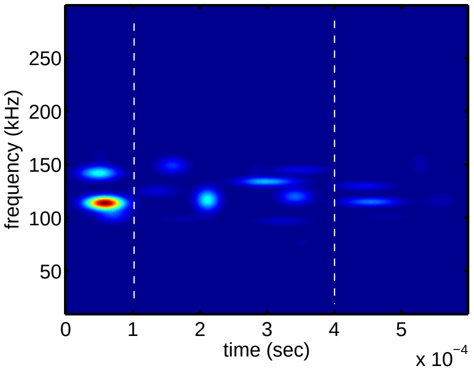


Fig. 4.1. MPD-TFR of a real signal from a bolted joint, demonstrating a 3-state definition used in the HMM classifier.

The data from each damage class is modeled using an HMM, with the available training data from each class used to learn the parameters of the corresponding HMM. Once the model parameters have been estimated, unknown test data is classified based on its likelihood as computed by the HMM associated with each damage class. Specifically, the classifier assigns a given test observation sequence Y^{te} to damage class k^*, given by

$$k^* = \underset{k}{\operatorname{argmax}}\, P\left(Y^{\mathrm{te}} | \boldsymbol{\theta}_{\mathrm{ML}}^{(k)}, \Lambda_S\right), \tag{4.8}$$

where $\boldsymbol{\theta}_{\mathrm{ML}}^{(k)}$ denotes the parameters of the kth HMM (trained on data from damage class k).

4.4. Variational Bayes for Estimating Number of HMM States

All other things being equal, the model complexity of the HMM is governed by the number of states S. The number of parameters to be estimated increases with S, and if the complexity of the model is not limited by the amount of available data, the result is overfitting. In this work, we use variational Bayes to select an HMM of appropriate complexity. Specifically, given a training set of data $\mathbb{Y}^{\mathrm{tr}}$, we employ the VB learning algorithm discussed in Section 4.2 to compute the evidence for the S-state HMM using $\Lambda_S \in \mathbb{M}_{\mathbf{HMM}}$ in Equation (4.6).

The appropriate number of states S^* is then determined by comparing the evidences for various S as

$$S^* = \underset{S}{\operatorname{argmax}}\, P\left(\mathbb{Y}^{\mathrm{tr}} | \Lambda_S\right). \tag{4.9}$$

Note that a good approximation to the full posterior distribution over the hidden states and the parameters is now available in the form of the optimized variational posterior:

$$P\left(\mathbf{x}, \boldsymbol{\theta} | \mathbb{Y}^{\mathrm{tr}}, \Lambda_{S^*}\right) \approx P\left(\mathbf{x}, \boldsymbol{\theta}\right). \tag{4.10}$$

where $\boldsymbol{\theta}$ are the parameters of Λ_{S^*}. VB learning provides an efficient framework for automatically selecting a model of appropriate complexity from the data, and helps avoid the overfitting problems encountered in conventional ML learning.

In a Bayesian HMM, we wish to infer the full posterior distribution over the parameters $\boldsymbol{\theta}$:

$$P\left(\boldsymbol{\theta} | \mathbb{Y}^{\mathrm{tr}}, \Lambda_{S^*}\right) \propto P\left(\mathbb{Y}^{\mathrm{tr}} | \boldsymbol{\theta}, \Lambda_{S^*}\right)\, P\left(\boldsymbol{\theta} | \Lambda_{S^*}\right). \tag{4.11}$$

Note that once the states have been estimated, we still use the regular EM algorithm. The HMM based damage classification is summarized in Algorithm 1.

Algorithm 1 HMM based damage classification.

```
Time-Frequency Feature Extraction using MPD
HMM Based Damage Classifier
Variational Bayes for Automatic Estimation of HMM State Number
```

The predictive likelihood of test observation sequence Y^{te} is now given by

$$P\left(Y^{\mathrm{te}}, \Lambda_{S^*}\right) = \int P\left(Y^{\mathrm{te}} | \boldsymbol{\theta}, \Lambda_{S^*}\right)\, P\left(\boldsymbol{\theta} | Y^{\mathrm{te}}, \Lambda_{S^*}\right)\, d\boldsymbol{\theta}. \tag{4.12}$$

This integral can be approximated by using the variational posterior resulting from the ensemble learning as

$$P\left(Y^{\mathrm{te}} | \mathbb{Y}^{\mathrm{tr}}, \Lambda_{S^*}\right) \approx \int P\left(Y^{\mathrm{te}} | \boldsymbol{\theta}, \Lambda_{S^*}\right)\, P\left(\boldsymbol{\theta}\right)\, d\boldsymbol{\theta}. \tag{4.13}$$

Because of the asymmetric KLD measure penalty in the optimization process, the variational posterior tends to over-neglect areas of low posterior probability in the parameter space. As a result, the quantity above might be an overestimate for the density of Y^{te}.

Integrating out the model parameters in Equation (4.13) is still as intractable as it was for the marginal likelihood in Equation (4.6). Point estimates such as the mean of the posterior can be used to simplify computation [97–99]. Note that this can be compared with the maximum likelihood approach in Equation (4.4), where one simply utilizes a point estimate $P\left(\boldsymbol{\theta}\right) \approx \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{ML}})$.

4.5. Comparing HMM and MPD-TFR Damage Classification

4.5.1. Time-Frequency Feature Extraction using MPD

The critical first step of a successful classification system is the extraction of effective discriminatory features. In this work, we employ the amplitude-time-frequency-scale features $\{\alpha_k, \tau_k, \nu_k, \kappa_k\}$ extracted by the MPD to encode the information necessary for distinguishing signals from different damage classes. Specifically, an N-iteration MPD is first applied to each measured signal, resulting in a collection of N continuous coefficients $\{\alpha_k, \tau_k, \nu_k, \kappa_k\}_{k=1,\dots,N}$. These are then cast as a sequence of 4-dimensional vectors of length $L_T = N$, and are subsequently used as a vector observation sequence to be statistically modeled for classification.

4.5.2. Experimental Setup and Data Collection

The data comprises cyclic fatigue loading tests conducted on an Al 6061-T651 single-lap bolted joint sample. The experimental setup used is shown in Figure 4.2. The lap is 0.125 inches thick, with a 5 mm notch machined in the top lap. Nine piezoelectric sensors are mounted on the lap, one is used as an actuator for transmitting a 130 kHz burst signal and the remaining ones are used as receivers. Both structural and material damage is induced (separately) in the joint by subjecting it to a 20 Hz, 90-900 kg tension-tension loading. Structural damage is defined here in terms of the torque on the bolts and material damage based on the length of the crack initiated at the notch in response to fatiguing. Data is collected for four structural damage classes: bolt torque = 0%, 30%, 60%, and 100% (healthy), and eight material damage classes: number of fatigue cycles = 0 (healthy), 130, 135, 140, 145, 155, 200, and 285 kilo-cycles, with 100 measurements taken for each case.

4.5.3. Choice of Model Parameters

The measured signals are first mean-centered, normalized, and time-aligned. For each signal, TF damage features are extracted using $N = 20$ MPD iterations with a dictionary composed of 5.8 million normalized time-frequency Gaussian atoms as defined in Equation (3.5). The MPD truncation limit was chosen such that a residual signal energy of about 5% could be achieved.

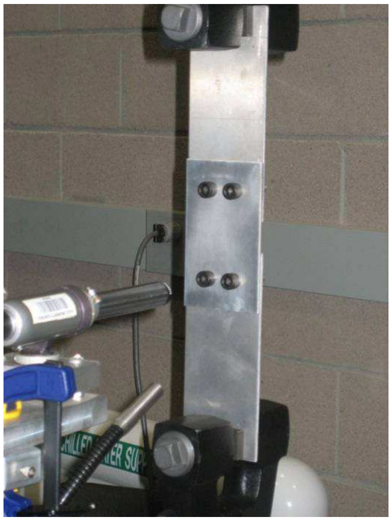


Fig. 4.2. Setup for structural and material damage in a bolted joint.

For modeling with the discrete HMM, the features are quantized to $K_{\text{obs}} = 64$, 128, and 256 symbols using the k-means algorithm [96]. In the continuous HMMs, $M_G = 6$ components are used with the Gaussian mixture models.

Figure 4.3 shows an example plot of the log-evidence $\log \Pr\left(\mathbb{Y}^{\text{tr}} | \Lambda_S\right)$ (actually, the lower bound computed using VB learning) as a function of the number of HMM states S, for the case of structural damage with bolt at 60% torque. We see that the largest evidence is for $S = 2$ to 5 HMM states, and it decreases as S is increased further (with increasing model complexity). In all the simulations, we use HMMs with $S = 3$ states. Note that this choice agrees well with the state definition shown in Figure 4.1, which was determined empirically.

Of the 100 signals available for each damage class, 40 signals are used for training the HMMs, 30 are used for validation (estimating M_G for the continuous HMM), and 30 for testing the classifier performance.

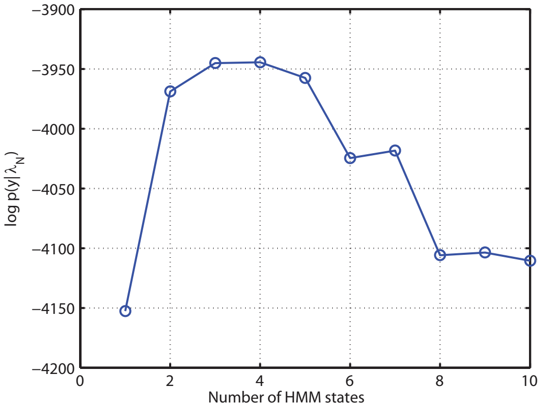


Fig. 4.3. Log-evidence $\log \Pr\left(\mathbb{Y}^{\mathrm{tr}}|\Lambda_S\right)$ as a function of number of HMM states S, for the case of structural damage with bolt at 60% torque.

4.5.4. Classification Results

The performance of the classifier is quantified here using confusion matrices. The (i,k)th element of a confusion matrix indicates the probability that data from class i is classified as being from class k. In all cases, the correct classification probabilities (the diagonal entries of the confusion matrices) are provided in bold font.

Tables 4.1 and 4.2 show the confusion matrices for structural damage classification using a discrete HMM with $K_{\mathrm{obs}} = 256$ symbols and a continuous HMM, respectively. While the $K_{\mathrm{obs}} = 256$-symbol discrete HMM classifier yields good classification results, the performance of the continuous HMM classifier is better (the average correct classification rate is higher).

Table 4.3 shows the average correct classification rates for structural damage classification obtained from the discrete HMM (as a function of the number K_{obs} of symbols used) and the continuous HMM. We see that the performance of the discrete HMM classifier improves as K_{obs} increases. This is expected, because the information loss associated with the quantization step is smaller for larger K_{obs}. In addition the performance of the continuous

HMM classifier is superior to that of the discrete HMM classifiers. This improvement in performance from the discrete to the continuous HMM (and between discrete HMMs with increasing K_{obs}) comes with added computational cost.

Class	1	2	3	4
1	**0.8667**	0	0.0667	0.0667
2	0	**0.9667**	0.0333	0
3	0	0.0333	**0.9333**	0.0333
4	0	0.0333	0	**0.9667**

Table 4.1. Structural damage classification using discrete HMM ($K_{\text{obs}} = 256$ symbols).

Class	1	2	3	4
1	**0.9667**	0	0.0333	0
2	0	**1.0000**	0	0
3	0	0	**0.9333**	0.0667
4	0	0	0.0667	**0.9333**

Table 4.2. Structural damage classification using continuous HMM.

No. of symbols K_{obs}	64	128	256	CHMM
Avg. correct classification rate	**0.8167**	**0.8583**	**0.9333**	**0.9583**

Table 4.3. Structural damage classification rates from discrete HMM (as a function of K_{obs}) and continuous HMM (CHMM).

Tables 4.4 and 4.5 show the confusion matrices for material damage classification using a discrete HMM with $K_{\text{obs}} = 256$ symbols and a continuous HMM, respectively. Table 4.6 shows the average correct classification rates for material damage classification obtained using the discrete HMM (as a function of number of symbols used K_{obs}) and the continuous HMM. High classification performance is again observed from both classifiers. And as before, the performance improves with K_{obs} for the discrete HMM; the continuous HMM yields the best performance [26].

Class	1	2	3	4	5	6	7	8
1	**0.83**	0	0	0.07	0	0.10	0	0
2	0	**0.87**	0	0	0.03	0.03	0.03	0.03
3	0	0.07	**0.77**	0	0	0	0	0.17
4	0	0	0	**1.00**	0	0	0	0
5	0	0	0	0	**0.97**	0.03	0	0
6	0	0	0.03	0	0	**0.93**	0	0.03
7	0	0	0	0	0	0.03	**0.97**	0
8	0	0.03	0	0.03	0	0.07	0	**0.87**

Table 4.4. Material damage classification using discrete HMM ($K_{\text{obs}} = 256$ symbols).

Class	1	2	3	4	5	6	7	8
1	**1.00**	0	0	0	0	0	0	0
2	0	**0.93**	0	0	0	0.07	0	0
3	0	0	**0.97**	0	0	0.03	0	0
4	0	0	0	**0.97**	0	0.03	0	0
5	0	0	0	0	**0.97**	0.03	0	0
6	0	0	0	0	0	**1.00**	0	0
7	0	0	0	0	0	0.03	**0.97**	0
8	0	0	0	0	0	0.07	0	**0.93**

Table 4.5. Material damage classification using continuous HMM.

No. of symbols K_{obs}	64	128	256	CHMM
Avg. correct classification rate	**0.7125**	**0.8030**	**0.9000**	**0.9667**

Table 4.6. Material damage classification rates from discrete HMM (as a function of K_{obs}) and continuous HMM.

4.6. Noise Performance

To study the performance of noise on the time-frequency based discrete HMM classification algorithm, we use material damage data simulated on ABACUS using a physics-based finite element model (see Appendix B). In this example, two damage cases were modeled. One was unfatigued (with no defects) and the other was fatigued (with a 6 mm crack). We add controlled amounts of Gaussian noise to the simulated data and then apply MPD to extract useful features from the signals and perform HMM classification on the extracted features.

4.6.1. Data Synthesis and Feature Extraction

The data obtained from the modeling result, is deterministic. We had synthetically added white Gaussian noise to introduce randomness in the otherwise deterministic process. The signal-to-noise ratio (SNR) is a measure of added noise used and is defined as

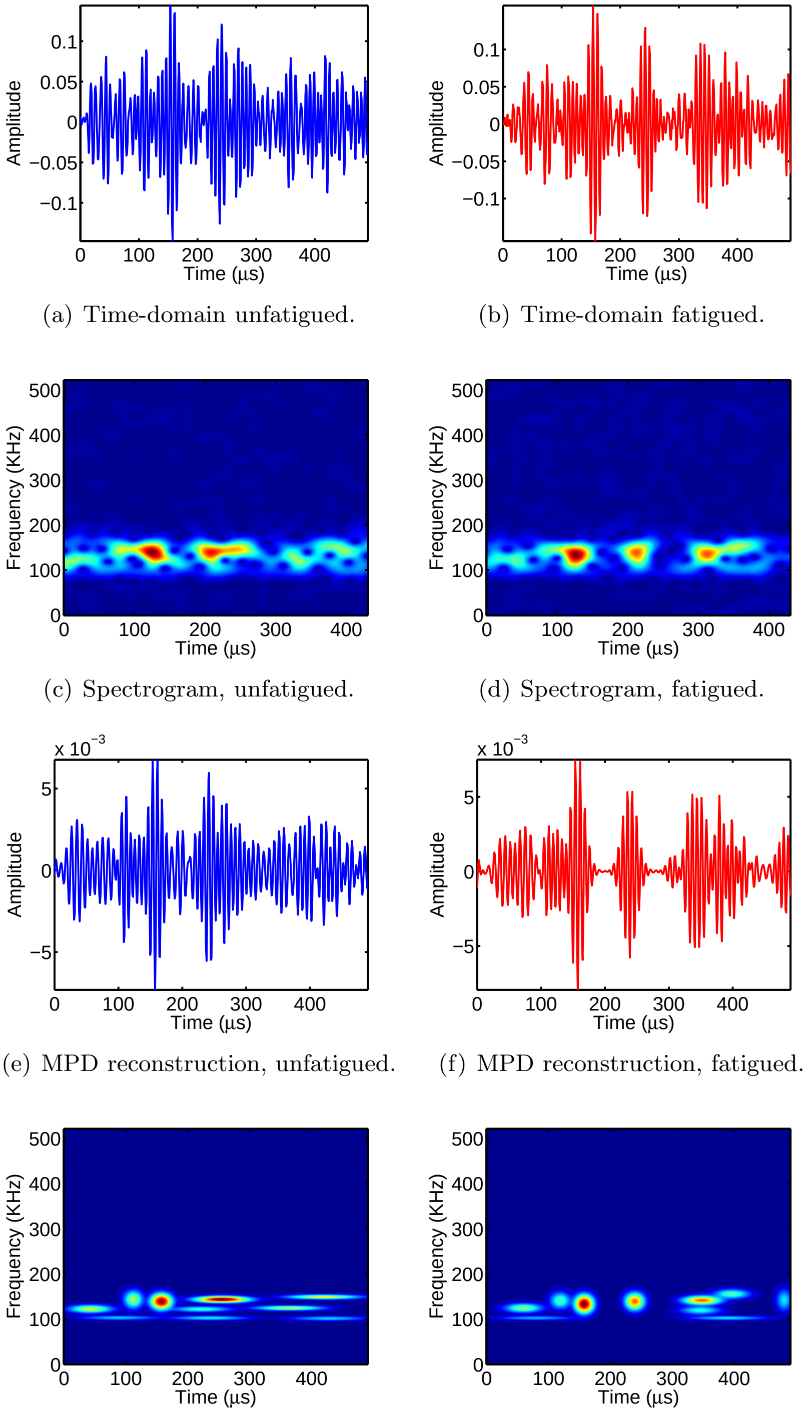
$$\mathrm{SNR} = 10 \log_{10} \left[\frac{1}{L_s \sigma^2} \sum_{n=0}^{L_s - 1} s^2(n) \right] \text{ dB}, \tag{4.14}$$

where $s(n)$ is a discrete time sequence of length L_s and σ^2 is the variance of the noise. We chose SNRs values of -3 to 5 dB, 10 dB, 15 dB and 20 dB. The very low noise 20 dB signal was considered the representative (noise free) class on which the HMM was trained. We had synthesized 450 signal realizations at each SNR for each damage case. Note that all the signals (with added noise) were normalized to unit energy. These time domain signals (shown in Figures 4.4(a) and 4.4(b)) were not separable in time, so the type of damage could not be identified. As the SNR decreases, the differences between the time series from two different damage classes are hard to distinguish, as shown in Figures 4.5(a) and 4.5(b). We also analyzed t he varying SNR signals using the spectrogram time-frequency representation. Figures 4.4(c) and 4.4(d) compare the spectrograms of the healthy and damage signals at 20 dB. A similar comparison is shown at 0 dB in Figures 4.5(c) and 4.5(d). The presence of in-band noise can be observed.

The MPD was performed on these signals as described in Section 3.1. We used a dictionary composed of 33 million time and frequency shifted Gaussian atoms as in Equation (3.5) of various scales, spanning a frequency range of 25-225 kHz (as indicated by the spectrogram analysis). Figures 4.4(e) and 4.4(f) show the time series generated by the MPD reconstruction as shown in Equation (3.1). The signals reconstructed after the MPD had reduced noise (both in-band and out-of-band) and retained the predominant characteristic information pertaining to each class of damage as shown in Figures 4.4(g) and 4.4(h). These MPD-TFRs are constructed according to the Equation (3.7). In the MPD-TFRs presented, the fatigued and the unfatigued signals can be differentiated, especially in the frequency region between 200-300 μs.

A comparison of the MPD performance in the presence and absence of noise can be compared using the MPD-TFR plots presented in Figures 4.4 and 4.5. Even with 0 dB SNR, the MPD reconstruction of the signal was fairly good and the MPD-TFR shows distinction between the fatigued and the unfatigued models.

For the classification, the number of MPD iterations was chosen as $N = 10$. Figures 4.6(a) and 4.6(b) show the residue energy after every MPD iteration for the first 30 iterations. For a 20 dB signal, the residue at the end of 30 iterations is about 1% of the total signal energy and is much less than 10% after 10 iterations. However, this fitting may be an over-fitting of the data and there may be components of noise present in the reconstructed signal. It was experimentally observed that the information after 10 iterations was



(a) Time-domain unfatigued.

(b) Time-domain fatigued.

(c) Spectrogram, unfatigued.

(d) Spectrogram, fatigued.

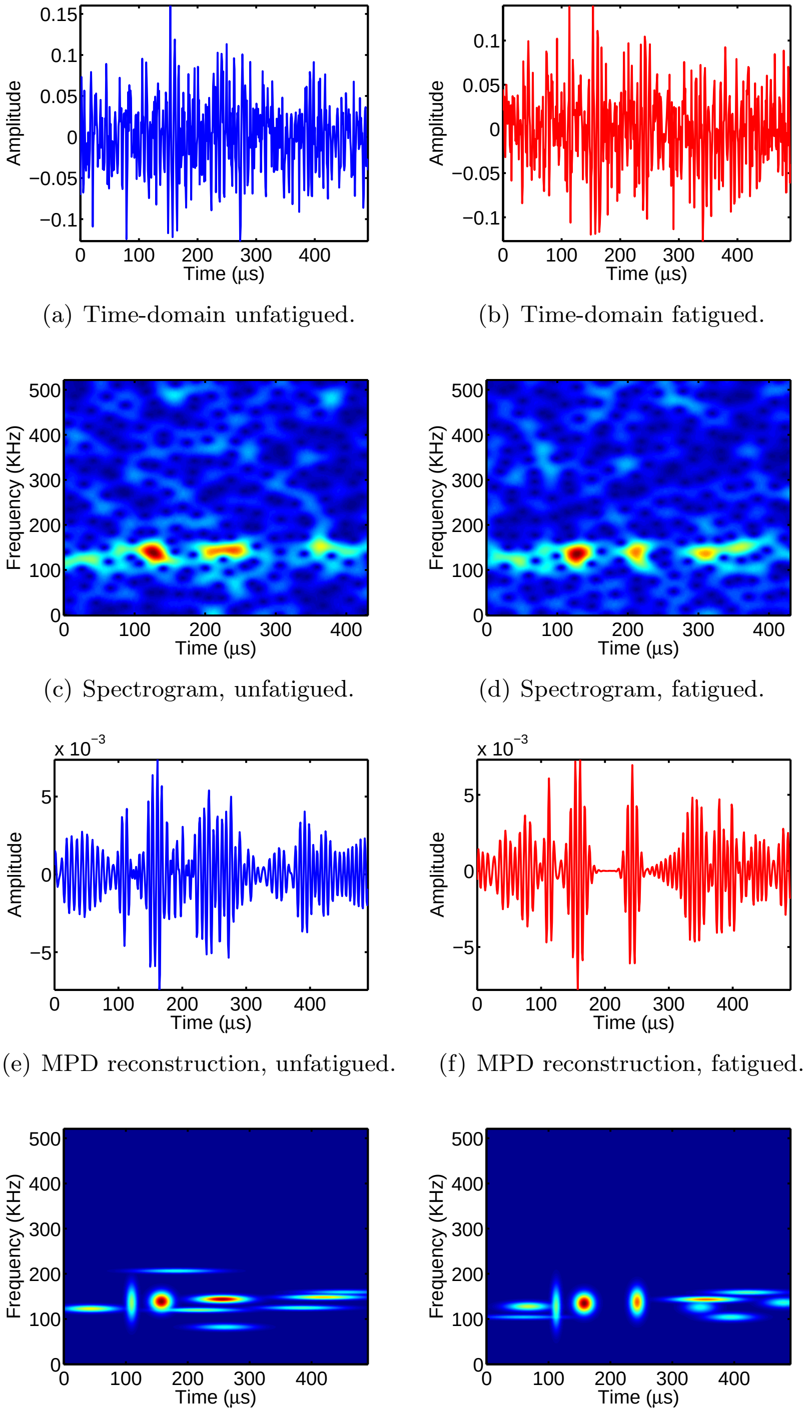
(e) MPD reconstruction, unfatigued.

(f) MPD reconstruction, fatigued.

(g) MPD-TFR, unfatigued.

(h) MPD-TFR, fatigued.

Fig. 4.4. Time domain and time-frequency domain plots of example signals from unfatigued and fatigued data at 20 dB SNR.



(a) Time-domain unfatigued.

(b) Time-domain fatigued.

(c) Spectrogram, unfatigued.

(d) Spectrogram, fatigued.

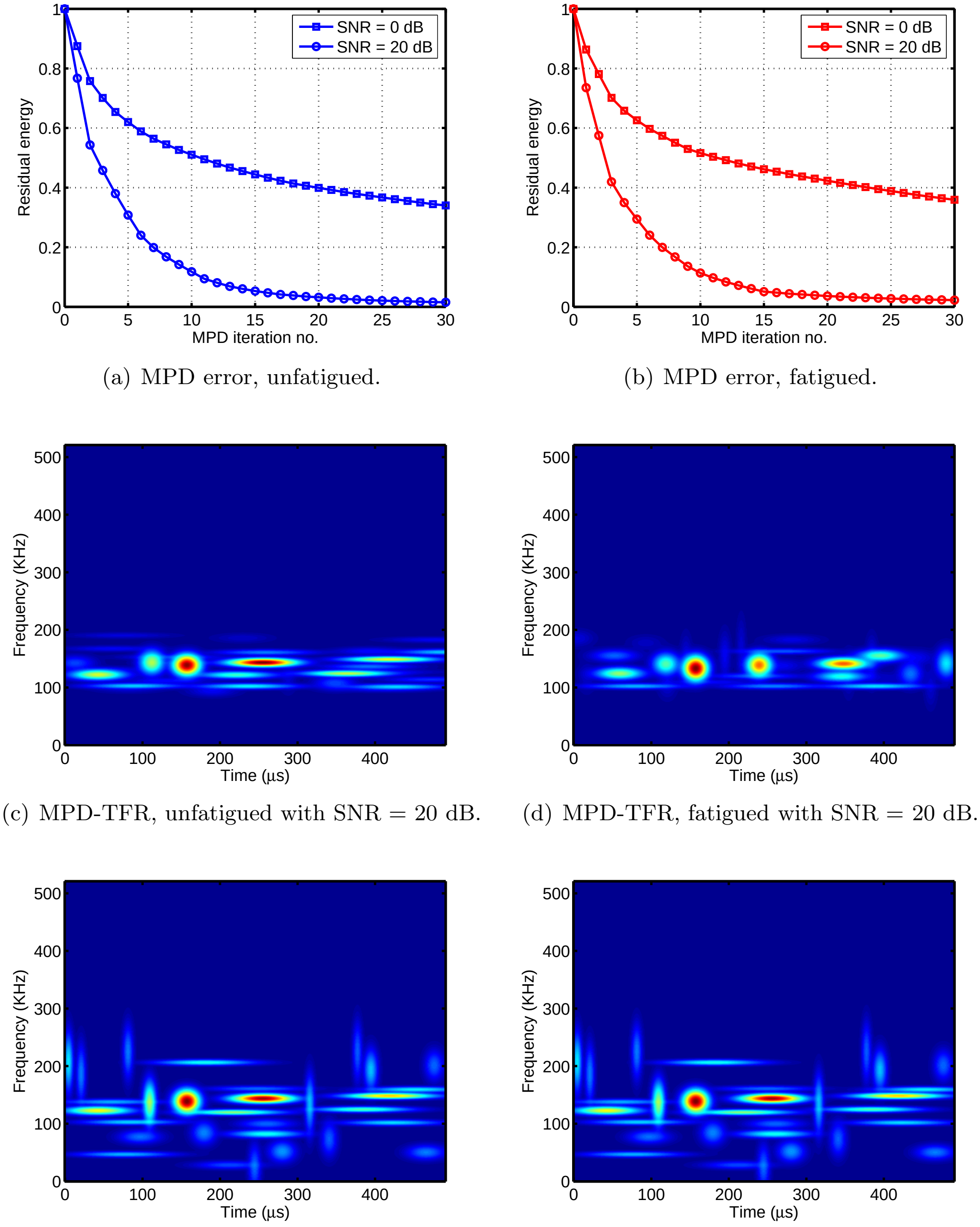
(e) MPD reconstruction, unfatigued.

(f) MPD reconstruction, fatigued.

(g) MPD-TFR, unfatigued.

(h) MPD-TFR, fatigued.

Fig. 4.5. Time domain and time-frequency domain plots of example signals from unfatigued and fatigued data at 0 dB SNR.



(a) MPD error, unfatigued.

(b) MPD error, fatigued.

(c) MPD-TFR, unfatigued with SNR = 20 dB.

(d) MPD-TFR, fatigued with SNR = 20 dB.

(e) MPD-TFR, fatigued with SNR = 0 dB.

(f) MPD-TFR, fatigued with SNR = 0 dB.

Fig. 4.6. Effect of noise on MPD residual error and TFR reconstruction (for 30 iterations).

sufficient to use with the high SNR data. This observation may be further extended to the 0 dB signals. In this case, the residue is considerably higher and after 10 iterations, the algorithm decomposed about 50% of the total signal energy. In addition, we may observe that the decay rate of the residue is slower in the noisy signal. Figures 4.6(c)- 4.6(f) show the MPD-TFR when $N = 30$ iterations were used to reconstruct the signal. As a lot of noise was still present in the data, when compared to Figures 4.4 and 4.5, we noticed an increased ambiguity between the damage classes.

4.6.2. Choice of Parameters for the Discrete HMM Classifier

In this application, we built two discrete HMMs, one for the fatigued class and one for the unfatigued one. Out of the 450 signals generated, 150 was used for training the HMMs as described in Section 4.1. However, certain assumptions were involved in the parameter estimation of the HMMs. Using information from the MPD-TFR plots, three HMM states were chosen. Note that the variational state estimation works only on a truly stochastic set of signals. Since these signals have a deterministic source, VB is not the method of choice.

The observation sequences were sorted according to the time shift parameters τ. The states were depicted as instants of time where the signal time-frequency behavior did not change considerably. An example is demonstrated in Figure 4.7. Here, the vertical dotted lines (at 180 and 280 μs) represent the state separation. The observation before 180 μs is associated to state 1 of the signal, 180-280 μs is state 2 and beyond 280 μs is state 3.

The observation vectors were quantized to finite codes. We chose 16 codes (4 bit information). Thirty EM iterations were chosen for the value of $\boldsymbol{\theta}_{\mathrm{ML}}$ to converge from an initial guess to the final maximum likely estimate. The parameter estimates were chosen by multiple trials by testing the models on a validation data of size 300 (150 from each

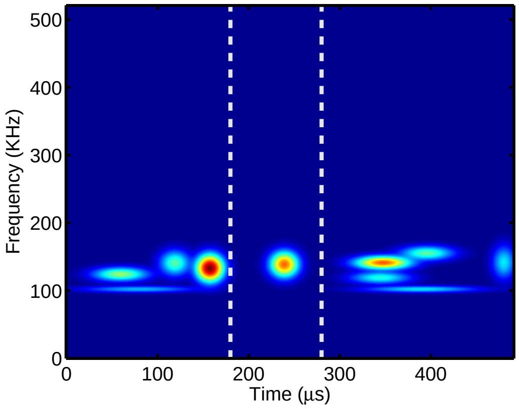


Fig. 4.7. MPD-TFR of synthesized lug joint signal at 20dB SNR, to aid in choosing number of HMM states.

class) observation sequences that were not used in the training. Model validation involved classifying (Equation (4.8)) the set of sequences and evaluating the performance of the classifiers. The parameters were tuned for optimizing the classifier in terms of correct classification rate and speed.

4.6.3. Classification Result

The performance of the classifier was tested on the remaining 150 signals from each class. Training and validation was done using the 20 dB signals and testing involved signals of all other SNRs.

The results were presented in a confusion matrix C. The entries along the diagonal of C represent the probability of correct classification for a given class, and a sum of the off-diagonal entries along a row gives the probability of miss-classification for a given class of signals. For example, Table 4.7(a) shows the confusion matrix for the dataset with SNR = 0 dB. According to this table, the unfatigued dataset has a correct classification rate of 93.3% and fatigued dataset has a correct classification rate of 73.33%. Thus, the average correct classification for signals with SNR = 0 dB is 83.33%.

	U	F
U	**0.9333**	0.0667
F	0.2667	**0.7333**

(a) SNR = 0 dB

	U	F
U	**1.0000**	0.0000
F	0.1000	**0.9000**

(b) SNR = 5 dB

	U	F
U	**1.0000**	0.0000
F	0.0533	**0.9467**

(c) SNR = 10 dB

	U	F
U	**1.0000**	0.0000
F	0.0000	**1.0000**

(d) SNR = 20 dB

Table 4.7. Confusion matrices for different SNRs (U: unfatigued, F: fatigued).

Table 4.7 summarizes the classification results. As expected, as the SNR increases, the rate of correct classification also increases. The correct classification rates at 5 dB, 10 dB and 20 dB SNRs are 95%, 97.34% and 100%, respectively. It may be observed that the performance of the classifier favored the unfatigued class.

The performance results provided were based on a specific classification threshold [24]. By varying this threshold, the probability of correct classification can be changed. A plot of the probability of correct classification versus the probability of false alarm (miss-classification) given a particular class for all possible thresholds is called a receiver operating characteristic (ROC) curve. The confusion matrix presented the performance at a specific point on this curve. The ROC curves of this classifier under the given assumption of unfatigued class is presented in Figure 4.8. By correct classification here we mean that signals actually from the damaged class are correctly classified as being from the damaged class. By false alarm here, we imply that a damage was detected when there was actually no damage. The ROC curve in Figure 4.8 also demonstrate that the classification performance improves with increasing SNR.

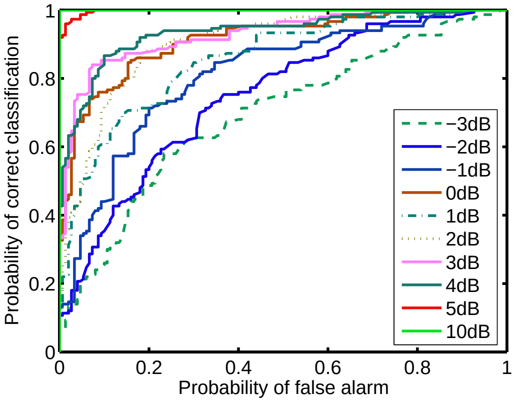


Fig. 4.8. Receiver operating characteristic (ROC) curves for different SNRs.

CHAPTER 5

Adaptive Learning using Markov Chain Monte Carlo & Estimation of Progressive Damage

Adaptive learning is a Bayesian non-parametric learning technique that learns the structure within the data without imposing any prior restrictions on the model complexity, yet penalizing over-complex models. It provides the modeling machinery needed to self-adapt to the structure within the data and allows the stochastic model to continuously evolve with the ever-changing environment.

Any set of data contains a certain structure within it. We assume that this structure can be modeled using a stochastic mixture model (e.g., Gaussian mixture model, hidden Markov mixture model) without any restrictions on the number of mixture components. Conventional parametric stochastic modeling assumes a fixed number of mixture components that may lead to over-fitting or under-fitting of the data. On the other hand, the Bayesian non-parametric modeling allows the number of mixture components to vary, and uses adequate number of mixture components needed to model the data. Such methods, by design, penalize over-complex models and avoid data over-fitting. Adaptive learning is one such approach/method to learn an optimally complex model that best fits the given data. (Here model complexity is governed by the number of mixture components present, i.e., a model with less mixture components is a less complex model).

In this chapter, we introduce the concept of Markov Chain Monte Carlo and build the necessary tools for adaptive learning on an available data set. Results from synthetic data are also presented.

5.1. Markov Chain Monte Carlo

5.1.1. Markov Chain Monte Carlo Integral

In Bayesian statistics, computing a posterior distribution often involves computationally intensive multidimensional integration. Monte Carlo methods find application in approximating these integrals. An integral can be expressed as the expectation of a distribution and then computed by approximating the expected value over that distribution by using random draws [61,101,102].

The integral of a function $f(x)$ of a variable $x \in [a, b]$,

$$\mathcal{I} = \int_a^b f(x)dx,$$

can be rewritten by factorizing $f(x) = h(x)\zeta(x)$,

$$\mathcal{I} = \int_a^b h(x)\zeta(x)dx.$$

If x is a random variable with probability density function (pdf) $\zeta(x)$ (with $\int_a^b \zeta(x)dx = 1$), the above integral can be written as

$$\mathcal{I} = \mathbf{E}\left[h(x)\right]_{\zeta(x)} = \mathbf{E}\left[h(x)|x \sim \zeta(x)\right],$$

where, $\mathbf{E}\left[\cdot\right]$ is the expectation operator defined as

$$\mathbf{E}\left[h(x)\right]_{\zeta(x)} = \int_a^b h(x)\zeta(x)dx.$$

Using L_ζ random draws x_l, $l = 1, \ldots, L_\zeta$ from $\zeta(x)$, the expected value of $h(x)$ (and thus the integral $\mathcal{I}$) can be approximated as

$$\mathcal{I} \approx \hat{\mathcal{I}} = \frac{1}{L_\zeta}\sum_{l=1}^{L_\zeta} h(x_l),$$

with $\hat{\mathcal{I}}$ almost surely approaching $\mathcal{I}$ as L_ζ approaches infinity,

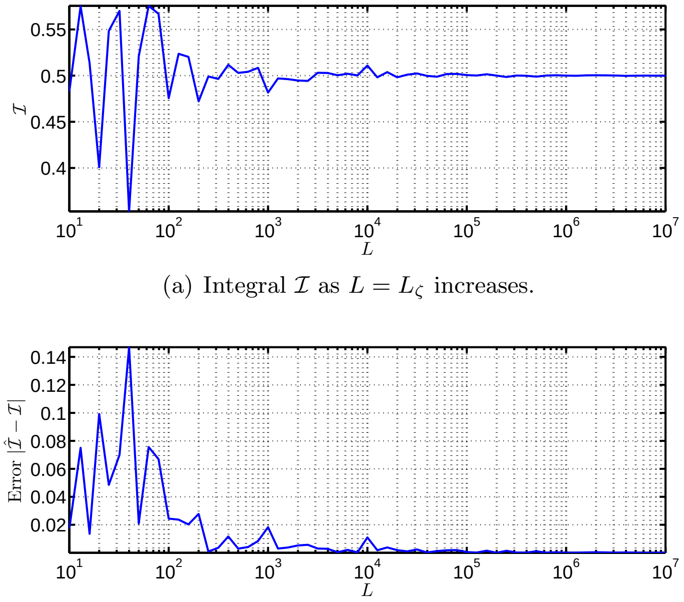
$$\lim_{L_\zeta \to \infty} \mathcal{I} - \hat{\mathcal{I}} \to 0.$$

This is called Monte Carlo integration.

To provide a specific example, let us consider the integral

$$\mathcal{I} = \int_0^\infty \sin(x) e^{-x} dx \tag{5.1}$$

which integrates to 0.5. Numerically, x_l can be drawn from an exponential distribution $\zeta(x) = e^{-x}$, and the integral can be computed as $\mathbf{E}\left[\sin(x)\right]_{\zeta(x)}$. Figure 5.1 shows the convergence of the integral to the correct result as a function of L_ζ.



(a) Integral $\mathcal{I}$ as $L = L_\zeta$ increases.

(b) Error $\left|\hat{\mathcal{I}} - \mathcal{I}\right|$ as $L = L_\zeta$ increases.

Fig. 5.1. Convergence of Monte Carlo integration for Equation (5.1).

This numerical estimation method is done is two stages: (a) sampling from a distribution and (b) computing the expectation. Among the several available approaches for approximating integrals [103–105], Markov chain Monte Carlo (MCMC) methods attempt to simulate direct draws from the distribution of interest. MCMC can also be used in empirically representing a complex distribution (or a posterior in Bayesian analysis) by samples drawn from it. In fact, MCMC provides an efficient sampling technique when samples cannot be drawn directly from $\zeta(x)$.

Markov chains are a stochastic processes that follows a Markov property, i.e., the conditional probability of future states depend only on the current state. In general a Markov chain can be described by a state transition operator A and an initial state distribution $\boldsymbol{\pi}_0$. The initial state x_0 of a Markov chain is distributed as $\boldsymbol{\pi}_0$. The successive states are distributed as $\boldsymbol{\pi}_1 = P\left(x_1|x_0\right) = A'\boldsymbol{\pi}_0$, $\boldsymbol{\pi}_2 = P\left(x_2|x_1\right) = A'\boldsymbol{\pi}_1$ and so on. Specifically, the future state distribution $\boldsymbol{\pi}_{l+1} = P\left(x_{l+1}|x_l\right)$ is conditioned on the current state as

$$\boldsymbol{\pi}_{l+1} = A'\boldsymbol{\pi}_l, \;\; l = 0, 1, 2, \ldots \tag{5.2a}$$

where A' is the adjoint of A. Iteratively at the Lth epoch, the conditional state distribution is

$$\boldsymbol{\pi}_L = (A')^L\boldsymbol{\pi}_0. \tag{5.2b}$$

If the largest eigenvalue of A' satisfies $|\lambda_{\max}| = 1$, the corresponding eigenfunction $\boldsymbol{\pi}(L)$ is the stationary distribution of the Markov chain defined by the transition operator A. Thus, a Markov chain is considered converged if $\boldsymbol{\pi}_l = \boldsymbol{\pi}_{l+1}$ up to some tolerance. This also implies that once a Markov chain has converged (say at epoch L), eigenfunction $\boldsymbol{\pi}(L)$ is the stationary distribution of the Markov chain defined by the transition operator A. Thus, a Markov chain is considered converged if $\boldsymbol{\pi}_l = \boldsymbol{\pi}_{l+1}$ up to some tolerance. This also implies that once a Markov chain has converged (say at epoch L),

$$(A')^L = (A')^{L+1}.$$

The states generated using the Markov chain prior to convergence ($l < L$ in this example) are termed *burn-in.* The successive states generated from the chain are treated as samples from the stationary distribution $\boldsymbol{\pi}(l; l > L)$.

In a discrete state Markov chain, A is a square state transition stochastic matrix with the same dimensionality as the number of states, and the initial state distribution $\boldsymbol{\pi}_0$ is a stochastic vector with the same dimensionality. With A' defined as the transpose of A, the same Equations as (5.2) holds. The stationary state distribution is given by the eigenvector that correspond to $\lambda_{\max}$, with $|\lambda_{\max}| = 1$.

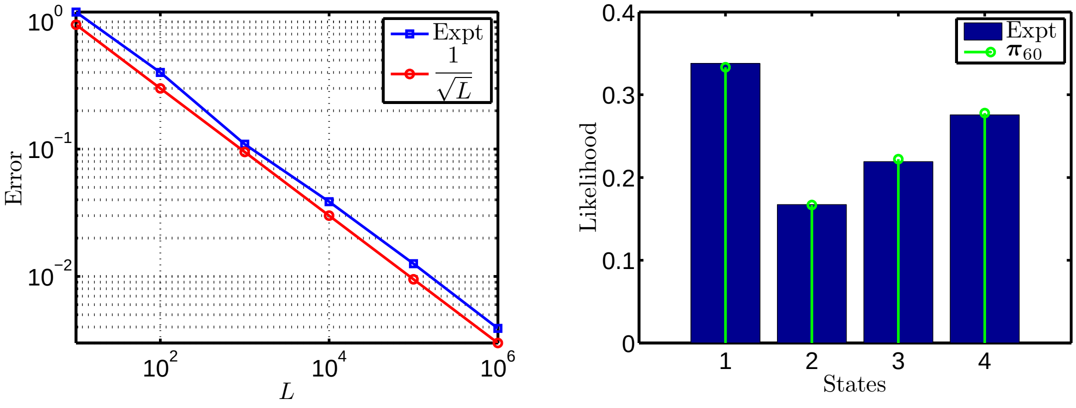
For example, let us assume a Markov chain can be in one of the four states 1, 2, 3 or 4 with a transition probability

$$A = \begin{bmatrix} 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \\ 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 \end{bmatrix}$$

i.e., given the current state 1, the probabilities of transition to the states 2 and 4 are 0.5. The initial state distribution is

$$\boldsymbol{\pi}_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Up to double precision, $A^{60} = A^{61}$. Thus, the Markov chain converges in $L = 60$ steps. Note that each row of A^{60} is the stationary distribution $\boldsymbol{\pi}_{60}$, which is also the eigenvector of A that corresponds to the eigenvalue 1. Figure 5.2(b) demonstrates that the empirical distribution computed from the simulation agrees with $\boldsymbol{\pi}_{60}$ and Figure 5.2(a) shows the convergence of



(a) Convergence rate for $P(x = 3)$.

(b) Comparing theoretical and simulated stationary distribution.

Fig. 5.2. Demonstrating that a Markov chain converges to a stationary distribution.

a point in the distribution, $P(x = 3)$. Note that more discussion on convergence can be found in [106–110].

Though we have restricted our discussion to discrete one dimensional states for simplicity, the same concept can be extended to multi-dimensional continuous states in which the state transition matrix is analogous to a state transition probability $P(\mathbf{x}_l|\mathbf{x}_{l-1})$, the eigenvectors are analogous to eigenfunctions and the state distribution probability density functions (pdfs) are continuous.

5.1.2. Gibbs Sampler

One of the applications of Monte Carlo techniques is obtaining samples from complex distributions. A well known solution is the Gibbs sampler [102, 111, 112], which is a special case of the Metropolis-Hastings sampler [59–61].

Let us assume the task of sampling from a complex D-dimensional distribution $\zeta(\mathbf{x})$, proportional to some function $f(\mathbf{x})$, where the constant of proportionality cannot be computed or is unknown. Starting with a initial value $\mathbf{x}_0$ such that $f(\mathbf{x}_0) > 0$, we draw a candidate $\hat{\mathbf{x}}$ for the sample at the next epoch from a proposal or candidate-generating distribution $P\left(\mathbf{x}_1|\mathbf{x}_0\right)$. The only constraint on the proposal distribution is that of symmetry, i.e., $P\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) = P\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$. Given $\hat{\mathbf{x}}$, compute the probability of move

$$\rho_{0\to 1} = \max\left(1, \frac{f(\hat{\mathbf{x}})}{f(\mathbf{x}_0)}\right).$$

Note that $\frac{P(\hat{\mathbf{x}})}{P(\mathbf{x}_0)} = \frac{f(\hat{\mathbf{x}})}{f(\mathbf{x}_0)}$. We accept the candidate $\mathbf{x}_1 = \hat{\mathbf{x}}$ with a probability $\rho_{0\to 1}$ or draw a new candidate. Done iteratively, the samples $\mathbf{x}_t$ given $\mathbf{x}_{t-1}$ are drawn with a probability of move

$$\rho_{(t-1)\to t} = \max\left(1, \frac{f(\hat{\mathbf{x}})}{f(\mathbf{x}_{t-1})}\right),$$

Since the sample at the tth epoch depends only on the sample at the $(t-1)$th epoch, a Markov Chain is generated. This is the Metropolis sampler. If the proposal distribution is non-symmetric, i.e. $P\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) \neq P\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$, the probability of move can be computed as [59]

$$\rho_{(t-1)\to t} = \max\left(1, \frac{f(\hat{\mathbf{x}})P\left(\hat{\mathbf{x}}|\mathbf{x}_{t-1}\right)}{f(\mathbf{x}_0)P\left(\mathbf{x}_{t-1}|\hat{\mathbf{x}}\right)}\right).$$

This procedure is known as the Metropolis-Hastings sampling algorithm and can be used to draw a Markov Chain with a target stationary distribution $\zeta(\mathbf{x}) \propto f(\mathbf{x})$.

The Gibbs sampler [111] is a special case of the Metropolis-Hastings sampler with $\rho_{(t-1)\to t} = 1$, i.e., the candidate sample is always accepted. Unlike the Metropolis-Hastings, the Gibbs sampler assumes that each draw is univariate, i.e. a D-dimensional variable is drawn by D conditional draws, each conditioned on the remaining $D-1$ components. For example, if we were to draw from a D-dimensional variable $\mathbf{x} = [x^{(1)} \ldots x^{(D)}]'$, we make $x^{(i)} \sim P\left(x^{(i)}|\mathbf{x}^{(-i)}\right)$, $i = 1, \ldots, D$ draws, where

$$\mathbf{x}^{(-i)} = \left[x^{(1)}\, x^{(2)} \ldots x^{(i-1)}\, x^{(i+1)} \ldots x^{(D)}\right]'.$$

This is advantageous because such conditional distributions are much easier to sample from compared to the joint D-dimensional one. The use of Gibbs sampler as compared to the Metropolis-Hastings is thus motivated in our application.

5.2. Learning and Inference via Markov Chain Monte Carlo

The goal of Bayesian inference is to find the posterior distribution $P\left(\boldsymbol{\theta}|\mathbb{Y}\right)$ over the parameters of interest $\boldsymbol{\theta}$ given the observed dataset $\mathbb{Y}$. MCMC methods [102] construct a Markov chain on the parameters, for which the target (stationary) distribution is the posterior conditioned on the data. In the Gibbs sampler [102], the Markov chain is obtained by iteratively sampling each random variable conditioned on the data and the previously sampled values of the other variables. Samples are then collected from the converged Markov chain and used to construct an empirical estimate of the posterior distribution over the parameters. This estimate can be used to approximate various posterior expectations of interest.

Consider a general data model

$$P\left(\mathbb{Y}|\boldsymbol{\theta},\mathbf{p}^*\right) = \sum_{m=1}^{M} p_m^* P\left(\mathbb{Y}|\Theta_m^*\right), \tag{5.3}$$

where $\mathbb{Y} = \{Y_1, \ldots, Y_T\}$ denotes an available set of T data, $\boldsymbol{\theta} = \{\Theta_1^*, \ldots, \Theta_M^*\}$ denotes a set of parameters, M represents a large number of mixture components, $p_m^* \in \mathbf{p}^*$ are the mixing weights of each component, such that $\sum_{m=1}^{M} p_m^* = 1$. If the optimum mixture size M is known, classical maximum-likelihood (ML) learning techniques can be applied to estimate the parameters of interest [96, 113]. In particular, the iterative expectation-maximization (EM) algorithm can be used to locally maximize the data likelihood [71, 95, 96]. The EM algorithm introduces auxiliary or hidden variables and iterates between inferring the posterior distribution over the hidden variables given a current parameter setting and computing a new parameter estimate by maximizing the likelihood using the learned statistics.

When the mixture size M is not available *a priori* and needs to be a part of the learning process, the ML technique is not suitable. An alternative and much more flexible approach to mixture modeling is provided by the Dirichlet process (DP) and is discussed in the following.

5.2.1. Dirichlet Process

Probabilistic models are widely used in the machine learning community to model a set of observations with a distribution. Traditional parametric models with a predetermined set of parameters limit the scope of modeling and often fraught with challenges that include over-fitting or under-fitting of data. Model selection in such learning tasks is non-trivial.

In the parametric Bayesian modeling philosophy, the observed data is assigned a prior and then the posterior is computed given the data. In general, this prior is given by a parametric family which constraints the distribution to lie within that family, limiting the scope and the type of inference that can be made. The nonparametric Bayesian approach on the other hand, provides a wide support which is typically the space of all distributions. However, for such an approach, it is important to have a tractable posterior computation. Currently, the Dirichlet process (DP) [47, 114] is one of the most popular nonparametric Bayesian models.

In general, a stochastic process [47,48,115] is a distribution over function spaces whose sample paths are random functions drawn from the distribution. On the other hand, a draw from a DP is a function with some special properties which allow it to be a distribution in a certain probability space $\mathbb{U}$. In fact, a DP can be called *a distribution over distributions* [116] in $\mathbb{U}$.

A DP can be characterized by a base distribution G_0 and an innovation parameter ι. If a random distribution G is distributed according to DP

$$G \sim DP\left(\iota, G_0\right), \tag{5.4}$$

the marginal distributions are also Dirichlet distributed. If we consider a set of finite large M measurable partitions $\Theta_m^* \in \mathbb{U}$, $k = 1, \ldots, M$ then

$$\Theta_n \sim G, \;\; n = 1, \ldots, \infty, \tag{5.5}$$

where Θ_n are non-unique draws from $\mathbb{U}$. The expected distribution of $G(\Theta)$ is $G_0(\Theta^*)$. For simplicity of notation, in the remaining document we represent $G(\Theta)$ as G and $G_0(\Theta^*)$ as G_0. The innovation parameter ι can be interpreted as the inverse variance of G. Since ι provides a measure of concentration of mass around the mean, as $\iota \to \infty$ we can weakly infer that $G \to G_0$. This inference is weak or point wise as G is almost surely discrete (discrete with probability 1) [114,116] though G_0 can be smooth.

Introducing the vector $\boldsymbol{\theta} = \{\Theta_1, \Theta_2, \ldots, \Theta_T\}$ where every Θ_i, $i = 1, \ldots, T$, is a draw from G, the distribution of $\boldsymbol{\theta}$ can be expressed [47,48,117–119] by integrating out G as

$$\begin{aligned} \mathrm{P}\left(\boldsymbol{\theta}|\iota, G_0\right) &= \int \mathrm{P}\left(\boldsymbol{\theta}|G, \iota, G_0\right) DP\left(\iota, G_0\right) dG \\ &= \int \mathrm{P}\left(\boldsymbol{\theta}|G, \iota, G_0\right) P\left(G|\iota, G_0\right) dG \end{aligned} \tag{5.6}$$

Further, if $\boldsymbol{\theta}_{-i} = \{\Theta_1, \ldots, \Theta_{i-1}, \Theta_{i+1}, \ldots, \Theta_T\}$ contains all the draws except Θ_i, the conditional distribution of Θ_i can be shown to obey the Pólya Urn property [47–49,117,120]

$$\mathrm{P}\left(\Theta_i|\boldsymbol{\theta}_{-i}, \iota, G_0(\Theta_i)\right) = \frac{1}{T+\iota-1}\sum_{k=1, i\neq k}^{T} \delta_{\Theta_i,\Theta_k} + \frac{\iota}{T+\iota-1} G_0(\Theta_i). \tag{5.7}$$

Note that the ordering of the Θ_i in $\boldsymbol{\theta}$ plays no role. If Θ^*_m, $m = 1, \ldots, M$ is used to denote the unique values of $\boldsymbol{\theta}$, and $N^i_{\Theta_m}$ denotes the number of times Θ^*_m appears in $\boldsymbol{\theta}_{-i}$ the above equation can be written as

$$\mathrm{P}\left(\Theta_i|\boldsymbol{\theta}_{-i}, \iota, G_0(\Theta_i)\right) = \frac{1}{T+\iota-1}\sum_{m=1}^{M} N^i_{\Theta_m}\delta_{\Theta_i,\Theta^*_m} + \frac{\iota}{T+\iota-1}G_0(\Theta_i). \tag{5.8}$$

In other words, each Θ_i assumes an existing value from $\boldsymbol{\theta}_{-i}$ with probability $\frac{N^i_{\Theta_m}}{T+\iota-1}$, or it is a fresh draw from G_0 with probability $\frac{\iota}{T+\iota-1}$. It can be shown that M can grow as $\mathcal{O}(\log T)$, and the DP prior is therefore non-parametric.

5.2.2. Stick Breaking Prior

An explicit characterization of G is provided in [116] as the stick breaking construction. Imagine that $G(\boldsymbol{\theta})$ is like a stick of unit length which will be broken into pieces such that the larger pieces are more likely to be broken. The length of these pieces are $p^*_m = \mathrm{P}\left(\Theta_i = \Theta^*_m\right)$. We can write G as an infinite mixture model,

$$G = \sum_{m=1}^{\infty} p^*_m \delta_{\Theta,\Theta^*_m}, \tag{5.9a}$$

where,

$$\Theta^*_m \sim G_0, \;\; m = 1, \ldots, \infty \tag{5.9b}$$

$$p^*_m = \left[\prod_{k=1}^{m-1}(1-v_k)\right] v_m \tag{5.9c}$$

$$v_k \sim \mathit{Beta}\left(1, \iota\right), \;\; k = 1, \ldots, \infty. \tag{5.9d}$$

5.2.3. Dirichlet Process Mixture Model

Let us consider a set of observed data $\mathbb{Y} = \{Y_1, \ldots, Y_T\}$ generated from a pdf $P\left(Y_n|\Theta_n\right)$ such that a one-to-one correspondence exists between every Y_n and Θ_n. Then, $\boldsymbol{\theta} = \{\Theta_1, \ldots, \Theta_T\}$ is the (latent) set of parameters from which the data is generated. Given the stick breaking construction, the DP can be used as a nonparametric prior in a hierarchical Bayesian model

$$G \sim DP(\iota, G_0) \tag{5.10a}$$

$$\Theta_n|G \sim G, \;\; n = 1, \ldots, T \tag{5.10b}$$

$$Y_n|\Theta_n \sim P(Y_n|\Theta_n), \;\; n = 1, \ldots, T \tag{5.10c}$$

Since the parameters are drawn from G and take on coincident values, there is a natural clustering among the (unobserved) Θ_n, which in turn leads to clustering in the data. In view of the stick breaking construction provided earlier, an infinite DP mixture model is imposed on the data,

$$P(Y|\mathbf{p}^*, \boldsymbol{\theta}^*) = \sum_{m=1}^{\infty} p_m^* P(Y|\Theta_m^*), \tag{5.11}$$

where $\mathbf{p}^* = \{p_1^*, \ldots, p_\infty^*\}$. This is summarized in Algorithm 2, where c_n gives the class membership of any data Y_n, i.e. $c_n \in \{1, 2, 3, \ldots\}$.

5.2.4. Pólya Urn Gibbs Sampler for Dirichlet Process

According to [118, 119] one can generate a Gibbs sampler to draw the Θ_i's. If $X^{(k)}$ is the the kth state in a realization of a Markov chain

$$X^{(0)} \rightarrow X^{(1)} \rightarrow \ldots \rightarrow X^{(k)} \rightarrow \ldots,$$

the state transition probability can be defined as $P\left(X^{(k)} \rightarrow X^{(k+1)}\right) = \mathrm{P}(\Theta_k|\boldsymbol{\theta}_{-k}, Y_k, \iota, G_0)$, where $\mathrm{P}(\Theta_k|\boldsymbol{\theta}_{-k}, Y_k, \iota, G_0)$ is the DP posterior [115] on any Θ_k. The parameters are not independent since they are draws from a DP, but the observations are actually independent.

Algorithm 2 Stick breaking construction of infinite DPMM.

```
$\Theta^*_m$ are drawn from DP base distribution, $G_0$
    $$\Theta^*_m \sim G_0, \;\; m = 1, \ldots, \infty \tag{5.12a}$$
Independent Beta random variables $v_k$ are drawn
    $$v_k \sim \; \mathit{Beta}\,(1, \iota)\,, \;\; k = 1, \ldots, \infty \tag{5.12b}$$
Compute the mixing weights of the DPMM
    $$p^*_m = \left[\prod_{k=1}^{m-1} (1 - v_k)\right] v_m, \;\; m = 1, \ldots, \infty \tag{5.12c}$$
The class membership of the mixture components are drawn according to the multinomial distribution given by $\mathbf{p}^* = \{p^*_1, \ldots, p^*_\infty\}$
    $$c_n | \mathbf{p}^* \sim \mathit{Mult}\,(\mathbf{p}^*)\,, \;\; n = 1, \ldots, T \tag{5.12d}$$
Data is obtained from a non-parametric distribution
    $$Y_n | c_n \sim P\left(Y_n | \Theta^*_{c_n}\right), \;\; n = 1, \ldots, T \tag{5.12e}$$
```

The dependencies in the following is shown by $\Leftrightarrow$.

$$\begin{tikzcd}
Y_1 \arrow[d, Leftrightarrow] & Y_2 \arrow[d, Leftrightarrow] & Y_3 \arrow[d, Leftrightarrow] & \cdots & Y_T \arrow[d, Leftrightarrow] \\
\Theta_1 \arrow[r, Leftrightarrow] & \Theta_2 \arrow[r, Leftrightarrow] & \Theta_3 \arrow[r, Leftrightarrow] & \cdots \arrow[r, Leftrightarrow] & \Theta_T
\end{tikzcd}$$

Based on the fact that the parameters are drawn from the stick breaking prior, we can systematically write the conditional posterior on the parameters as

$$\mathrm{P}\left(\Theta_k | \boldsymbol{\theta}_{-k}, Y_k, \iota, G_0\right) = \sum_{i=1, i \neq k}^{T} q_i \delta_{\Theta_k, \Theta_i} + q_0 P\left(Y_k | \Theta_k\right) G(\Theta_k), \tag{5.13}$$

where

$$\begin{aligned} q_i &= bP\left(Y_k | \Theta_k\right) \qquad (5.14a) \\ q_0 &= b_0 \iota \int P\left(Y_k | \Theta\right) G_0(\Theta) d\Theta. \qquad (5.14b) \end{aligned}$$

Here, q_0 and q_i are normalizing constants,

$$\sum_{i=1}^{T} q_i + q_0 = 1. \tag{5.15}$$

Computing q_0 can be made simple if $G_0(\Theta_i)$ and $P\left(\Theta_i|Y_i\right)$ are conjugate pairs for the likelihood $P\left(Y_i|\Theta_i\right)$. In Equation (5.13), each parameter can be drawn conditioned on the remaining parameters and the available data iteratively from $\mathrm{P}\left(\Theta_k|\boldsymbol{\theta}_{-k}, Y_k, \iota, G_0\right)$. This iterative drawing represents Gibbs sampling. Observe that larger values of q_0 induces innovation. In fact, the DP innovation parameter ι and q_0 share a monotonic relation. If the values of q_0 get smaller than q_i, the probability of drawing newer value of unique parameters will reduce and the algorithm will settle for the existing unique parameter set indicating that no new mixture components are likely to be formed.

These samples realize a Markov chain which is schematically represented in the following.

$$\begin{tikzcd}
X^{(0)} \arrow[r] \arrow[d, Leftrightarrow] & X^{(1)} \arrow[r] \arrow[d, Leftrightarrow] & X^{(2)} \arrow[r] \arrow[d, Leftrightarrow] & \cdots \arrow[r] & X^{(T)} \arrow[r] \arrow[d, Leftrightarrow] & \cdots \\
\boldsymbol{\theta}^{(0)} & \boldsymbol{\theta}^{(1)} & \boldsymbol{\theta}^{(2)} & & \boldsymbol{\theta}^{(T)} &
\end{tikzcd} \tag{5.16}$$

Here, each state being associated to a unique T-dimensional $\boldsymbol{\theta}$. Specifically, a state transition in this Markov chain is associated to updating a Θ_i, conditioned on $\boldsymbol{\theta}_{-i}$ and data Y_i, as discussed below.

$$\begin{tikzcd}
X^{(0)} \arrow[r] \arrow[d, Leftrightarrow] & X^{(1)} \arrow[r] \arrow[d, Leftrightarrow] & X^{(2)} \arrow[r] \arrow[d, Leftrightarrow] & \cdots \arrow[r] & X^{(T)} \arrow[r] \arrow[d, Leftrightarrow] & \cdots \\
\begin{bmatrix} \Theta_1^{(0)} \\ \Theta_2^{(0)} \\ \vdots \\ \Theta_T^{(0)} \\ \vdots \end{bmatrix} & \begin{bmatrix} \Theta_1^{(1)} \\ \Theta_2^{(0)} \\ \vdots \\ \Theta_T^{(0)} \\ \vdots \end{bmatrix} & \begin{bmatrix} \Theta_1^{(1)} \\ \Theta_2^{(1)} \\ \vdots \\ \Theta_T^{(0)} \\ \vdots \end{bmatrix} & \cdots & \begin{bmatrix} \Theta_1^{(1)} \\ \Theta_2^{(1)} \\ \vdots \\ \Theta_T^{(1)} \\ \vdots \end{bmatrix} &
\end{tikzcd} \tag{5.17}$$

In the above schematic $X^{(0)} \rightarrow X^{(1)}$ transition is achieved by updating $\Theta_1^{(0)}$ to $\Theta_1^{(1)}$, $X^{(1)} \rightarrow X^{(2)}$ is achieved by updating $\Theta_2^{(0)}$ to $\Theta_2^{(1)}$ and so on using Equation (5.13). This is called the Pólya Urn Gibbs sampling.

The Pólya Urn sampler, though a very versatile technique, has several limitations of which the following are significant [115].

i *Convergence* — In this approach, the conditional distribution of each parameter set Θ_i relies on the others $\boldsymbol{\theta}_{-i}$, which compels updating $\boldsymbol{\theta}$ one component at a time. This makes the progress of the chain and learning the mixture slow. In addition, the value of q_i grows and that inhibits drawing of new parameters and slows down the mixing of the Markov chain.

ii *Conjugate priors* — Conjugate priors make computing q_0 in Equation (5.14b) relatively simpler. However, not all models on the data can have conjugate priors. There are some proposed algorithms [121] when $G_0(\Theta)$ is not the conjugate prior for $P\left(Y_k|\Theta\right)$, but they are non-trivial.

An approximate but computationally efficient approach is proposed in [115], called the blocked Gibbs sampler which is discussed in the following section.

5.2.5. Blocked Gibbs Sampler

In the blocked Gibbs sampler, the infinite mixture is truncated to M components (also known as clusters) such that the truncated mixture is a close approximation of the true infinite component mixture model. This truncated model has several benefits [115] including computational benefits and can be expressed as an alternative to (5.9a),

$$G = \sum_{k=1}^{M} p_k \delta_{\Theta_i,\Theta_k^*}. \tag{5.18}$$

Here $\delta_{\Theta_i,\Theta_k^*}$ is a Kronecker delta function and is defined as

$$\delta_{\Theta_i,\Theta_k^*} = \begin{cases} 1, & \Theta_i = \Theta_k^* \\ 0, & \Theta_i \neq \Theta_k^* \end{cases}.$$

Note that p_k and p_k^* are not the same, but

$$\sum_{k=1}^{\infty} p_k^* \delta_{\Theta_i,\Theta_k^*} = \sum_{k=1}^{M} p_k \delta_{\Theta_i,\Theta_k^*} = 1.$$

Also, p_k^* is computed as shown in (5.9c) but,

$$p_k = \frac{\text{number of } \Theta_k = \Theta_k^*}{T}.$$

If the truncation limit M is sufficiently large (but finite), the mixture in Equation (5.9a) can be approximated by the one in Equation (5.18) by approximating $p_k \approx p_k^*$

for $k = 1, \ldots, M$ with a 1-norm error

$$\begin{aligned} \mathcal{E} &= \left\| \sum_{k=1}^{M} p_k^* \delta_{\Theta_i, \Theta_k^*} - \sum_{l=1}^{\infty} p_l^* \delta_{\Theta_i, \Theta_l^*} \right\|_1 \\ &= 4Me^{-\frac{M-1}{\iota}}. \end{aligned} \tag{5.19}$$

The error decreases exponentially as M increases, but increases with increasing ι. Figure 5.12 demonstrates the values of the error for a range of values of M and ι. The choice of M is critical in this approach. A poor choice of M can lead to undesired modeling error. If M is too small, it may not be enough to describe the data adequately, and if M is too large there will be wasted computation.

In this approach, the data model provided in (5.3) is assumed to have a reasonable number of mixture components and the parameter sets contain finite and predetermined number of parameters $\boldsymbol{\theta}^* = \{\Theta_0, \ldots, \Theta_M\}$ and $\mathbf{p}* = \{p_1^*, \ldots, p_M^*\}$. If the true number of mixtures are less than M, the redundant components receive a very small (nearly 0) mixing weight. The update of the Markov chain

$$\begin{tikzcd} X^{(0)} \arrow[r] \arrow[d, Leftrightarrow] & X^{(1)} \arrow[r] \arrow[d, Leftrightarrow] & X^{(2)} \arrow[r] \arrow[d, Leftrightarrow] & \cdots \arrow[r] & X^{(T)} \arrow[r] \arrow[d, Leftrightarrow] & \cdots \\ \boldsymbol{\theta}^{*(0)} & \boldsymbol{\theta}^{*(1)} & \boldsymbol{\theta}^{*(2)} & & \boldsymbol{\theta}^{*(T)} & \end{tikzcd} \tag{5.20}$$

is provided by updating $\boldsymbol{\theta}^{*(l)} \rightarrow \boldsymbol{\theta}^{*(l+1)}$ and not component wise. The update schematic can rewritten as

$$
\begin{array}{ccccccccccc}
X^{(0)} & \rightarrow & X^{(1)} & \rightarrow & X^{(2)} & \rightarrow & \cdots & \rightarrow & X^{(T)} & \rightarrow & \cdots \\
 & & & & & & & & & & \\
\Updownarrow & & \Updownarrow & & \Updownarrow & & & & \Updownarrow & & \\
 & & & & & & & & & & \\
\begin{bmatrix} \Theta_1^{*(0)} \\ \Theta_2^{*(0)} \\ \vdots \\ \Theta_M^{*(0)} \end{bmatrix} & & \begin{bmatrix} \Theta_1^{*(1)} \\ \Theta_2^{*(1)} \\ \vdots \\ \Theta_M^{*(1)} \end{bmatrix} & & \begin{bmatrix} \Theta_1^{*(2)} \\ \Theta_2^{*(2)} \\ \vdots \\ \Theta_M^{*(2)} \end{bmatrix} & & \cdots & & \begin{bmatrix} \Theta_1^{*(T)} \\ \Theta_2^{*(T)} \\ \vdots \\ \Theta_M^{*(T)} \end{bmatrix} & &
\end{array}
\tag{5.21}
$$

For an available set of data $\mathbb{Y} = \{Y_1, Y_2, \ldots, Y_T\}$, the blocked Gibbs sampler is summarized in Algorithm 3.

5.3. Learning Gaussian Mixture Model using Markov Chain Monte Carlo

We can use the blocked Gibbs sampler with DP prior to learn the clustering present in data $\mathbb{Y} = \{y_1, y_2, \ldots, y_T\}$. Let us assume that the data can be modeled using a 1-D Gaussian mixture model (GMM)

$$
\mathcal{G}(\mathbb{Y}|\boldsymbol{\theta}) = \sum_{m=1}^{\tilde{M}} p_m^* \mathcal{N}\left(\mu_m^*, \sigma_m^{2*}\right), \tag{5.23}
$$

where each unique parameter Θ_m^* is comprised of the mean μ_m^* and the variance σ_m^{*2} of the mth Gaussian distribution in the mixture. The parameter set $\boldsymbol{\theta} = \{\Theta_1, \Theta_2, \ldots, \Theta_T\}$ correspond to the data $\mathbb{Y}$. The correct association of Θ_is to the observation will result in clustering according to the $\tilde{M}$ groups suggested by the unique parameters Θ_m^*, $m = 1, \ldots, \tilde{M}$. The clustering of $\mathbb{Y}$ can be learned using a DP prior(G_0) on the parameters $\boldsymbol{\theta}$. The learning can be conducted by constructing a Markov chain.

Algorithm 3 Blocked Gibbs Sampling.

```
1: Given the conditional distribution over the parameters at $l$th epoch,

   $P\left(\Theta_m^{*(l)}|\mathbf{c},\mathbb{Y}\right) = \begin{cases} G_0(\Theta_m^{*(l-1)})\prod_{t=1}^{T} P\left(Y_t|\Theta_m^{*(l-1)}\right), & c_t = m \text{ for each } m \\ G_0(\Theta_m^{*(l-1)}), & c_t \neq m \text{ for all } m \end{cases},$

   draw
   $\Theta_m^{*(l)} \sim P\left(\Theta_m^{*(l-1)}|\mathbf{c},\mathbb{Y}\right) \ \forall m = 1,\, 2,\, \ldots,\, M \qquad (5.22a)$

2: Given the distribution over a cluster index $c_i \in \{1,\, 2,\, \ldots,\, M\}$,

   $\Pr\left(c_i^{(l)}|\boldsymbol{\theta},\mathbf{p}^*,\mathbb{Y}\right) = \sum_{m=1}^{M} p_m^* P\left(Y_i|\Theta_m^*\right)\delta_{c_i^{(l-1)},m},$

   draw the new cluster indices
   $c_i^{(l)} \overset{\text{ind}}{\sim} \Pr\left(c_i^{(l)}|\boldsymbol{\theta},\mathbf{p}^*,\mathbb{Y}\right) \ \forall i = 1,\, 2,\, \ldots,\, T \qquad (5.22b)$

3: Re-estimate the number of repeating parameters in group $m$,

   $N_{\Theta_m^*}^{(l)} = \sum_{k=1}^{T} \delta_{\Theta_m^*,\Theta_k} \ \forall m = 1,\, 2,\, \ldots,\, M \qquad (5.22c)$

4: Draw

   $v_m^{(l)} \sim \begin{cases} Beta\left(1 + N_{\Theta_m^*}\ ,\ \iota + \sum_{k=j+1}^{M} N_{\Theta_k^*}\right), & m = 1,\, \ldots,\, M-1 \\ v_M = 1, & m = M \end{cases} \qquad (5.22d)$

5: Update the mixing weights

   $p_m^{*(l)} = \begin{cases} v_m, & m = 1 \\ v_m \prod_{k=1}^{m-1}(1 - v_k), & m = 2,\, 3,\, \ldots,\, M \end{cases} \qquad (5.22e)$

6: Iterate over the above steps up to some large epoch $l = L$ when the Markov chain is
   believed to have converged. Iterating further $l > L$ draws *(burn-out) samples from the
   target distribution $P\left(\boldsymbol{\theta}^*,\mathbf{p}^*,\mathbf{c}|\mathbb{Y}\right)$, or $P\left(\boldsymbol{\theta}^*|\mathbb{Y}\right)$ for simplicity.*
7: *The predictive likelihood on the data can be computed after $L_{\text{out}}$ burn-out samples as*

   $P\left(Y_{T+1}|\mathbb{Y},\iota,G_0\right) = \frac{1}{L_{\text{out}}}\sum_{l=1}^{L_{\text{out}}}\left[\sum_{m=1}^{M} p_m^{*(l)} P\left(Y_{T+1}|\theta^{*(l)}\right)\right] \qquad (5.22f)$
```

Assumptions are made to appropriately choose a base distribution G_0. Following this, we attempt to find the actual values of the parameter sets Θs that best defines $\mathcal{G}$ by iteratively drawing samples from a Markov chain using a blocked Gibbs sampler presented in Algorithm 3.

The possible number of mixtures chosen M is much larger than the actual mixture components in $\mathcal{G}$, $M \gg \tilde{M}$. In this application, the base distribution in chosen a Normal-Gamma

$$\begin{aligned} P\left(\Theta_m^*\right) &= P\left(\mu_m^*, \frac{1}{\sigma_m^{2*}} \middle| u_\mu, u_\tau, u_a, u_b\right) \\ &= G_0\left(\mu_m^*, \frac{1}{\sigma_m^{2*}} \middle| u_\mu, u_\tau, u_a, u_b\right) \\ &= \mathcal{N}Gamma\left(\mu_m^*, \frac{1}{\sigma_m^{2*}} \middle| u_\mu, u_\tau, u_a, u_b\right) \\ &= \mathcal{N}\left(\mu_m^* \middle| u_\mu, \frac{\sigma_m^{2*}}{u_\tau}\right) Gamma\left(\frac{1}{\sigma_m^{2*}} \middle| u_a, u_b\right), \end{aligned}$$

where u_a, u_b, u_τ and u_μ are the hyper-parameters of the Normal-Gamma distribution. For all the data $\mathbb{Y}_m \subset \mathbb{Y}$ belonging to any cluster m, $m = 1, \ldots, M$, the distribution of data is $P\left(y_m | \mu_m^*, \sigma_m^{2*}\right)$. By conjugacy of prior, the posterior distribution of the Θ_m^* conditioned on the data from the same cluster is a Normal-Gamma with different hyper-parameters

$$\begin{aligned} P\left(\Theta_m^* | \mathbb{Y}_m\right) &= P\left(\mu_m^*, \frac{1}{\sigma_m^{2*}} \middle| \mathbb{Y}_m\right) \\ &\propto P\left(\mathbb{Y}_m | \mu_m^*, \sigma_m^{2*}\right) P\left(\mu_m^*, \frac{1}{\sigma_m^{2*}} \middle| u_\mu, u_\tau, u_a, u_b\right) \\ &= \mathcal{N}Gamma\left(\mu_m^*, \frac{1}{\sigma_m^{2*}} \middle| \mathfrak{u}_\mu, \mathfrak{u}_\tau, \mathfrak{u}_a, \mathfrak{u}_b\right), \end{aligned}$$

where

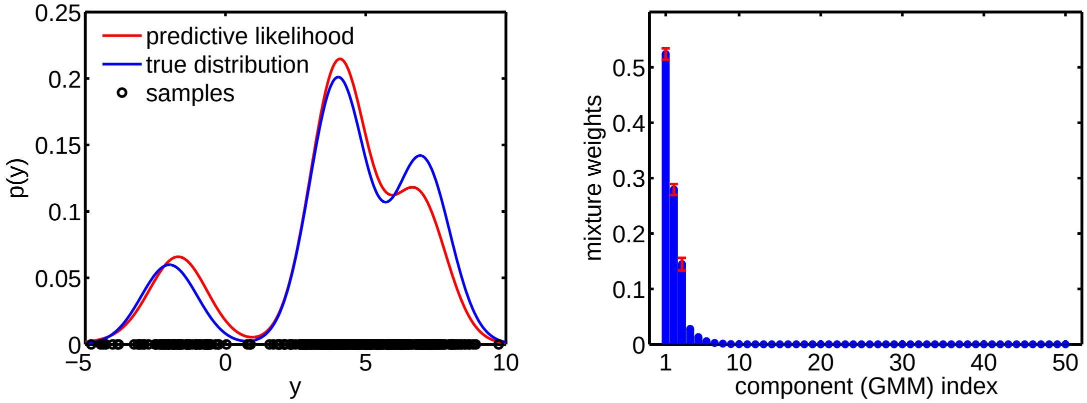
$$\begin{aligned}
\mathfrak{u}_\mu &= \frac{u_\tau u_\mu + T_m \hat{\mu}_m}{u_\tau + T_m}, \\
\mathfrak{u}_\tau &= \frac{\hat{\sigma}_m^2}{u_\tau + T_m}, \\
\mathfrak{u}_a &= u_a + \frac{T_m}{2}, \\
\mathfrak{u}_b &= \left(\frac{1}{u_b} + \frac{T_m \hat{\sigma}_m^2}{2} + \frac{u_\tau T_m (\hat{\mu}_m - u_\mu)^2}{2(u_\tau + T_m)} \right)^{-1},
\end{aligned}$$

are the hyperparameters of the posterior Normal-Gamma distribution. Here, $\hat{\mu}_m$ and $\hat{\sigma}_m^2$ are the sample mean and sample variance estimates of $\mathbb{Y}_m$, respectively and $T_m = |\mathbb{Y}_m|$ is the size of set $\mathbb{Y}_m$.

When converged, the Markov chain yields samples from the posterior $P(\boldsymbol{\theta}, \mathbf{c}, \mathbf{p}^* | \mathbb{Y})$ that we seek, and $p_m \approx 0$ for $m > \tilde{M}$. Note that multi-dimensional data could be modeled by common structures like a multi-dimensional GMM or a HMM mixture. Appendix E provides the necessary equations.

Now we demonstrate the learning of parameters using MCMC with DP prior using synthetic data from a Gaussian mixture model. Data are drawn from a $\tilde{M} = 3$ component GMM. Five hundred samples were used for this experiment. A burn in $L_B = 2000$ is used followed by a burn out of 1000 samples from the Markov chain. The truncation limit on the blocked Gibbs sampler is $M = 50$. This poses a total of 150 unknown parameters to estimate (50 means, 50 variances and 50 mixing weights).

Figure 5.3 shows that the MCMC converged (with some tolerance) to the correct distribution and describes the data adequately. We could correctly identify the 3 groups (3 out of 50 mixing weights were non-zero) in the data that correspond to the 3 mixture components in the GMM from which the data was drawn. This simulation took about 1.1



(a) True distribution and the learned predictive likelihood.

(b) Mixture weight estimates with 2σ confidence intervals.

Fig. 5.3. Blocked Gibbs sampling for learning a 1-dimensional Gaussian Mixture Model using Dirichlet Process.

minutes to execute on a 2 GHz processor.

5.4. Adaptive Learning Framework for Progressive Damage Estimation

We used an extension of the adaptive learning method described above to formulate a progressive damage estimation framework. In our approach, we used the Dirichlet process mixture model as the prior for the Bayesian non-parametric estimation and deployed adaptive learning to learn the posterior distribution over the parameters of the data model. In this application, we incorporated a dynamic environment and an evolving set of data.

The proposed progressive damage estimation technique relied on the assumption that data is continuously made available. The block diagram in Figure 5.4 summarizes the steps involved in this method. Data was obtained from the sensors, preprocessed and feature extracted using matching pursuit decomposition (MPD) (see Section 3.1). This was further extended to create an MPD based probability density function (MPD-PDF), which was used to compress the multi-dimensional signal to a one-dimensional (1-D) feature. This was achieved by computing a statistical measure of similarity from a reference set of MPD-PDFs,

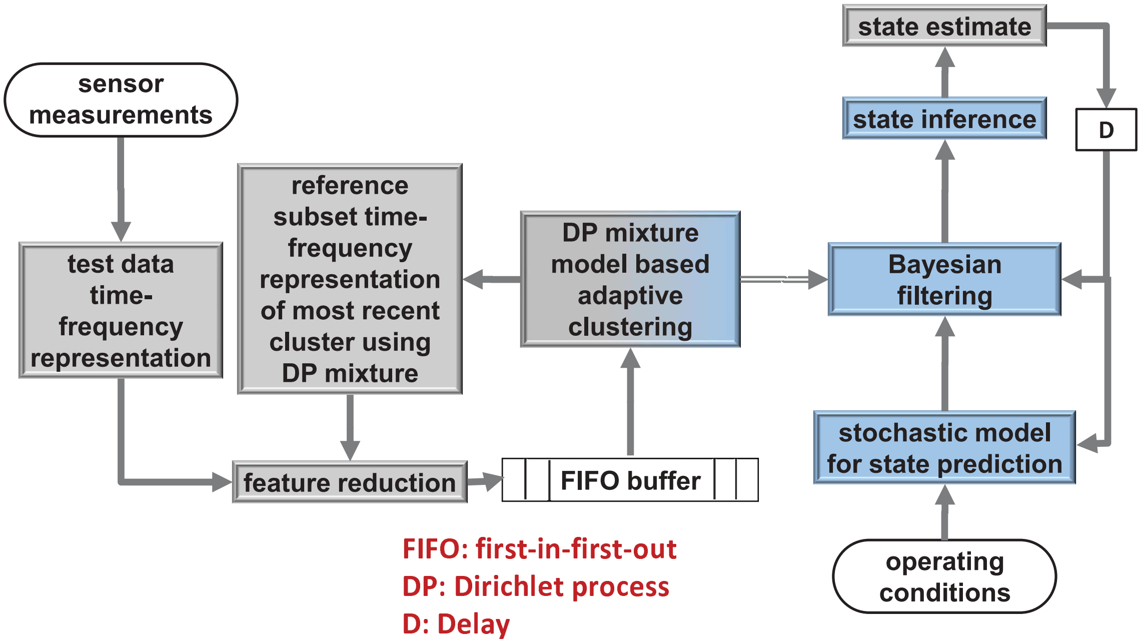

state estimate
state inference
D
sensor measurements
reference subset time-frequency representation of most recent cluster using DP mixture
DP mixture model based adaptive clustering
Bayesian filtering
test data time-frequency representation
stochastic model for state prediction
feature reduction
FIFO buffer
FIFO: first-in-first-out
DP: Dirichlet process
D: Delay
operating conditions

Fig. 5.4. Block diagram showing progressive damage estimation using adaptive learning.

chosen actively using an active data selection algorithm.

The 1-D features were stored and refreshed in a first-in-first-out (FIFO) buffer, i.e., the oldest feature was replaced by the most recent one. The number of clustered needed to most reasonably represent the features in the buffer was adaptively learned. More than one clusters implies the occurrence of change in the features, thereby implying occurrence of change in the data, which can either be due to the change in damage state or change in environmental/operational conditions. A physics-based estimate of damage in conjunction to the adaptive learning were used in a Bayesian framework to estimate the damage state. In this section, the individual concepts are discussed in detail, followed by results on actual data.

5.4.1. Matching Pursuit Decomposition based Probability Density Function

The MPD of each signal provided us with a 4-D observation sequence (with sequence length same as the MPD iterations) from time-shifts (τ), frequency shifts (ν), scales (κ) and MPD coefficients (α). These features were used to create a time-frequency representation by the linear combination of the Wigner distributions (WD) of the chosen dictionary atoms as discussed in Section 3.2.

We emphasize that, at the nth MPD iteration, the Gaussian function $g_n(t)$ was chosen with corresponding features $(\tau_n, \nu_n, \kappa_n)$ whose WD is the 2-D Gaussian function in Equation (3.6). This WD can also be interpreted as a 2-D Gaussian probability density function with mean vector $\begin{bmatrix} \tau_n \\ \nu_n \end{bmatrix}$ and covariance matrix $\begin{bmatrix} \frac{1}{4\kappa_n} & 0 \\ 0 & \frac{\kappa_n}{4\pi^2} \end{bmatrix}$. Thus, we could define the MPD-TFR as a MPD based probability density function (MPD-PDF) for the measurement $s(t)$ as

$$P_s(t,f) \triangleq \frac{1}{Z} \left| \mathcal{E}_s(t,f) \right|, \tag{5.24}$$

where $Z = \sum_{n=0}^{N-1} |\alpha_n|^2$ is the normalizing constant. For a dictionary of Gaussian functions, the MPD-PDF assumes the form of a Gaussian mixture model (GMM) defined as

$$P_s(t,f) = \frac{1}{Z} \sum_{n=0}^{N-1} |\alpha_n|^2 \mathcal{N} \left(\begin{bmatrix} \tau_n \\ \nu_n \end{bmatrix}, \begin{bmatrix} \frac{1}{4\kappa_n} & 0 \\ 0 & \frac{\kappa_n}{4\pi^2} \end{bmatrix} \right), \tag{5.25}$$

where $\mathcal{N}(\cdot,\cdot)$ is a Gaussian distribution parameterized by a mean and a covariance.

The MPD-PDF maps the signals to two-dimensional (time-frequency) pdfs and provides a versatile tool for computing statistical similarity measures between the signals. In reference to Figure 5.4, we can denote the set of test MPD-PDFs by $\mathbb{P}^{\text{te}}$ and the set of reference MPD-PDFs by $\mathbb{P}^{\text{ref}}$. Next, we can compute the statistical similarity measure corresponding to each MPD-PDF in $\mathbb{P}^{\text{te}}$ from $\mathbb{P}^{\text{ref}}$.

5.4.2. Statistical Measure of Similarity

The statistical similarity between two pdfs can be quantified by well known measures such as Kullback-Leibler distance (KLD) [96, 122, 123], Bhattacharyya distance [124, 125], and Hellinger distance [126]. Among these, the most popular is the KLD defined for pdfs $\mathfrak{p}(t,f)$ and $\mathfrak{q}(t,f)$ as[1]

$$\mathcal{D}_{\text{KL}}(\mathfrak{p}||\mathfrak{q}) \triangleq \iint \mathfrak{p}(t,f) \log \frac{\mathfrak{p}(t,f)}{\mathfrak{q}(t,f)} \, dt \, df, \tag{5.26}$$

provided $\mathfrak{p}(t,f) > 0$ and $\mathfrak{q}(t,f) > 0$ for all values of t and f. Note that $\mathcal{D}_{\text{KL}}(\mathfrak{p}||\mathfrak{q}) \geq 0$, with equality if $\mathfrak{p}(t,f) = \mathfrak{q}(t,f)$. However, this distance is non-symmetric, i.e., $\mathcal{D}_{\text{KL}}(\mathfrak{p}||\mathfrak{q}) \neq \mathcal{D}_{\text{KL}}(\mathfrak{q}||\mathfrak{p})$, and its computation involves evaluating logarithms, which is expensive. In [46, 127], a correlation based distance

$$\mathcal{D}^{\text{corr}}(\mathfrak{p}||\mathfrak{q}) \triangleq \iint \mathfrak{p}(t,f) \, \mathfrak{q}(t,f) \, dt \, df \tag{5.27}$$

is considered. This distance measure has the following properties.

(a) This is linear measure of statistical similarity.

(b) It is defined for any distributions $\mathfrak{p}(t,f)$ and $\mathfrak{q}(t,f)$ for all values of (t,f).

(c) The distance is bounded $0 \leq \mathcal{D}^{\text{corr}}(\mathfrak{p}||\mathfrak{q}) \leq 1$ if $\mathfrak{p}$ and $\mathfrak{q}$ are pdfs.

(d) $\mathcal{D}^{\text{corr}}(\mathfrak{p}||\mathfrak{q}) = 0$ if and only if $\mathfrak{p}$ amd $\mathfrak{q}$ are orthogonal.

(e) $\mathcal{D}^{\text{corr}}(\mathfrak{p}||\mathfrak{q}) = 1$ if and only if $\mathfrak{p}$ amd $\mathfrak{q}$ are identical.

(f) It exhibits symmetry, $\mathcal{D}^{\text{corr}}(\mathfrak{p}||\mathfrak{q}) = \mathcal{D}^{\text{corr}}(\mathfrak{q}||\mathfrak{p})$.

[1] Unless otherwise stated, limits of integration ranges from $-\infty$ to $+\infty$ throughout the paper.

(g) If samples (t, f) are independent draws from a pdf $\mathfrak{p}(t, f)$, the distance can be computed as $\mathcal{D}^{\mathrm{corr}}(\mathfrak{p}||\mathfrak{q}) = \mathbf{E}\left[q(t, f)\right]_{\mathfrak{p}(t,f)}$.

If $\{t_l, f_l\}_{l=1}^{L} \sim \mathfrak{p}(t, f)$ are independent draws from $\mathfrak{p}(t, f)$, the distance $\mathcal{D}^{\mathrm{corr}}(\mathfrak{p}||\mathfrak{q})$ in (5.27) can be approximated [101] by

$$\hat{\mathcal{D}}^{\mathrm{corr}}(\mathfrak{p}||\mathfrak{q}) = \frac{1}{L}\sum_{l=1}^{L} q(t_l, f_l). \tag{5.28}$$

The estimate is accurate for large L,

$$\lim_{L\to\infty}\left|\hat{\mathcal{D}}^{\mathrm{corr}}(\mathfrak{p}||\mathfrak{q}) - \mathcal{D}^{\mathrm{corr}}(\mathfrak{p}||\mathfrak{q})\right| \to 0.$$

The statistical similarity of a signal to a reference signal set is computed to represent a multi-dimensional signal with a 1-D feature using the following two-fold process called feature-reduction. Given a test signal $s^{\mathrm{te}}(t)$ and a reference signal $s_r^{\mathrm{ref}}(t) \in \mathbb{S}^{\mathrm{ref}}$ from a reference signal set $\mathbb{S}^{\mathrm{ref}}$ of R signals, respective MPD-PDFs, $P_{s^{\mathrm{te}}}(t, f)$ and $P_{s_r^{\mathrm{ref}}}(t, f)$ for $r = 1, \ldots, R$ are first computed using Equation (5.25), then the average statistical similarity $\mathcal{D}^{\mathrm{avg}}$ of $P_{s^{\mathrm{te}}(t)}$ to the MPD-PDFs of the reference set is computed as

$$\mathcal{D}^{\mathrm{avg}} = \mathcal{D}^{\mathrm{corr}}(P_{s_{\mathrm{te}}}||P_{\mathbb{S}^{\mathrm{ref}}}) = \frac{1}{R}\sum_{r=1}^{R}\mathcal{D}^{\mathrm{corr}}(P_{s^{\mathrm{test}}}||P_{s_r^{\mathrm{ref}}}), \tag{5.29}$$

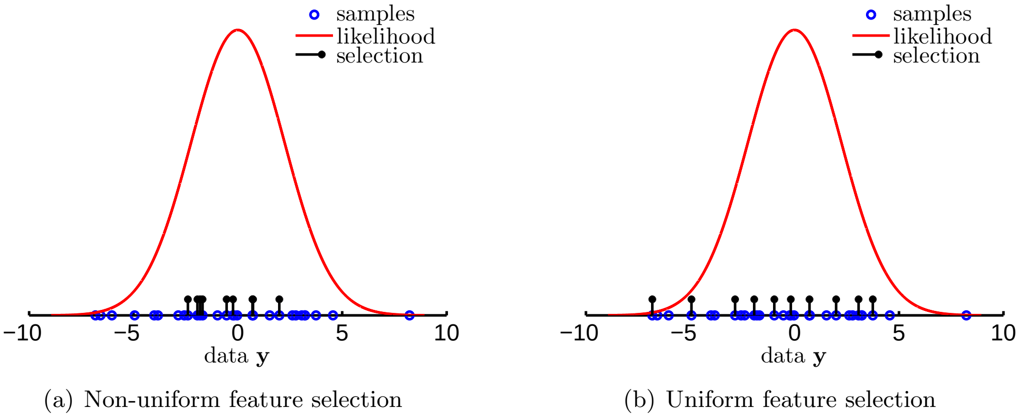
Due to the existence of linearity, it can be shown that $P_{\mathbb{S}^{\mathrm{ref}}}(t, f) = \dfrac{1}{R}\displaystyle\sum_{r=1}^{R} P_{s_r^{\mathrm{ref}}}(t, f)$ and can be denoted as the average MPD-PDF of the reference set. The pdf $P_{\mathbb{S}^{\mathrm{ref}}}(t, f)$ collects the statistical variability in the reference set $\mathbb{P}^{\mathrm{ref}}$. In the case of MPD with Gaussian dictionary, $P_{\mathbb{S}^{\mathrm{ref}}}(t, f)$ is a GMM.

In the application outlined by Figure 5.4, we had several test signals that were being considered simultaneously. The set of MPD-PDFs from the test signals was $\mathbb{P}^{\mathrm{te}}$ and the set of their representative 1-D distance features was $\mathbb{Y}^{\mathrm{te}}$. The existing set of features in the

FIFO buffer $\mathbb{Y}^{\text{FIFO}}$ was updated by replacing a subset of oldest features by the most recent $\mathbb{Y}^{\text{te}}$. The set $\mathbb{Y}^{\text{te}}$ would contain most informative features to represent the test signals if the reference set $\mathbb{P}^{\text{ref}}$ contained a statistically diverse collection of MPD-PDFs. Also, the reference set $\mathbb{P}^{\text{ref}}$ contained the MPD-PDFs that represented a set of features $\mathbb{Y}^{\text{ref}} \subset \mathbb{Y}^{\text{FIFO}}$. This implies that a judicious choice of $\mathbb{Y}^{\text{ref}}$ would benefit the the overall performance by choosing a statistically diverse $\mathbb{P}^{\text{ref}}$.

5.4.3. Minimum Discrepancy Uniform Reference Feature Selection

The goal of uniform feature selection is to choose a subset of *uniformly distributed* points from a given sample set. Figure 5.5 shows a simulated example of a feature set comprising samples, marked as circles, from a 1-D Gaussian distribution, shown by the continuous line, and a selected sample subset, marked as stem-dots. The selection in Figure 5.5(a) are clustered near the mean of the Gaussian distribution whereas those in Figure 5.5(b) are distributed more uniformly over the feature space and are therefore more representative of the entire feature space.



(a) Non-uniform feature selection

(b) Uniform feature selection

Fig. 5.5. A comparison of non-uniform vs. uniform feature selection.

The notion of uniformity can be quantified by the *discrepancy* [128] $\mathscr{D}_T(\mathbb{Y})$ defined for a set of points $\mathbb{Y} = \{y_1, \ldots, y_T\}$ within an interval $[y^{\min}, y^{\max}]$ as

$$\mathscr{D}_T(\mathbb{Y}) \triangleq \sup_{y^{\min} \leq y^l \leq y^h \leq y^{\max}} \left| \frac{\left|\mathbb{Y} \cap [y^l, y^h]\right|}{T} - \frac{y^h - y^l}{y^{\max} - y^{\min}} \right|, \tag{5.30}$$

where $[y^l, y^h]$ defines any subinterval of $[y^{\min}, y^{\max}]$, and $\left|\mathbb{Y} \cap [y^l, y^h]\right|$ represents the cardinality[2] of $\mathbb{Y}$ in $[y^l, y^h]$. The set $\mathbb{Y}$ is said to be uniformly distributed on the interval $[y^{\min}, y^{\max}]$ if $\lim\limits_{T\to\infty} \mathscr{D}_T(\mathbb{Y}) = 0$ [128,129]. For the given set $\mathbb{Y}$ in $[y^{\min}, y^{\max}]$, the discrepancy $\mathscr{D}_T(\mathbb{Y})$ can be used as a measure of uniformity. In particular, the discrepancy is observed to be minimum for a uniform $\mathbb{Y}$ in $[y^{\min}, y^{\max}]$. In the example shown in Figure 5.5, the discrepancy of the selected samples is greater in Figure 5.5(a) than it is in Figure 5.5(b).

Note that the set of 1-D features $\mathbb{Y}^{\text{FIFO}} = \{y_1, \ldots, y_T\}$ in the FIFO buffer in Figure 5.4 is assumed to be distributed according to a GMM. The task that remains here is to select a uniform subset $\mathbb{Y}^{\text{ref}}$. We define an indicator vector $\mathbb{Z} = \{z_1, \ldots, z_T\}$ to indicate the samples selected from the measurement $\mathbb{Y}^{\text{ref}}$:

$$z_i = \begin{cases} 1, & \text{if } y_i \text{ is selected} \\ 0, & \text{otherwise} \end{cases}, \; i = 1, \ldots, T.$$

The uniform feature selection problem can now be stated formally as: given a feature set comprising T points $\mathbb{Y}^{\text{FIFO}} = \{y_1, \ldots, y_T\}$ in the interval $[y^{\min}, y^{\max}]$, find a selection $\mathbb{Y}^{\text{ref}}$ of cardinality $\hat{T}$ which minimizes the discrepancy $\mathscr{D}_{\hat{T}}(\mathbb{Y}^{\text{FIFO}}_{\mathbb{Z}})$. Here, $\mathbb{Y}^{\text{FIFO}}_{\mathbb{Z}}$ represents the set of features in $y_i \in \mathbb{Y}^{\text{FIFO}}$ for which the indicator variable $z_i = 1$. The optimum selection is obtained by solving the problem

$$\begin{aligned} \text{minimize} \quad & \mathscr{D}_{\hat{T}}(\mathbb{Y}^{\text{FIFO}}_{\mathbb{Z}}) \\ \text{subject to} \quad & z_i \in \{0, 1\}, \;\; i = 1, \ldots, T \\ & |\mathbb{Z} \cap \{1\}| = \hat{T}. \end{aligned} \tag{5.31}$$

[2] Cardinality of a set is the number of elements in the set.

The minimization problem in Equation (5.31) is non-convex and in general very difficult to solve exactly [130]. Direct enumeration is ruled out because the number of possible selections is $\binom{T}{\hat{T}}$ which is very large even for relatively small T and $\hat{T}$. Evaluating the discrepancy can also be computationally expensive, and some of the estimation techniques to compute it include techniques based on sampling, searching, lower and upper bounds, and convex programming [129, 131–133].

In this dissertation, we propose an approximate and efficient method for uniform feature selection by considering the relaxed minimization problem

$$\begin{aligned} &\text{minimize} \quad && \tilde{\mathscr{D}}_{\hat{T}}(\mathbb{Y}_{\mathbb{Z}}^{\text{FIFO}}) \\ &\text{subject to} \quad && z_i \in \{0,1\}, \;\; i = 1, \ldots, T \\ & && |\mathbb{Z} \cap \{1\}| \approx \hat{T}, \end{aligned} \tag{5.32}$$

where the objective function is defined as

$$\tilde{\mathscr{D}}_{\hat{T}}(\mathbb{Y}_{\mathbb{Z}}^{\text{FIFO}}) \triangleq \frac{1}{H} \sum_{h=1}^{H} \left| \frac{\left|\mathbb{Y}_{\mathbb{Z}}^{\text{FIFO}} \cap [y^{\min} + (h-1)\Delta y, y^{\min} + h\Delta y]\right|}{\hat{T}} - \frac{\Delta y}{y^{\max} - y^{\min}} \right|, \tag{5.33}$$

with $\Delta y = (y^{\max} - y^{\min})/H$. Here, H denotes the number of equal length partitions in $[y^{\max} - y^{\min}]$. While the discrepancy in Equation (5.30) is defined as a supremum over all possible subintervals of $[y^{\min}, y^{\max}]$, the objective function of Equation (5.32) is defined as an average over the H equal-length subintervals partitioning $[y^{\min}, y^{\max}]$. The constrain on the number of data points to be selected, for the minimization problem in Equation (5.32), has also been relaxed. Algorithm 4 provides an approach to the solution of Equation (5.32) and it leads to a reasonably uniform data selection.

Algorithm 4 Uniform data selection via solution of problem (5.32)

```
Initialize $z_i = 0$ for all $i = 1, \ldots, T$.
Compute the ideal number of elements of $\mathbb{Y}^{\mathrm{FIFO}}_{\mathbb{Z}}$ that should belong to each $\Delta y$ interval and store it as the nearest integer $\mathtt{n} = \textbf{round}\left(\hat{T}/H\right)$.
for $h = 1$ to $H$ do
    The set $\mathbb{I}$ contains indices $i$ for which $\{y_i \in [y^{\min} + (j-1)\Delta y, y^{\min} + j\Delta y]\}$.
    Store the cardinality of $\mathbb{I}$ in $\mathtt{m}$.
    if $\mathtt{m} \leq \mathtt{n}$ then
        Select all the features in this interval, $z_i = 1\ \forall i \in \mathbb{I}$.
    else
        Randomly select $\mathtt{n}$ features $\hat{\mathbb{I}} \subset \mathbb{I}$ (i.e., $\left|\hat{\mathbb{I}}\right| = \mathtt{n}$) and $z_i = 1\ \forall i \in \hat{\mathbb{I}}$. Note that, the user may consider choosing $\hat{\mathbb{I}}$ by minimizing $\mathscr{D}_{\mathtt{n}}(\hat{\mathbb{I}})$, but it might be computationally expensive without any significant benefit.
    end if
end for
```

5.4.4. Progressive Clustering Using Adaptive Learning

The 1-D features in the FIFO buffer were modeled with a GMM,

$$\mathcal{G}(\mathbb{Y}^{\mathrm{FIFO}}|\boldsymbol{\theta}) = \sum_{m=1}^{\tilde{M}} p_m^* \mathcal{N}\left(\mu_m^*, \sigma_m^{2*}\right), \tag{5.34}$$

where $\mathbb{Y}^{\mathrm{FIFO}}$ were the features in the FIFO buffer and $\boldsymbol{\theta}$ and $\mathbf{p}* = \{p_1, \ldots, p_{\tilde{M}}\}$ were the parameters to learn. The learning was carried out in a Dirichlet process framework using a blocked Gibbs sampler, as outlined in Algorithm 3.

Initially, all the features in the buffer were assumed to be in the same cluster and $p_1 \approx 1$. As more data were collected, features could evolve to represent the concurrent damage states, which in this case, is the length of a crack in a compact tension sample. Every new collection of a set of signals is considered as one epoch in this experiment. The growing crack due to cyclic loading increased and altered the time-frequency signature of the signals collected by the sensors. These altered signals generated MPD-PDFs that were different from the previously seen set of MPD-PDFs. Thus, we obtained a set of $\mathbb{Y}^{\mathrm{te}}$ which was different from the ones present in $\mathbb{Y}^{\mathrm{FIFO}}$. Note that $\mathbb{Y}^{\mathrm{te}}$ is a much smaller set than

$\mathbb{Y}^{\text{FIFO}}$ and in this application, we implemented $\left|\mathbb{Y}^{\text{FIFO}}\right| = 3\left|\mathbb{Y}^{\text{te}}\right|$. The oldest features in the buffer were refreshed by the most recent $\mathbb{Y}^{\text{te}}$ and adaptive learning was performed on this new $\mathbb{Y}^{\text{FIFO}}$. At this stage, $\mathbf{p}^*$ converged to multiple non-zero components. The new features were learned and a new reference set $\mathbb{P}^{\text{ref}}$ was build. Progressively, this method generated $\mathbf{p}^{*(k)}$ at every kth epoch.

In certain epochs, the features did not exhibit significant change and no new clusters were learned from what was learned before. This indicated that there was no likely increase in crack length (or change in damage state). However, under certain situations changes in features were observed due to changes in environmental condition, which in this case was the cyclic loading envelope. This lead to false clustering. Bayesian filtering provided the optimal framework to incorporate information from physics based state model with the observed clustering to provide a more accurate estimate of the crack length.

The multinomial distribution $\mathbf{p}^{*(k)}$ was used as the observation distribution for the Bayesian filter at every epoch k. Next, this is discussed in details.

5.4.5. Bayesian Filtering

In a state-space framework, Bayesian filter [134] provides a general unifying framework for sequential state estimation. Given the state and the observation probability distribution, a Bayesian filter estimates the posterior distribution over the state given concurrent observations.

In our discussion of Bayesian filtering, we use $x^{(k)}$ and $\mathbf{y}^{(k)}$ as the state and observations at the kth epoch respectively. We also introduce an environment parameter $\Phi^{(k)}$ to account for any alterations due to environmental or operating conditions. In SHM applications, the observation and the state evolution is not independent of the concurrent variations in environmental/operational condition. The state distribution at the kth epoch can be modeled as $P\left(x^{(k)}|x^{(k-1)},\Phi^{(k)}\right)$ and the concurrent observations distribution can be modeled as $P\left(\mathbf{y}^{(k)}|x^{(k)},\Phi^{(k)}\right)$. The following assumptions hold for SHM applications:

(a) $\Phi^{(k)}$ is fairly accurately measurable at any epoch k

(b) the states and the damage do not depend on previous or future environmental/operating conditions.

Using Bayes' theorem, the observation model can be related to the state posterior as

$$P\left(x^{(k)}|\mathbf{y}^{(k)},\mathbf{y}^{(1:k-1)},\Phi^{(1:k)}\right) \propto P\left(\mathbf{y}^{(k)}|x^{(k)},\mathbf{y}^{(1:k-1)},\Phi^{(1:k)}\right) P\left(x^{(k)}|\mathbf{y}^{(1:k-1)},\Phi^{(1:k)}\right), \tag{5.35a}$$

where $\mathbf{y}^{(1:k-1)}$ are the observations up to epoch $k-1$ and $\Phi^{(1:k)}$ are the environmental conditions up to epoch k. Under the assumptions,

(a) the current observation $\mathbf{y}^{(k)}$ is independent of the past observations $\mathbf{y}^{(1:k-1)}$

(b) the current observation $\mathbf{y}^{(k)}$ depends only on the current environmental conditions $\Phi^{(k)}$

(c) the current state is not affected by the environmental conditions from the past or the future,

the above equation can be simplified to give the state posterior inference

$$P\left(x^{(k)}|\mathbf{y}^{(1:k)},\Phi^{(1:k)}\right) \propto P\left(\mathbf{y}^{(k)}|x^{(k)},\Phi^{(k)}\right) P\left(x^{(k)}|\mathbf{y}^{(1:k-1)},\Phi^{(k)}\right). \tag{5.35b}$$

The state prior can be obtained by

$$P\left(x^{(k)}|\mathbf{y}^{(1:k-1)},\Phi^{(k)}\right) = \int P\left(x^{(k)}|x^{(k-1)},\Phi^{(k)}\right) P\left(x^{(k-1)}|\mathbf{y}^{(1:k-1)},\Phi^{(k)}\right) dx^{(k-1)}. \tag{5.35c}$$

In fatigue crack estimation for SHM, the state $x^{(k)}$ is the length of crack and the observations $\mathbf{y}^{(k)}$ is mixture weights $\mathbf{p}^*$ at the kth epoch. The state distribution can be assumed discrete with N_x possibilities. Equations (5.35b) and (5.35c) can be reformulated as

$$\begin{aligned}\Pr\left(x^{(k)}|\mathbf{y}^{(1:k)}, \Phi^{(k)}\right) &\propto \Pr\left(\mathbf{y}^{(k)}|x^{(k)}, \Phi^{(k)}\right) \Pr\left(x^{(k)}|\mathbf{y}^{(1:k-1)}, \Phi^{(k)}\right) && (5.36\text{a})\\ \Pr\left(x^{(k)}|\mathbf{y}^{(1:k-1)}, \Phi^{(k)}\right) &= \sum_{l=1}^{N_x} \Pr\left(x^{(k)}|{x_l}^{(k-1)}, \Phi^{(k)}\right) \Pr\left({x_l}^{(k-1)}|\mathbf{y}^{(1:k-1)}, \Phi^{(k-1)}\right), && (5.36\text{b})\end{aligned}$$

respectively.

For compact tension (CT) sample under cyclic fatigue loading, the physics based stochastic crack propagation model is presented in [135–138]. It is assumed that $P\left(x^{(k)}|x^{(k-1)}, \Phi^{(k)}\right)$ is a log-normal distribution under a known loading condition $\Phi^{(k)}$. The choice of log-normal distribution was motivated by the properties (a) it can be represented by two parameters, mean and variance, and (b) the shape envelopes $P\left(x^{(k)}|x^{(k-1)}, \Phi^{(k)}\right)$. The sequential state evolution is formulated as

$$x^{(k)} = x^{(k-1)} + \chi\mathcal{H}(x^{(k-1)}, \Phi^{(k)}),$$

where $\mathcal{H}(x^{(k-1)}, \Phi^{(k)})$ is a function of $x^{(k-1)}$ and $\Phi^{(k)}$, and χ is a draw from a discrete log-normal distribution as described in Appendix C.2. A discrete assumption of the state variable is computationally convenient and justified because in practical applications, there may be minimum change in crack length below which there is no perceptible change is the observations. Thus, the conditional state distribution can be assumed discrete and is given as

$$\Pr\left(x^{(k)}|x^{(k-1)}, \Phi^{(k)}\right) = \overline{\mathit{Log}\text{-}\mathcal{N}}\left(\mu_{x^{(k)}}, \sigma_{x^{(k)}}, \Delta_\chi\right)$$

such that

$$\begin{aligned}\mathbf{E}\left[x^{(k)}|x^{(k-1)},\Phi^{(k)}\right] &= x^{(k-1)}+\mathbf{E}\left[\chi\right]\mathcal{H}(x^{(k-1)},\Phi^{(k)})\\ \mathbf{Var}\left[x^{(k)}|x^{(k-1)},\Phi^{(k)}\right] &= \mathbf{Var}\left[\chi\right]\mathcal{H}(x^{(k-1)},\Phi^{(k)})^2,\end{aligned}$$

where $\mathbf{E}\left[\chi\right]$ is the mean of χ, $\mathbf{Var}\left[\chi\right]$ is the variance of χ and Δ_χ is a suitable choice of the discretization level.

We propose a modified approach to the discussed Bayesian approach. Here, we use the change in damage as the state Δx for the Bayesian filter. Thus, we can formulate the state distribution as

$$\Pr\left(\Delta x^{(k)}|x^{(k-1)},\Phi^{(k)}\right)\sim\overline{\mathit{Log}\text{-}\mathcal{N}}\left(\mu_{\Delta x^{(k)}},\sigma_{\Delta x^{(k)}},\Delta_\chi\right) \tag{5.37a}$$

such that

$$\mathbf{E}\left[\Delta x^{(k)}|x^{(k-1)},\Phi^{(k)}\right] = \mathbf{E}\left[\chi\right]\mathcal{H}(x^{(k-1)},\Phi^{(k)})\text{ and} \tag{5.37b}$$

$$\mathbf{Var}\left[\Delta x^{(k)}|x^{(k-1)},\Phi^{(k)}\right] = \mathbf{Var}\left[\chi\right]\mathcal{H}(x^{(k-1)},\Phi^{(k)})^2, \tag{5.37c}$$

where, the change in crack is

$$\Delta x^{(k)}\triangleq x^{(k)}-x^{(k-1)}=\chi\mathcal{H}(x^{(k-1)},\Phi^{(k)}). \tag{5.38}$$

A mean estimate of the state is used in this application and at any epoch k it is

$$\bar{x}^{(k)}=\mathbf{E}\left[x^{(k)}\right], \tag{5.39}$$

and all the state estimates up to kth epoch can be collectively addressed as $\bar{x}^{(1:k)}$.

Further, we model the change in the observations $\Delta\mathbf{y}^{(k)}$ as the change in the clustering ($\mathbf{p}^{*(k-1)}\rightarrow\mathbf{p}^{*(k)}$) obtained from $k-1$ to kth epoch. This change in clustering can be associated to a given change in the damage state $\Delta x^{(k)}$. Note that in this formulation, there no observation model that maps the states to the observations. Thus, an empirical observation was used in this application, which is estimated at every epoch. Specifically, the distribution of $\Delta \mathbf{y}^{(k)}$ can be approximated by a negative-binomial

$$\Pr\left(\Delta \mathbf{y}^{(k)} | \Delta x^{(k)}\right) = \textit{Neg-Bin}\left(\mathbf{E}\left[\Delta \mathbf{y}^{(k)}\right], \mathbf{Var}\left[\Delta \mathbf{y}^{(k)}\right]\right)$$

as discussed in Appendix D, with mean and variance

$$\begin{aligned} \mathbf{E}\left[\Delta \mathbf{y}^{(k)}\right] &= \mathbf{E}\left[\mathbf{y}^{(k)}\right] - \mathbf{E}\left[\mathbf{y}^{(k-1)}\right] \\ \mathbf{Var}\left[\Delta \mathbf{y}^{(k)}\right] &= \mathbf{Var}\left[\mathbf{y}^{(k)}\right] + \mathbf{Var}\left[\mathbf{y}^{(k-1)}\right], \end{aligned}$$

respectively. The choice of negative-binomial distribution was an appropriate choice for the empirical observation distribution because the negative-binomial distribution has that following characteristics:

(a) it is bi-parametric and can be uniquely characterized by mean and variance (or mode and variance)

(b) it is discrete

(c) it can approximated to a finite support.

The modified Bayesian filter approach for progressive fatigue crack estimation has been summarized in Algorithm 5.

The proposed algorithm faces a few challenges. Preprocessing stage of the feature extraction involved MPD, which is an iterative process. Samples need to be constantly drawn from a two dimensional GMM to estimate the statistical measure of similarity, every

Algorithm 5 Bayesian filtering for progressive fatigue crack damage estimation.

```
Estimate the mean and variance of the change in observations
    $\mathbf{E}\left[\Delta\mathbf{y}^{(k)}\right] = \mathbf{E}\left[\mathbf{y}^{(k)}\right] - \mathbf{E}\left[\mathbf{y}^{(k-1)}\right] \quad (5.40a)$
    $\mathbf{Var}\left[\Delta\mathbf{y}^{(k)}\right] = \mathbf{Var}\left[\mathbf{y}^{(k)}\right] + \mathbf{Var}\left[\mathbf{y}^{(k-1)}\right], \quad (5.40b)$
  and realize the change in observation distribution as
    $\Pr\left(\Delta\mathbf{y}^{(k)}|\Delta x^{(k)}, \Phi^{(k)}\right) = \textit{Neg-Bin}\left(\mathbf{E}\left[\Delta\mathbf{y}^{(k)}\right], \mathbf{Var}\left[\Delta\mathbf{y}^{(k)}\right]\right). \quad (5.40c)$
Estimate the Bayesian prior
    $\Pr\left(\Delta x^{(k)}|\Delta\mathbf{y}^{(1:k-1)}, \bar{x}^{(1:k-2)}, \Phi^{(k)}\right) = \sum_{l=1}^{N_x} \Pr\left(\Delta x^{(k)}|{x_l}^{(k-1)}, \Phi^{(k)}\right) \Pr\left({x_l}^{(k-1)}|\Delta\mathbf{y}^{(1:k-1)}, \bar{x}^{(1:k-2)}, \Phi^{(k-1)}\right). \quad (5.40d)$
Compute the Bayesian posterior
    $\Pr\left(\Delta x^{(k)}|\Delta\mathbf{y}^{(1:k)}, \bar{x}^{(1:k-2)}, \Phi^{(k)}\right) \propto \Pr\left(\Delta\mathbf{y}^{(k)}|\Delta x^{(k)}, \Phi^{(k)}\right) \Pr\left(\Delta x^{(k)}|\Delta\mathbf{y}^{(1:k-1)}, \bar{x}^{(1:k-2)}, \Phi^{(k)}\right). \quad (5.40e)$
Given $\Pr\left(x^{(k-1)}|\Delta\mathbf{y}^{(1:k-1)}, \bar{x}^{(1:k-2)}, \Phi^{(k-1)}\right)$ estimate the state at of previous epoch
    $\bar{x}^{(k-1)} = \mathbf{E}\left[x^{(k-1)}\right]. \quad (5.40f)$
Compute the current state distribution
    $\Pr\left(x^{(k)}|\Delta\mathbf{y}^{(1:k)}, \bar{x}^{(1:k-1)}, \Phi^{(k)}\right) = \Pr\left(\Delta x^{(k)} + \bar{x}^{(k-1)}|\Delta\mathbf{y}^{(1:k)}, \bar{x}^{(1:k-1)}, \Phi^{(k)}\right). \quad (5.40g)$
```

time the representative subset is updated, which limits the use of data models to the ones from which samples can be drawn. The Bayesian filter relies on an accurate stochastic physics based model that incorporates models for variable working conditions. Such models are not truly stochastic and do not incorporate the stochastic nature of material property and the effect of operating conditions. In addition, these models are non-trivial to develop in complex structures.

5.5. Adaptive Estimation of Fatigue Crack Damage in Compact Tension Sample

5.5.1. Experimental Setup and Data Collection

In this application, we use a aluminum 2024 T3 6.31 mm thick compact tension (CT) sample as shown in Figure 5.6. The CT specimens were fabricated according to ASTM E647-93 with a width of 25.4 mm, from the center of the pin hole to the edge of the specimen. An initial notch of length 5 mm was made to pre-determine the location of crack initiation. The experiments were performed in an Instron 1331 servohydraulic load frame operating at 20 Hz. To simulate real flight conditions a typical center wing load spectrum was programmed into the digital controller of the load frame, with an envelope as shown in Figure 5.7. The variable amplitude cyclic load applied was up to about 45 kilocycles to induce fatigue crack damage. Two surface mounted piezoelectric (PZT) sensors (placed symmetrically on the sample as shown in Figure 5.6) were used to make measurements at several stages of fatigue loading cycles. One PZT was used as the actuator and the other was used as a sensor to measure the response signals. A 130 kHz burst signal was used for excitation. Further details of the experimental setup and data collection can be found

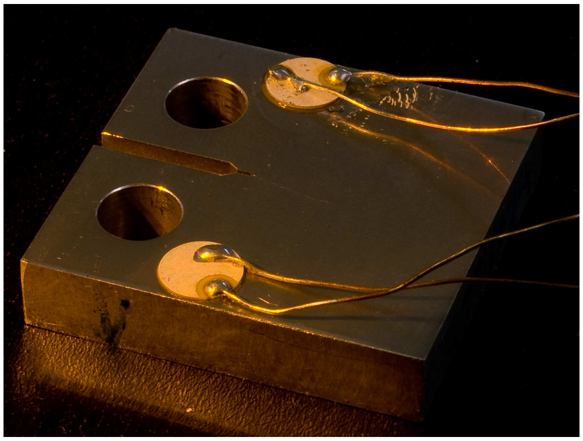


Fig. 5.6. The compact tension (CT) sample.

in [135, 139].

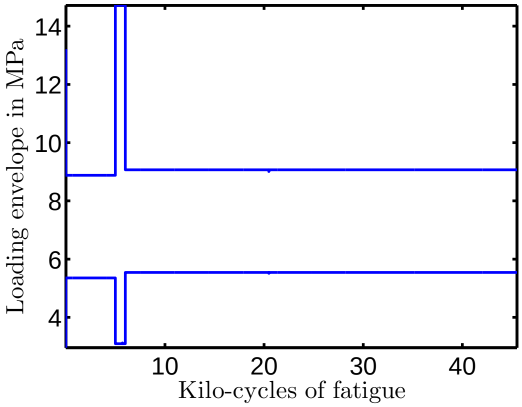
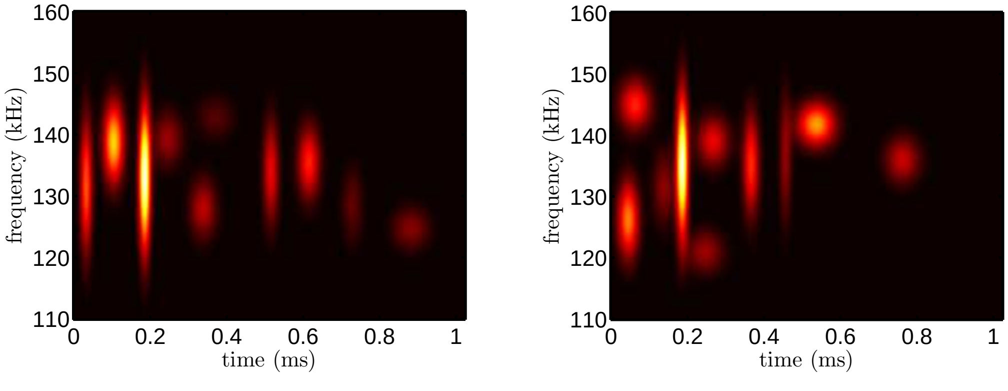


Fig. 5.7. Envelope of loading cycle for CT sample.

Signals were collected in sets of 210 at 4452, 5123, 5720, 20480, 34451 and 45507 cycles of loading. Preprocessing of these signals included mean removal, low-pass filtering (to remove high frequency noise), downsampling to 333 kHz and normalized to unit energy. Then, MPD was performed on each signal with $N = 10$ iterations, with a dictionary of 8 million atoms and the MPD-TFRs were constructed. Figure 5.8 shows the MPD-TFR of signals from two different damage states, 6.17mm and 6.28mm cracks measured at 4452 and 5123 cycles of damage respectively. The crack length measurements were done using a

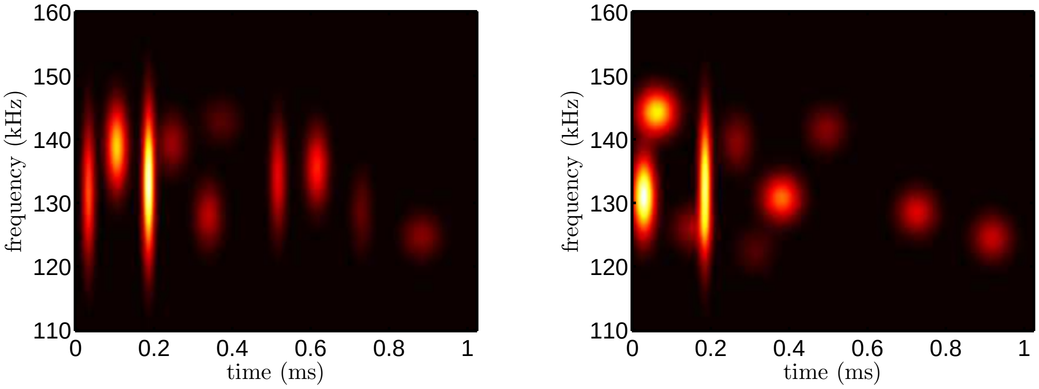


(a) MPD-TFR from 6.17mm crack.

(b) MPD-TFR from 6.28mm crack.

Fig. 5.8. Example MPD-TFR of signals from two of the observed damage states.

scanning electron microscopy (SEM) in an FEI XL-30 operating at 15 kV [135]. Figure 5.9 shows the MPD-TFRs of two signals measured obtained by successive measurements for a crack length of 6.17 mm. From the plots, we see marked difference in the time-frequency structure of signals from the same damage state. This example demonstrates the inherent variability present in measurements from a given damage state.



(a)

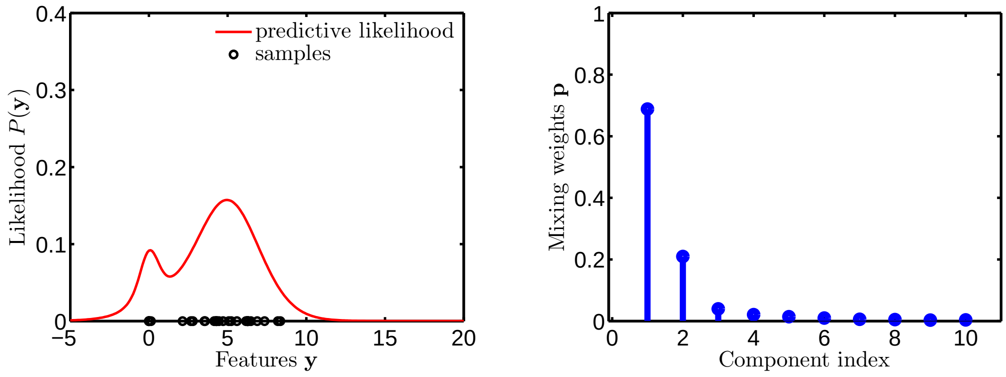
(b)

Fig. 5.9. MPD-TFRs of two signals for a crack length of 6.17 mm.

Active data selection provided a mechanism to address this variability. In this work, we incorporate the proposed uniform data selection algorithm into the adaptive learning framework as described in Section 5.4. Specifically, at any given epoch, we first computed the statistical similarity features between the current measured signals and the reference set. Then, adaptive clustering was performed on these features and used subsequently for damage estimation. Finally, the active data selection was used to update the reference set.

5.5.2. Results

Figure 5.10 shows an intermediate clustering learned by the adaptive learning process. Here, the DP base distribution (G_0) was normal-gamma with hyperparameters $u_\mu = 0$, $u_\tau = 0$, $u_a = 1$ and $u_b = 1$. The provided features were from two groups. A burn-in of 100



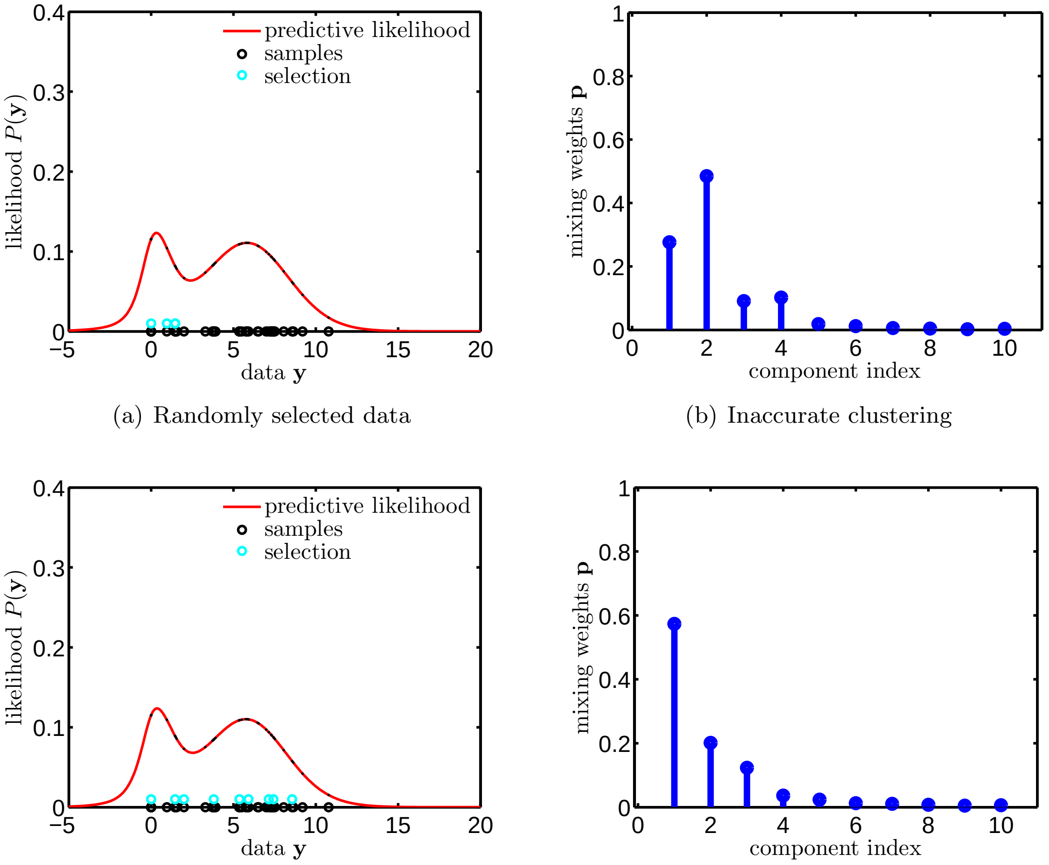
(a) Predictive likelihood.

(b) Mixing weights.

Fig. 5.10. Adaptive learning on features showing two clusters.

and a burn-out of 100 was used. A longer burn-in or a longer burn-out would only slow the process without any practical benefit in this application. In Figure 5.10(a) we see that the GMM learned from the data by the adaptive learning methodology is fairly accurate. The mixing weights are shown in Figure 5.10(b). Leaning at every epoch required about 0.3 seconds on a 2.8 GHz processor.

For the reference MPD-PDF $\mathbb{P}^{\text{ref}}$, $\hat{T} = 10$ out of $T = 30$ features were selected from the FIFO. Figure 5.11 compares the performance of the adaptive clustering of the 1-D features in the FIFO at one epoch with and without active data selection. Note that we obtained similar performance at different time epochs. Figure 5.11(a) shows the features, the adaptively learned likelihood, and a random selection for the reference set $\mathbb{P}^{\text{ref}}$. The corresponding adaptive clustering results are given in Figure 5.11(b). Figure 5.11(c) shows the features, the adaptively learned likelihood, and a uniformly selection of features for the reference set $\mathbb{P}^{\text{ref}}$. The corresponding adaptive clustering results are given in Figure 5.11(d). In this example, we observed that if the features belonged to the same damage state, a single cluster could be correctly identified by using the adaptive learning technique. To ensure this reliable performance, the plots in Figure 5.11 demonstrates the usefulness of using uniform feature selection. In the case of randomly selected features, the number of clusters were incorrectly identified as two.



(a) Randomly selected data

(b) Inaccurate clustering

(c) Uniformly selected data

(d) Accurate clustering

Fig. 5.11. Active data selection for improved adaptive learning.

A DP innovation of $\iota = 1.5$ was used with a truncation limit of $M = 10$ groups. The error as shown in (5.19) could be computed as

$$\mathcal{E} = 4Me^{-\frac{M-1}{\iota}} = 0.099$$

The $\mathcal{E}$ depends on the the values of M and ι as shown in Figure 5.12. Notice that a larger value of M would reduce the $\mathcal{E}$ but at the cost of increased model complexity, thereby

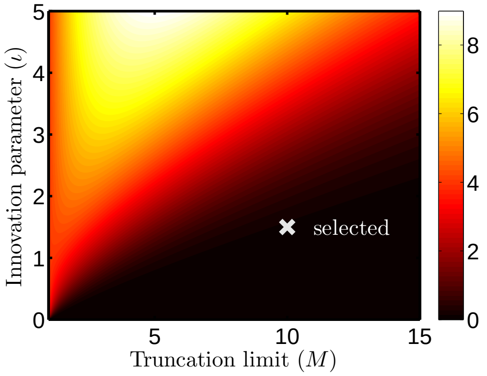
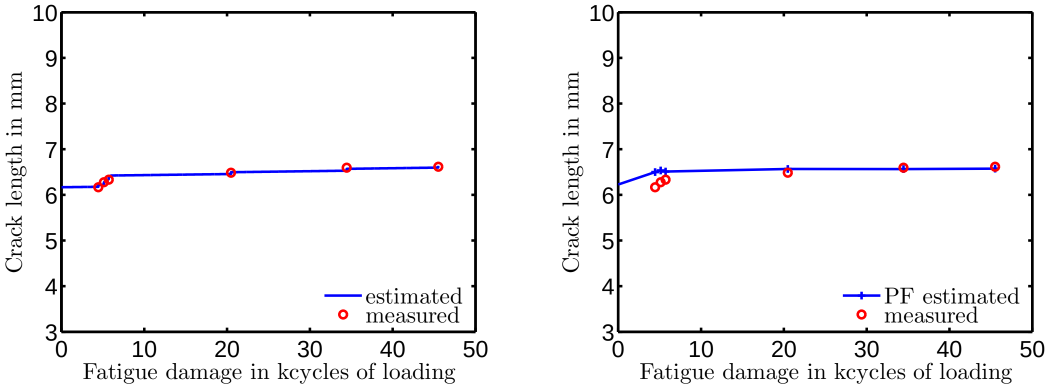


Fig. 5.12. 1-norm of the difference ($\mathcal{E}$) in true and estimated probability density function as a function of the Dirichlet process truncation limit M and innovation parameter ι.

raising the computational cost. On the other hand, a lower ι would yield smaller $\mathcal{E}$, but it would also reduce the chance of generating new groups from the given data. By assigning an ι too small would involve the risk of clustering all the features into a single cluster and affect the sensitivity of the algorithm to evolving data.

Signals were available at 4452, 5123, 5720, 20480, 34451 and 45507 cycles of loading. For all other cycles, the damage was estimated by solely using the physics based model. When observations were available, Bayesian filter was used to accurately estimate the distribution over the damage. The load value was available at every cycle of loading and was used to estimate the crack length with the aid of the physics based state equation. Figure 5.13(a) shows the performance of the proposed algorithm. The average squared error in the estimation is in the order of 10^{-3}mm. Figure 5.13(b) is provided to compare the performance of the (training/testing based) particle filtering (PF) approach described in [27]. In the adaptive learning approach, no learning templates (unlike the particle filtering algorithm) was necessary, yet they both perform equivalently.



(a) Using adaptive learning approach.

(b) Using particle filtering approach (courtesy [27])

Fig. 5.13. Crack estimation at every damage cycle.

CHAPTER 6

Transfer Learning based Reduced Training Damage Classification

6.1. Background

The conventional classification task entails classifying a set of test data $\mathbb{Y}^{\text{te}} = \{Y_1^{\text{te}}, \ldots, Y_{N_{\text{te}}}^{\text{te}}\}$ to M possible classes, parameterized by $\boldsymbol{\theta}_m$, $m = 1, \ldots, M$. The class parameters $\boldsymbol{\theta}_m$ are estimated using the set of training data $\mathbb{Y}^{\text{tr}} = \{Y_1^{\text{tr}}, \ldots, Y_{N_{\text{tr}}}^{\text{tr}}\}$, where the class memberships $\mathbf{c}^{\text{tr}} = \{c_1^{\text{tr}}, \ldots, c_{N_{\text{tr}}}^{\text{tr}}\}$ of the corresponding data are known. The class membership is a value that indicates to which class a data belongs, e.g., if $c_1^{\text{tr}} = 5$, Y_1^{tr} belongs to class $m = 5$. The parameters of each class can be estimated from the training data as

$$\boldsymbol{\theta}_m = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P\left(\mathbb{Y}^{\text{tr}} | \boldsymbol{\theta}, \mathbf{c}^{\text{tr}}\right).$$

This process is called training. Note that the above equation gives the maximum-likelihood (ML) estimate of the parameters and is a suggested parameter estimation technique for parametric models. In the Bayesian framework, any test data $Y_i^{\text{te}} \in \mathbb{Y}_{\text{te}}$ can be classified to one of the M classes as

$$c_i^{\text{te}} = \underset{m}{\operatorname{argmax}}\, P\left(Y_i^{\text{te}} | \boldsymbol{\theta}_m\right) \Pr\left(\boldsymbol{\theta}_m\right).$$

Here $P\left(Y_i^{\text{te}} | \boldsymbol{\theta}_m\right)$ is the likelihood of the data Y_i^{te} in the mth class and $\Pr\left(\boldsymbol{\theta}_m\right)$ is the prior (discrete) probability of the classes. In this dissertation, we extend this classification task to the transfer learning (TL) framework.

In the TL framework, we have a source domain data and task, and a target domain data and task. Formally, TL can be defined as follows. *Given a source domain and a source learning task, and a target domain and a target learning task, TL aims to help improve the learning of the target predictive function in the target domain using the knowledge in the source domain and the source learning task* [62].

Here, the source and target domain tasks are classification, as discussed earlier. In the source domain (abbreviated as sor), we have training (abbreviated as tr) and testing (abbreviated as te) sets of data $\mathbb{Y}^{\mathrm{tr}}_{\mathrm{sor}} = \{Y^{\mathrm{tr,sor}}_1, \ldots, Y^{\mathrm{tr,sor}}_{N_{\mathrm{tr,sor}}}\}$ (with known class membership $\mathbf{c}^{\mathrm{tr}}_{\mathrm{sor}} = \{c^{\mathrm{tr,sor}}_1, \ldots, c^{\mathrm{tr,sor}}_{N_{\mathrm{tr,sor}}}\}$) and $\mathbb{Y}^{\mathrm{te}}_{\mathrm{sor}} = \{Y^{\mathrm{te,sor}}_1, \ldots, Y^{\mathrm{te,sor}}_{N_{\mathrm{te,sor}}}\}$, respectively. Similarly, in the target domain (abbreviated as tar), we have $\mathbb{Y}^{\mathrm{tr}}_{\mathrm{tar}} = \{Y^{\mathrm{tr,tar}}_1, \ldots, Y^{\mathrm{tr,tar}}_{N_{\mathrm{tr,tar}}}\}$ (with known class membership $\mathbf{c}^{\mathrm{tr}}_{\mathrm{tar}} = \{c^{\mathrm{tr,tar}}_1, \ldots, c^{\mathrm{tr,tar}}_{N_{\mathrm{tr,tar}}}\}$) and $\mathbb{Y}^{\mathrm{te}}_{\mathrm{tar}} = \{Y^{\mathrm{te,tar}}_1, \ldots, Y^{\mathrm{te,tar}}_{N_{\mathrm{te,tar}}}\}$ in the target domain. However, $\mathbb{Y}^{\mathrm{tr}}_{\mathrm{tar}}$ is a statistically insignificant set and the parameters $\boldsymbol{\theta}^{\mathrm{tar}}_m$ estimated using $\mathbb{Y}^{\mathrm{tr}}_{\mathrm{tar}}$ and $\mathbf{c}^{\mathrm{tr}}_{\mathrm{tar}}$ for the classes cannot be reliably estimated. This increases the rate of mis-classification in the set of test data. The TL approach described in the following, addresses this issue and provides a framework by learning a target predictive function that relates the estimates of the parameters from the source domain $\boldsymbol{\theta}^{\mathrm{sor}}_m$ to the task in target domain and improves the classification of the target domain test data $\mathbb{Y}^{\mathrm{te}}_{\mathrm{tar}}$.

6.2. Formulation of Transfer Learning Methodology

The training data in the source domain is used to model the M classes, from which data is available. Under the assumption of parametric model for the classes $m = 1, \ldots, M$, $\boldsymbol{\theta}^{\mathrm{sor}}_m$ are the parameters for class m and can be estimated as

$$\boldsymbol{\theta}^{\mathrm{sor}}_m = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P\left(\mathbb{Y}^{\mathrm{tr}}_{\mathrm{sor}} | \boldsymbol{\theta}, \mathbf{c}^{\mathrm{tr}}_{\mathrm{sor}}\right). \tag{6.1}$$

Subsequently, classification of the test data is initially performed to learn the class memberships of the testing data in the source domain

$$c^{\mathrm{sor,te}}_i = \underset{m}{\operatorname{argmax}}\, P\left(Y^{\mathrm{sor,te}}_i | \boldsymbol{\theta}^{\mathrm{sor}}_m\right) \Pr\left(\boldsymbol{\theta}^{\mathrm{sor}}_m\right). \tag{6.2}$$

This classification serves as a validation and demonstrates how well the parameters have been estimated. Ideally, this should demonstrate minimum mis-classification.

Similarly, in the target domain, training and classification can be performed using the following equations

$$\begin{aligned}
\boldsymbol{\theta}_m^{\text{tar}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, P\left(\mathbb{Y}_{\text{tar}}^{\text{tr}}|\boldsymbol{\theta}, \mathbf{c}_{\text{tar}}^{\text{tr}}\right) && \text{(6.3a)}\\
c_i^{\text{tar,te}} &= \underset{m}{\operatorname{argmax}}\, P\left(Y_i^{\text{tar,te}}|\boldsymbol{\theta}_m^{\text{tar}}\right) \Pr\left(\boldsymbol{\theta}_m^{\text{tar}}\right), && \text{(6.3b)}
\end{aligned}$$

respectively. The training in Equation (6.3a) is likely to be inaccurate due to statistically insufficient training data. This results in large mis-classification when classified according to Equation (6.3b).

In this application, we can learn the target predictive function in two steps. First, we rewrite the objective function in Equation (6.3b) using Bayes' theorem

$$P\left(Y_i^{\text{tar,te}}|\boldsymbol{\theta}_m^{\text{tar}}\right) \Pr\left(\boldsymbol{\theta}_m^{\text{tar}}\right) = P\left(Y_i^{\text{tar,te}}, \boldsymbol{\theta}_m^{\text{tar}}\right).$$

Next, we assume that we can improve the classification in the target domain by jointly using the information learnt in the source domain and the data available in the target domain. Thus, the objective function can be modified to $P\left(\boldsymbol{\theta}_m^{\text{tar}}, Y_i^{\text{tar,te}}, \boldsymbol{\theta}_m^{\text{sor}}, Y_i^{\text{sor,te}}\right)$ and simplified as follows.

$$\begin{aligned}
& P\left(\boldsymbol{\theta}_m^{\text{tar}}, Y_i^{\text{tar,te}}, \boldsymbol{\theta}_m^{\text{sor}}, Y_i^{\text{sor,te}}\right)\\
= & P\left(Y_i^{\text{tar,te}}|\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}, Y_i^{\text{sor,te}}\right) P\left(\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}, Y_i^{\text{sor,te}}\right)\\
= & P\left(Y_i^{\text{tar,te}}|\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}, Y_i^{\text{sor,te}}\right) P\left(Y_i^{\text{sor,te}}|\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}\right) \Pr\left(\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}\right)\\
= & P\left(Y_i^{\text{tar,te}}|\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}, Y_i^{\text{sor,te}}\right) P\left(Y_i^{\text{sor,te}}|\boldsymbol{\theta}_m^{\text{sor}}\right) \Pr\left(\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}\right)\\
& [\text{since, } Y_i^{\text{sor,te}} \text{ does no depend on } \boldsymbol{\theta}_m^{\text{tar}}] \qquad \text{(6.4)}
\end{aligned}$$

Thus, the classification in the target domain can be reformulated as

$$c_i^{\text{tar,te}} = \underset{m}{\operatorname{argmax}}\, P\left(Y_i^{\text{tar,te}}|\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}, Y_i^{\text{sor,te}}\right) P\left(Y_i^{\text{sor,te}}|\boldsymbol{\theta}_m^{\text{sor}}\right) \Pr\left(\boldsymbol{\theta}_m^{\text{tar}}, \boldsymbol{\theta}_m^{\text{sor}}\right). \qquad \text{(6.5)}$$

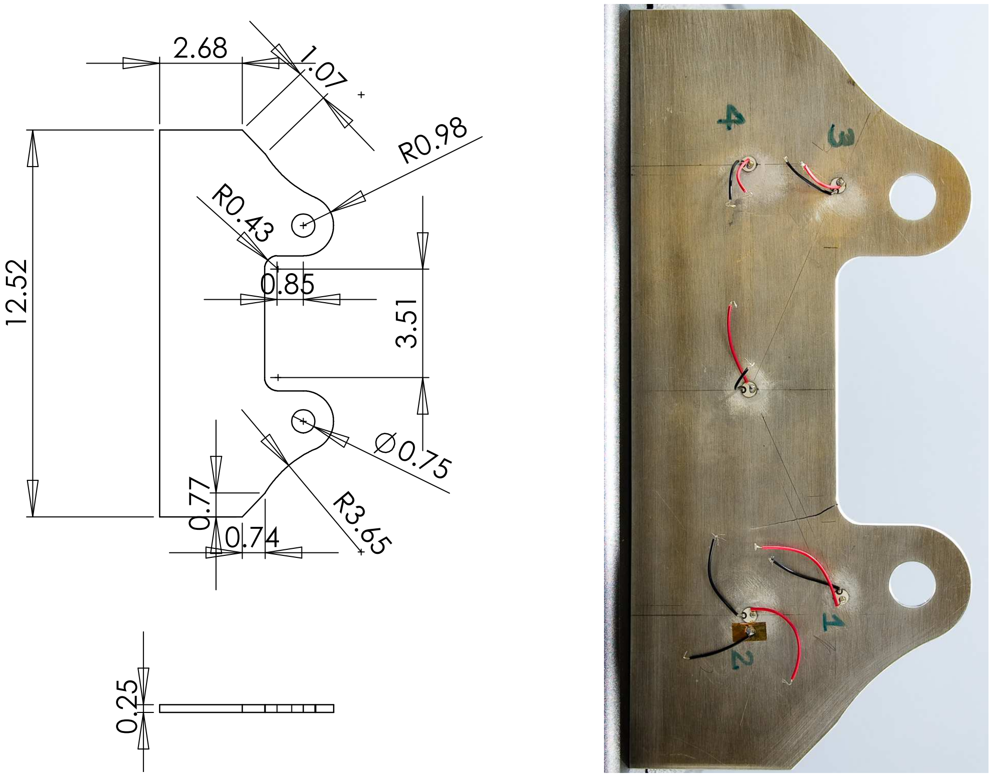
As an extension to this analysis, let us assume an experiment where source and target domain data are from two independent sensors. This is a typical problem of detecting a signal under independent and identically distributed (i.i.d.) random noise channel, where the noise in each sensor are independent. In this case, $Y_i^{\mathrm{sor,te}}$ is independent of $Y_i^{\mathrm{tar,te}}$. It also implies that $Y_i^{\mathrm{tar,te}}$ does not depend on the source class parameters $\boldsymbol{\theta}_m^{\mathrm{sor}}$, and $\boldsymbol{\theta}_m^{\mathrm{sor}}$ and $\boldsymbol{\theta}_m^{\mathrm{tar}}$ are also independent. Then, Equation (6.5) simplifies to a Bayesian sensor fusion problem given by

$$c_i^{\mathrm{tar,te}} = \underset{m}{\operatorname{argmax}}\, P\left(Y_i^{\mathrm{tar,te}} | \boldsymbol{\theta}_m^{\mathrm{tar}}\right) \Pr\left(\boldsymbol{\theta}_m^{\mathrm{tar}}\right)\; P\left(Y_i^{\mathrm{sor,te}} | \boldsymbol{\theta}_m^{\mathrm{sor}}\right) \Pr\left(\boldsymbol{\theta}_m^{\mathrm{sor}}\right). \tag{6.6}$$

6.3. Application of Transfer Learning Methodology

In this application, we are demonstrating the use of TL in fatigue crack damage classification. The sample under investigation is a aluminum lug sample with dimensions as shown in Figure 6.1(a). Figure 6.1(b) shows the location of piezoelectric sensors. There are four sensors, marked 1 through 4, placed symmetrically about a piezoelectric actuator. Observe the existence of a crack close to sensors 1 and 2. The actuation was done with a tone burst of 250 kHz, with a duration of 5 oscillations as shown in Figure 6.2(a). In total there were $M = 11$ damage classes, with 2402, 22493, 32253, 62617, 72428, 82597, 112558, 154037, 154431, 155106 and 155705 cycles of fatigue, corresponding to $m = 1, \ldots, 11$. Each damage class has 100 measurements from each of the four sensors.

The matching pursuit decomposition (MPD) [42] with a limited Gaussian atom dictionary was used as a feature extraction procedure [21] from the signals, obtained from the four sensors. The scaling parameter described in [21] was s function of the frequency so that strictly 5 oscillations of the sinusoid were accommodated in the Gaussian window. This helped in reducing the size of the dictionary, which in turn helped in reducing the



(a) Dimensions in inches.

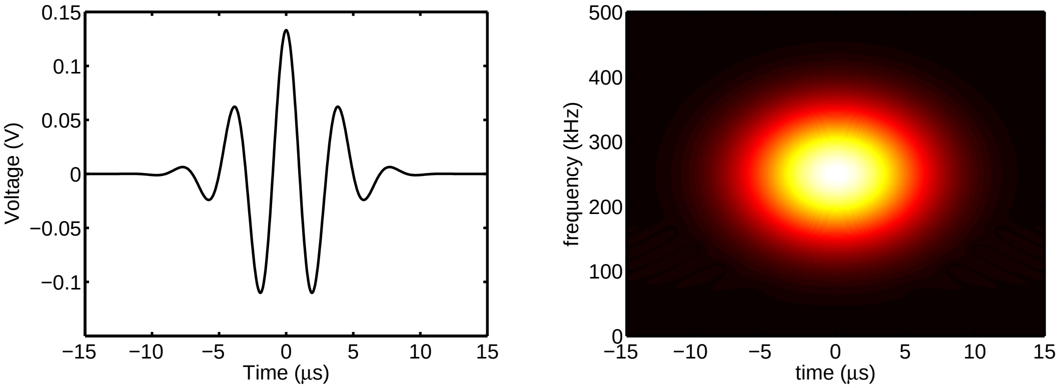
(b) Sensor placement and crack location.

Fig. 6.1. The lug sample under investigation.

dimensionality of the MPD features. The feature extraction process was made 20 times faster. We considered $N = 10$ MPD iterations in this application. The MPD residue error at each of the 10 iterations and the MPD based time-frequency representation (MPD-TFR) for signals from two different damage classes are shown in Figure 6.3. Let us assume that the time-shift, frequency-shift and MPD coefficient of the MPD atoms chosen at any nth iteration is given by τ_n, ν_n and α_n, then any signal $s(t)$ can be represented as a sequence

$$s(t) \leftrightarrow \left[\begin{bmatrix} \tau_1^s \\ \nu_1^s \\ \alpha_1^s \end{bmatrix}, \begin{bmatrix} \tau_2^s \\ \nu_2^s \\ \alpha_2^s \end{bmatrix}, \ldots, \begin{bmatrix} \tau_N^s \\ \nu_N^s \\ \alpha_N^s \end{bmatrix} \right]$$

such that $\tau_1 < \tau_2 < \ldots < \tau_N$. This sequential representation of the signals moti-



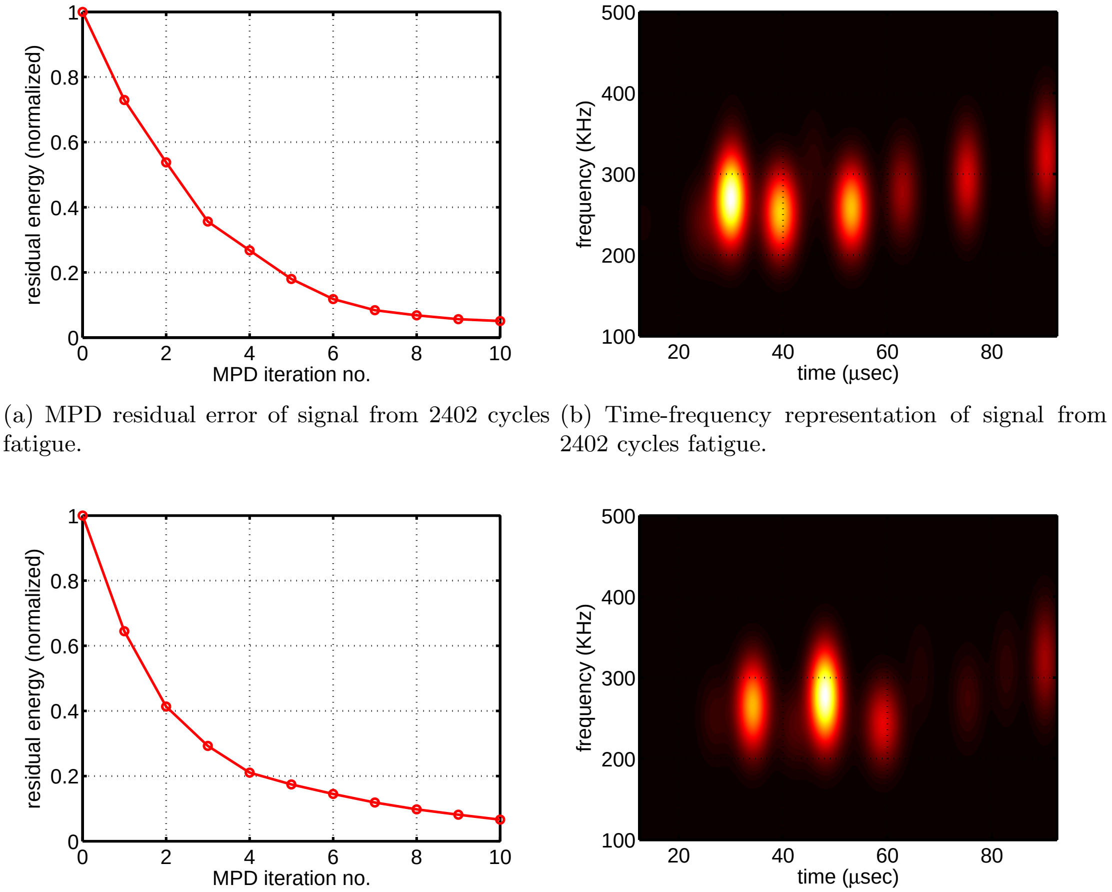
(a) Time plot.

(b) Time-frequency representation.

Fig. 6.2. The actuation signal.

vate us to model each class with a hidden Markov model (HMM) [94]. For each class, $\boldsymbol{\theta}_m = \{A_m, B_m, \boldsymbol{\pi}_m\}$, where A is the transition probability matrix, B is the observation density matrix and $\boldsymbol{\pi}$ is the initial state distribution vector [26]. In this application, we had selected 3 state HMMs, with random initial guess.

Damage classes can be modeled using HMM with signals coming from each sensor. The sensors (1 and 2) that are closest to the damage is expected to be most sensitive to the changes due to damage. Since sensor 1 was the closest, we chose sensor 1 and the source domain. Sensors 2, 3 and 4 are used as target domains. This is shown graphically in Figure 6.3. In the source domain, 50 data points from each class was used for training. On the other hand, we used only 5 data point from each of the target domains. Training in the source and the target domains were conducted using Equations (6.1) and (6.3a) respectively. Next, source domain probability of data association to each class $P\left(\mathbb{Y}_{\text{sor}}^{\text{te}}|\boldsymbol{\theta}_m\right)$, $m = 1, \ldots, 11$ was estimated and the class memberships $\mathbf{c}_{\text{sor}}^{\text{te}}$ were computed as shown in (6.2). The corresponding testing data from the other domains were first classified according to Equation (6.3b) and then TL was used as shown in Equation (6.5). The classification results with and with-



(a) MPD residual error of signal from 2402 cycles fatigue.

(b) Time-frequency representation of signal from 2402 cycles fatigue.

(c) MPD residual error of signal from 2402 cycles fatigue.

(d) Time-frequency representation of signal from 2402 cycles fatigue.

Fig. 6.3. MPD and residual error plots of signal from sensor 1, corresponding to two different fatigue levels.

out TL is shown in Figure 6.5. These figures demonstrate a confusion matrix. It has 11 rows and columns, each corresponding to a damage class. The rows represented the known information and the columns represent the classification result. The colors correspond to the fraction of data classified. For example, if 10% of the signals coming from class 1 are classified as class 2 (which is a mis-classification), we would see 0.1 at the intersection of row 1 and column 2. Using TL, a significant improvement in the classification can be observed. This improvement is summarized in Table 6.1.

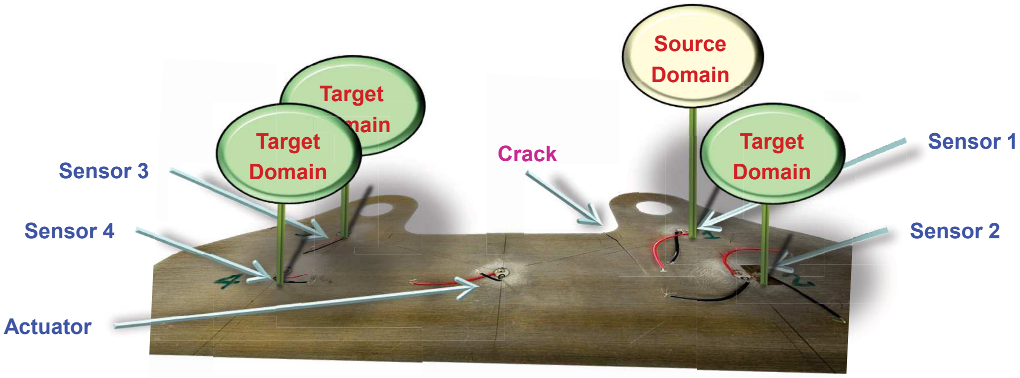
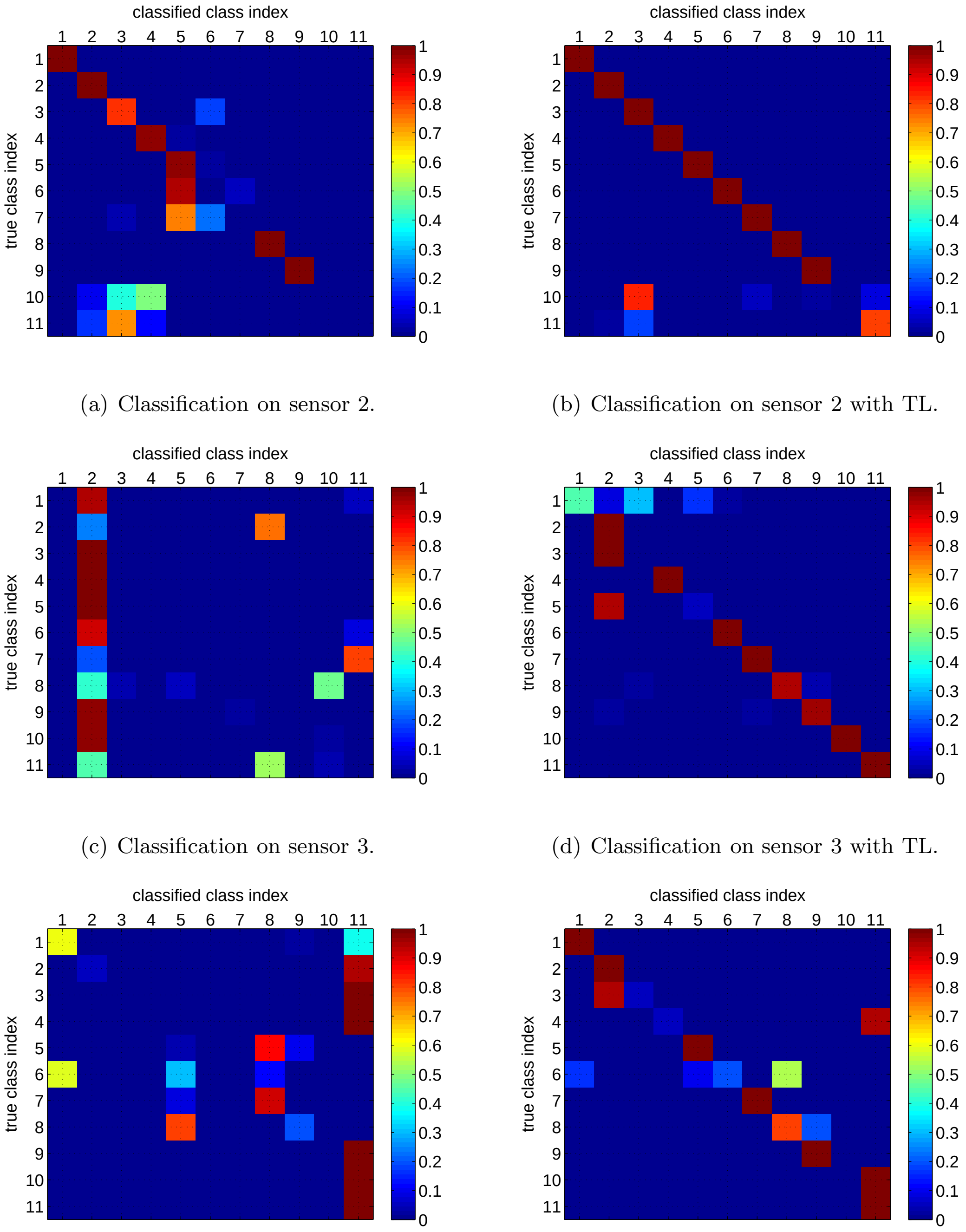


Fig. 6.4. Transfer Learning for Fatigue Damage Classification

Sensors	2	3	4
without TL	62%	2%	15%
with TL	89%	76%	65%

Table 6.1. Summary of correct classification.



(a) Classification on sensor 2.

(b) Classification on sensor 2 with TL.

(c) Classification on sensor 3.

(d) Classification on sensor 3 with TL.

(e) Classification on sensor 4.

(f) Classification on sensor 4 with TL.

Fig. 6.5. Graphical representation of confusion matrix.

CHAPTER 7

Conclusion and Future Directions

7.1. Conclusions

In this work, we proposed new methodologies for processing structural health monitoring (SHM) signals. The first method we proposed is based on decomposing measurements (obtained from sensors placed on structures) using the matching pursuit decomposition (MPD) algorithm and then utilizing joint time-frequency analysis to discriminate between data from various damage conditions. Two different dictionaries were considered for the decomposition. The first dictionary consisted of time-frequency shifted and scaled versions of a basic Gaussian atom; the second dictionary used time-frequency shifted versions of real measured data from the sensors (modified MPD or MMPD). The MPD-TFR classifier used training and validation data sets to compute representative MPD-TFRs for each damage class and validate model parameters. Classification of test data was performed using 2-D time-frequency correlations between MPD-TFRs of training data and the MPD-TFR of the test signals. The MMPD classifier used training data to build the dictionary, and signals were again classified according to the magnitude of the projections onto the various damage classes. A Bayesian sensor (decision) fusion approach was implemented to improve classification performance by statistically combining the information collected by multiple distributed sensors.

When applied to the detection of fastener failure damage in an aluminum plate, the MPD-TFR classifier resulted in an average correct classification rate of 91.6% and the MMPD classifier resulted in an average correct classification rate of 98.8%. The MMPD classifier resulted in more accurate classification than the MPD-TFR classifier, while remaining computationally efficient as it uses a smaller dictionary and requires fewer MPD iterations. Note, however, that the MPD-TFR classifier has some analytical and computational advantages, due to the availability of closed-form expressions because of the Gaussian dictionary atoms. Importantly, the dictionary is not limited by the acquired training data. The Bayesian decision fusion improved the average correct classification rate of the MPD-TFR classifier to 97.4%. The decision fusion procedure requires additional data and computational effort in order to learn the statistics of the local classifiers.

A structural damage classification technique based on ultrasonic sensing and time-frequency analysis was also presented. Ultrasonic measurements taken across a fatigue damaged region in Al-6061 and Al-2024 plates demonstrated that material stiffness is largely unaffected by damage, whereas attenuation (here related to ultrasonic amplitudes) is related to damage. This can be seen by comparing the longitudinal and shear amplitudes of the waveforms from various regions. Data collected by ultrasonic sensors was decomposed into the time-frequency plane using the MMPD algorithm. The resulting features were then used to classify structural damage. Application to the detection of fatigue damage in Al-6061 and Al-2024 plates demonstrates good classification performance. Note that our classification of the ultrasonic data demonstrated that results from non-destructive evaluation (NDE) can be used in conjunction with novel classification techniques to detect damage at very small (micro) length scales for which off-the-shelf sensors and vibration/impedance/conventional wave based techniques cannot be used. Both experimental and modeling efforts are currently underway to obtain a sample set that is more representative statistically.

We also consider a new SHM signal processing algorithm for the classification of structural damage based on time-frequency feature extraction and continuous hidden Markov models (HMMs). We applied the algorithm to the detection of fatigue-induced structural and material damage using Lamb waves in a bolted joint and it resulted in very good performance; the average correct classification rates observed were nearly 90%. Note that we compared the MPD-TFR and the HMM based damage classification algorithms in [23].

The performance of the discrete HMM classifier is a function of the number of symbols used; the accuracy of the algorithm increases with the number of symbols. The performance of the continuous HMM classifier is superior to that of the discrete HMM classifier. The continuous HMM, however, used a more complex model and has more parameters to be estimated. As a result, the amount of data required for training the continuous HMM model can be greater. In addition, the number of mixture components to use in the Gaussian mixture models has to be determined using techniques such as cross-validation. Moreover, learning and inference in the continuous HMM is generally more computationally intensive than that of the discrete case.

A variational Bayesian learning algorithm was applied to automatically select the number of states used in the HMM. This choice was shown to correlate well with what one might choose empirically when examining the time-frequency features of the data in question.

We also studied the effect of noise on the HMM classifier. Synthetic data was generated using a physics-based model of a lug joint sample. Different realizations of the signals at various noise levels were used to test the proposed HMM based classification technique. By first using an MPD based feature extraction technique, the added noise could significantly be reduced. The resulting discrete HMM damage classifier was observed to perform well even for low SNR values. When applied to the classification of crack damage in lug joints, the performance was good (over 90% correct classification) even for high levels of noise.

As it is not always possible to have sufficient amounts of data for training SHM algorithms under varying environmental conditions, we developed a baseline-free, adaptive learning damage classification technique.

The adaptive methodology is based on the following steps: (a) characteristic time-frequency (TF) features were extracted from sensor observations using MPD; (b) measurements were generated from the TF features by computing a MPD based probability distribution function (MPD-PDF); (c) these measurements were modeled according to a GMM and the GMM parameters were learnt using adaptive learning methodology, resulting in a measurement distribution; (d) with this measurement distribution and a state distribution computed from a physics-based progressive damage model, crack length was progressively estimated using a sequential Bayesian filter. Note that such an MCMC driven adaptive learning approach has never before been integrated in an iterative framework with constantly evolving data. Also, the Bayesian filter framework has been adapted to progressively infer damage by estimating changes in damage. We applied the progressive damage estimation approach to a CT sample specimen, and we obtain high classification performance. The estimates were correct in the order of a thousandth of a millimeter. Thus, the new adaptive damage classification technique has been developed that uses time-frequency damage features to classify previously unseen damages in varying environmental conditions without the need of a priori training.

Note that the current work can be extended to incorporate a hybrid state-space Gaussian process prognostic model [135] that can estimate cracks with greater accuracy. Based on the characteristics of available data, the MPD features may be directly used in a multi-dimensional Gaussian mixture model (GMM) framework to eliminate the computation of the statistical measure of similarity. Note that, an HMM framework is not recommended with the adaptive learning algorithm since HMMs would require a large number of parameters to be learned. In addition, the method can be made more efficient if active data selection methods are incorporated to select optimal data to identify structural changes.

The transfer learning method provided the framework to reuse information learnt from one sensor to use it for classification of data obtained from another. In structural health monitoring applications like we demonstrated in this paper, having data from multiple sensors is usual and under it may be expensive to collect a large set of training data from all the sensors. TL has shown promising results when the trained information from one sensor was used for damage classifications using data from other sensors from which not enough training data was available. The improvements in the classification was over significant. This framework can be extended to using different sensing modalities used in structural health monitoring.

7.2. Future Work

Our new signal processing methodologies for SHM applications have brought forth new desirable developments: how to adaptively learn and classify new structural damage under varying conditions without requiring a priori training. At the same time, the new developments have also brought forth new directions of research such as the ones listed next.

- We want to test the adaptive learning classification algorithm for different data sets and also for different scenarios. We currently do not have any real data to demonstrate the algorithm performance with the effect of environmental conditions on structural damage. Furthermore, as currently the available observations are not at regular intervals, a dataset with more frequent observations at regular intervals (for example, at every 100 cycles of damage) would enable a better testing of the algorithm.

- The current state model is not truly stochastic and does not provide accurate predictions at large crack lengths. We are working in collaboration with the Mechanical Engineers in our group to develop a better model.

- A multidimensional GMM can be used to model the features available from the MPD feature extraction method for the adaptive learning classification algorithm. This would eliminate the need to build a representative set and estimate a statistical measure of similarity. We will then investigate the effect of the multi-dimensional GMM on the algorithm performance in terms of computational time and accuracy.

- Transfer learning has opened new avenues for research which can be used strategically with the proposed adaptive learning methodology for adaptive transfer of information to make a more intelligent structural health management system.

REFERENCES

[1] C. R. Farrar and K. Worden, “An introduction to structural health monitoring,” *Philosophical Transactions of the Royal Society A*, vol. 365, pp. 303–315, 2007.

[2] W. J. Staszewski, C. Boller, and G. Tomlinson, *Structural Health Monitoring of Aerospace Structures, Smart Sensor Technologies and Signal Processing.* John Wiley and Sons, Ltd, 2004.

[3] H. T. Yolken and G. A. Matzkanin, “Recent trends in structural health monitoring technologies,” *The AMMTIAC Quarterly*, vol. 3, no. 4, pp. 3–6, 2008.

[4] C. R. Farrar and N. A. J. Lieven, “Damage prognosis: The future of structural health monitoring,” *Royal Society of London Transactions Series A*, vol. 365, pp. 623–632, 2007.

[5] S. Doebling and C. R. Farrar, “Statistical damage identification techniques applied to the I-40 bridge over the Rio Grande,” in *International Modal Analysis Conference*, Santa Barbara, CA, 1998.

[6] C. Farrar, D. Nix, T. Duffey, P. Cornwell, and G. Pardoen, “Damage identification with linear discriminant operators,” in *International Modal Analysis Conference*, Kissimmee, FL, 1999.

[7] H. Sohn and C. R. Farrar, “Damage diagnosis using time series analysis of vibration signals,” *Smart Material Structures*, vol. 10, pp. 446–451, June 2001.

[8] H. Sohn, C. R. Farrar, N. F. Hunter, and K. Worden, “Structural health monitoring using statistical pattern recognition techniques,” *Transactions of the ASME*, vol. 123, pp. 706–711, 2001.

[9] G. Park, H. Sohn, C. R. Farrar, and D. J. Inman, “Overview of piezoelectric impedance-based health monitoring and the path forward,” *Shock and Vibration Digest*, vol. 35, no. 6, pp. 451–463, November 2003.

[10] L. Gelman, V. Giurgiutiu, and I. Petrunin, “Advantage of using the Fourier components pair instead of power spectral density for fatigue crack diagnostics,” *International Journal of Condition Monitoring and Diagnostic Engineering Management*, vol. 7, pp. 18–22, 2004.

[11] H. Sohn and K. H. Law, “Bayesian probabilistic damage detection of a reinforced-concrete bridge column,” *Earthquake Engineering Structural Dynamics*, vol. 29, pp. 1139–1152, 2000.

[12] M. Nguyen, X. Wang, Z. Su, and L. Ye, “Damage identification for composite structures with a Bayesian network,” in *Intelligent Sensors, Sensor Networks and Information Processing Conference*, 2004, pp. 307–311.

[13] H. Sohn, D. W. Allen, K. Worden, and C. R. Farrar, “Structural damage classification using extreme value statistics,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 127, pp. 125–132, 2005.

[14] S. Das, A. N. Srivastava, and A. Chattopadhyay, “Classification of damage signatures in composite plates using one-class SVMs,” in *Aerospace Conference, 2007 IEEE*, March 2007, pp. 1–19.

[15] W. J. Staszewski, “Structural and mechanical damage detection using wavelets,” *The Shock and Vibration Digest*, vol. 30, pp. 457–472, 1998.

[16] H. Jeong and Y. Jang, “Fracture source location in thin plates using the wavelet transform of dispersive waves,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 47, pp. 612–619, 2000.

[17] A. Sun and C. C. Chang, “Statistical wavelet-based method for structural health monitoring,” *Journal of Structural Engineering*, vol. 130, no. 7, pp. 1055–1062, July 2004.

[18] V. Pakrashia, B. Basu, and A. O. Connora, “Structural damage detection and calibration using a wavelet-kurtosis technique,” *Engineering Structures*, vol. 29, no. 9, pp. 2097–2108, September 2007.

[19] A. Karasaridis, M. Maalej, S. Pantazopoulou, and D. Hatzinakos, “Time-frequency analysis of sensor data for detection of structural damage in instrumented structures,” in *International Conference on Digital Signal Processing*, vol. 2, 1997, pp. 817–820.

[20] M. M. R. Taha, A. Noureldin, J. L. Lucero, and T. J. Baca, “Wavelet transform for structural health monitoring: A compendium of uses and features,” *Structural Health Monitoring*, vol. 5, no. 3, pp. 267–295, 2006.

[21] D. Chakraborty, N. Kovvali, J. Wei, A. Papandreou-Suppappola, D. Cochran, and A. Chattopadhyay, “Damage Classification Structural Health Monitoring in Bolted Structures Using Time-frequency Techniques,” *Journal of Intelligent Material Systems and Structures, special issue on Structural Health Monitoring*, vol. 20, no. 11, pp. 1289–1305, July 2009.

[22] L. Channels, D. Chakraborty, B. Butrym, N. Kovvali, J. Spicer, A. Papandreou-Suppappola, M. Afshari, D. Inman, and A. Chattopadhyay, “A comparative study of

fatigue damage sensing in aluminum alloys using electrical impedance and laser ultrasonic methods," in *Proc. of SPIE, Smart Structures and Materials & Non-destructive Evaluation and Health Monitoring,*, vol. 7295, 2009, pp. 72 950Q–1 – 10.

[23] D. Chakraborty, W. Zhou, D. Simon, N. Kovvali, A. Papandreou-Suppappola, D. Cochran, and A. Chattopadhyay, "Time-frequency methods for structural health monitoring," in *SenSIP workshop*, Sedona, AZ, May 2008.

[24] D. Chakraborty, S. Soni, J. Wei, N. Kovvali, A. Papandreou-Suppappola, D. Cochran, and A. Chattopadhyay, "Physics based modeling for time-frequency damage classification," in *Proc. of SPIE, Smart Structures and Materials & Non-destructive Evaluation and Health Monitoring,*, vol. 6926, 2008, pp. 69 260M1–12.

[25] L. Channels, D. Chakraborty, D. Simon, N. Kovvali, J. Spicer, A. Papandreou-Suppappola, D. Cochran, P. Peralta, and A. Chattopadhyay, "Ultrasonic sensing and time-frequency analysis for detecting plastic deformation in an aluminum plate," in *Proc. of SPIE, Smart Structures and Materials & Non-destructive Evaluation and Health Monitoring,*, vol. 6926, 2008, pp. 69 260P1–10.

[26] W. Zhou, D. Chakraborty, N. Kowali, A. Papandreou-Suppappola, D. Cochran, and A. Chattopadhyay, "Damage classification for structural health monitoring using time-frequency feature extraction and continuous hidden Markov models," in *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers ACSSC 2007*, November 2007, pp. 848–852.

[27] W. Zhou, N. Kovvali, A. Papandreou-Suppappola, P. Peralta, and A. Chattopadhyay, "Progressive damage estimation using sequential Monte Carlo techniques," in *The 7th International Workshop on Structural Health Monitoring*, Stanford, CA, 2009.

[28] Z. Feng and F. Chu, "Nonstationary vibration signal analysis of a hydroturbine based on adaptive chirplet decomposition," *Structural Health Monitoring*, vol. 6, no. 4, pp. 265–279, 2007. [Online]. Available: http://shm.sagepub.com/cgi/content/abstract/6/4/265

[29] J. Michaels and T. Michaels, "Detection of structural damage from the local temporal coherence of diffuse ultrasonic signals," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, vol. 52, no. 10, pp. 1769–1782, October 2005.

[30] L. Yu and V. Giurgiutiu, "Advanced signal processing for enhanced damage detection with piezoelectric wafer active sensors," *Smart Structures and Systems*, vol. 1, no. 2, pp. 185–215, 2005.

[31] B. Liu, S. Ling, and R. Gribonval, “Bearing failure detection using matching pursuit,” *NDTE International*, vol. 35, pp. 255–262, 2002.

[32] V. Giurgiutiu, J. Bao, and W. Zhao, “Active sensor wave propagation health monitoring of beam and plate structures,” in *SPIE International Symposium on Smart Structures and Materials*, March 2001, pp. 1–12.

[33] V. Giurgiutiu and A. Zagra, “Damage detection in thin plates and aerospace structures with the electro-mechanical impedance method,” *Structural Health Monitoring*, vol. 4, no. 2, pp. 99–118, 2005.

[34] A. Graps, “An introduction to wavelets,” *Computational Science & Engineering, IEEE*, vol. 2, no. 2, pp. 50 – 61, 1995.

[35] L. Eren and M. J. Devaney, “Bearing damage detection via wavelet packet decomposition of the stator current,” *IEEE Transactions on Instrumentation and Measurement*, vol. 53, pp. 431–436, 2004.

[36] C. A. Paget, S. Grondel, K. Levin, and C. Delebarre, “Damage assessment in composites by Lamb waves and wavelet coefficients,” *Smart Materials and Structures*, vol. 12, no. 3, pp. 393–402, 2003.

[37] U. Junga and B.-H. Koh, “Structural damage localization using wavelet-based silhouette statistics,” *Journal of Sound and Vibration*, vol. 321, pp. 590–604, 2009.

[38] Z. Hou, M. Noori, and R. S. Amand, “Wavelet-based approach for structural damage detection,” *Journal of Engineering Mechanics*, vol. 12, no. 7, pp. 677–683, 2000.

[39] A. Nuruzzaman, O. Boyraz, and B. Jalali, “Time-stretched short-time Fourier transform,” *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 2, pp. 598 – 602, 2006.

[40] R. A. Altes, “Detection, estimation and classification with spectrograms,” *The Journal of the Acoustic Society of America*, vol. 67, no. 4, pp. 1232–1246, April 1980.

[41] L. Cohen, “Time-frequency distribution - A review,” *IEEE proceedings*, vol. 77, no. 7, pp. 941–981, July 1989.

[42] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. on Signal Processing*, vol. 41, pp. 3397–3415, December 1993.

[43] P. Seung-Hee, Y. Chung-Bang, and R. Yongrae, "PZT-induced lamb waves and pattern recognitions for on-line health monitoring of jointed steel plates," in *Sensors and smart structures technologies for civil, mechanical, and aerospace systems :*, vol. 5765. San Diego, CA: International Society of Photo-Optical Instrumentation Engineers, Bellingham, WA, March 2005.

[44] S. E. Olson, M. P. DeSimio, and M. M. Derriso, "Fastener damage estimation in a square aluminum plate," *Structural Health Monitoring*, vol. 5, pp. 173–183, 2006.

[45] N. Kovvali, S. Das, D. Chakraborty, D. Cochran, A. Papandreou-Suppapola, and A. Chattopadhyay, "Time-frequency based classification of structural damage," in *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, Hawaii, April 2007, pp. 2047–2055.

[46] D. Chakraborty, N. Kovvali, J. Zhang, A. Papandreou-Suppappola, and A. Chattopadhyay, "Adaptive learning for damage classification in structural health monitoring," in *43rd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, November 2009.

[47] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The Annals of Statistics*, vol. 1, pp. 209–230, 1973.

[48] C. E. Antoniak, "Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems," *The Annals of Statistics*, vol. 2, pp. 1152–1174, 1974.

[49] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, vol. 90, pp. 577–588, 1995.

[50] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Advances in Neural Information Processing Systems 12*, S. A. Solla, T. K. Leen, and K.-R. Muller, Eds. MIT Press, 2000, pp. 554–560.

[51] A. Ranganathan, "The Dirichlet process mixture (dpm) model," September 2004. [Online]. Available: http://biocomp.bioen.uiuc.edu/journal_club_web/dirichlet.pdf

[52] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121–144, 2006.

[53] P. Fearnhead, "Particle filters for mixture models with an unknown number of components," *Journal of Statistics and Computing*, vol. 14, pp. 11–21, 2004. [Online]. Available: citeseer.ist.psu.edu/754210.html

[54] Y. Qi, J. Paisley, and L. Carin, "Music analysis using hidden Markov mixture models," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5209–5224, 2007.

[55] X. Wang, X. Ma, and W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical Bayesian models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 31, no. 3, pp. 539–555, Mar. 2009.

[56] E. Ozkan, I. Y. Ozbek, and M. Demirekler, "Dynamic speech spectrum representation and tracking variable number of vocal tract resonance frequencies with time-varying Dirichlet process mixture models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1518–1532, Nov. 2009.

[57] F. Caron, M. Davy, A. Doucet, E. Duflos, and P. Vanheeghe, "Bayesian inference for linear dynamic models with Dirichlet process mixtures," *IEEE Transaction on Signal Processing*, vol. 56, no. 1, pp. 71–84, Jan. 2008.

[58] I. Pruteanu-Malinici and L. Carin, "Infinite hidden Markov models for unusual-event detection in video," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 811–822, May 2008.

[59] W. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[60] N. Metropolis, A. Rosenbluth, M. N. Rosenbluth, A.Teller, and H. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1091, 1953.

[61] N. Metropolis and S. Ulam, "The Monte Carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, September 1949.

[62] S. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, pp. 1 – 15, 2009, pre-published version.

[63] S. Thrun, "Is learning the n-th thing any easier than learning the first?" in *Advances in Neural Information Processing Systems*. The MIT Press, 1996, pp. 640–646.

[64] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[65] S. Thrun and L. Y. Pratt, Eds., *Learning to learn.* Kluwer Academic Publishers, 1998.

[66] J. Baxter, “A model of inductive bias learning,” *Journal of Artificial Intelligence Research*, vol. 12, pp. 149–198, 2000.

[67] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught learning: Transfer learning from unlabeled data,” in *Proceedings of the 24th international conference on Machine learning*, 2007.

[68] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, “Translated learning: Transfer learning across different feature spaces,” in *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS-08)*, Vancouver, Canada, 2008.

[69] A. Arnold, R. Nallapati, and W. W. Cohen, “A comparative study of methods for transductive transfer learning,” in *7th IEEE International Conference on Data Mining Workshops*, IEEE Computer Society, Washington, DC, USA, 2007, pp. 77 – 82.

[70] G. Hinton and T. J. Sejnowski, Eds., *Unsupervised Learning: Foundations of Neural Computation.* MIT Press, 1999.

[71] D. G. S. Richard O. Duda, Peter E. Hart, *Pattern classification*, 2nd ed. New York: Wiley, 2001, ch. Unsupervised Learning and Clustering, p. 571.

[72] S. Kotsiantis and P. Pintelas, “Recent advances in clustering: A brief survey,” *WSEAS Transactions on Information Science and Applications*, vol. 1, no. 1, pp. 73–81, 2004.

[73] S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. Shevitz, “Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review,” Los Alamos National Laboratory, New Mexico, Tech. Rep. LA-13070-MS, 1996.

[74] H. Sohn, C. R. Farrar, F. M. Hemez, D. D. Shunk, D. W. Stinemates, and B. R. Nadler, “A review of structural health monitoring literature: 1996 – 2001,” Los Alamos National Laboratory, Tech. Rep. LA-13976-MS, 2003.

[75] H. Sohn and C. R. Farrar, “Time series analyses for locating damage sources in vibration systems,” in *International Conference on Noise and Vibration Engineering*, Leuven, Belgium, September 2000.

[76] D. George, N. Hunter, C. Farrar, and R. Deen, “Identifying damage sensitive features using nonlinear time-series and bispectral analysis,” in *IMAC 18*, San Antonio, Texas, February 2000.

[77] V. Giurgiutiu and J. Bao, "Embedded-ultrasonics structural radar for *In Situ* structural health monitoring of thin-wall structures," *Structural Health Monitoring*, vol. 3, no. 2, pp. 121–140, 2004.

[78] O. Balogun, G. Cole, R. Huber, D. Chinn, T. Murray, and J. Spicer., "High spatial resolution sub-surface acoustic microscopy in mesoscale structures using a laser-based ultrasonic technique," *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 2008.

[79] O. Balogun, R. Huber, D. Chinn, and J. Spicer, "Laser ultrasonic inspection of the microstructural state of thin metal foils," *Journal of the Acoustical Society of America*, vol. 125, no. 3, pp. 1437–1443, 2009.

[80] V. Rao, "Kurtosis as a metric in the assessment of gear damage," *The Shock and Vibration Digest*, vol. 31, no. 6, pp. 443–448, 1999.

[81] L. J. Hadjileontiadisa and E. Doukab, "Kurtosis analysis for crack detection in thin isotropic rectangular plates," *Engineering Structures*, vol. 29, no. 9, pp. 2353–2364, September 2007.

[82] B. P. Bogert, M. J. R. Healy, and J. W. Tukey, *The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo Autocovariance, Cross-Cepstrum and Saphe Cracking.* New York: Wiley, 1963, ch. 15, pp. 209–243.

[83] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing*, 4th ed. Prentice Hall, 2004.

[84] A. Morassi and F. Vestroni, Eds., *Dynamic Methods for Damage Detection in Structures*, 1st ed. Springer, 2009.

[85] S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. Academic Press, 1998.

[86] L. Cohen, *Time Frequency Analysis: Theory and Applications.* Prentice Hall, 1994.

[87] F. Hlawatsch and G. F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representations," *IEEE Signal Processing Magazine*, vol. 9, no. 2, pp. 21–67, April 1992.

[88] A. Papandreou-Suppappola, Ed., *Applications in Time-Frequency Signal Processing.* Florida: CRC Press, 2002.

[89] A. Papandreou-Suppappola and S. B. Suppappola, “Analysis and classification of time-varying signals with multiple time-frequency structures,” *IEEE Signal Processing Letters*, vol. 9, pp. 92–95, 2002.

[90] S. P. Ebenezer, A. Papandreou-Suppappola, and S. B. Suppappola, “Classification of acoustic emissions using modified matching pursuit,” *EURASIP Journal on Applied Signal Processing*, vol. 3, pp. 347–357, 2004.

[91] L. Channels, D. Chakraborty, D. Simon, N. Kovvali, J. Spicer, A. Papandreou-Suppappola, D. Cochran, P. Peralta, and A. Chattopadhyay, “Ultrasonic sensing and time-frequency analysis for detecting plastic deformation in an aluminum plate,” in *International Symposium on Smart structures and Materials & Nondestructive Evaluation and Health Monitoring*, vol. 6926, 2008.

[92] D. Chakraborty, S. Soni, J. Wei, N. Kovvali, A. Papandreou-Suppappola, D. Cochran, and A. Chattopadhyay, “Physics based modeling for time-frequency damage classification,” in *International Symposium on Smart structures and Materials & Nondestructive Evaluation and Health Monitoring*, vol. 6926, 2008.

[93] L. R. F. Rose, “Point-source representation for laser-generated ultrasound,” *Journal of the Acoustical Society of America*, vol. 75, pp. 723–732, 1984.

[94] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” in *Proceedings of the IEEE*, vol. 77, 1989, pp. 257–286.

[95] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[96] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[97] M. J. Beal, “Variational algorithms for approximate Bayesian inference,” Ph.D. dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.

[98] D. J. C. MacKay, “Ensemble learning for hidden Markov models,” Cavendish Laboratory, University of Cambridge, Tech. Rep., 1997.

[99] S. Ji, B. Krishnapuram, and L. Carin, “Variational Bayes for continuous hidden Markov models and its application to active learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 522–532, 2006.

[100] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[101] S. Weinzierl, "Introduction to monte carlo methods," 2000. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:hep-ph/0006269

[102] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice.* Chapman & Hall/CRC, 1996.

[103] A. F. M. Smith, "Bayesian computational methods," *Philosophical Transactions: Physical Sciences and Engineering*, vol. 337, no. 1647, pp. 369–386, 1991. [Online]. Available: http://www.jstor.org/stable/53988

[104] M. Evans and T. Swartz, "Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems," *Statistical Science*, vol. 10, no. 3, pp. 254–272, 1995.

[105] M. A. Tanner, *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*, 3rd ed., ser. Springer Series in Statistics. Springer, 1996.

[106] M. K. Cowles and B. P. Carlin, "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review," *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 883–904, Jun 1996.

[107] A. E. Raftery and S. Lewis, "How many iterations in the gibbs sampler," in *In Bayesian Statistics 4*, vol. 4, 1992, pp. 763–773. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.6474

[108] A. E. Raftery and S. M. Lewis, "Comment: One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo," *Statistical Science*, vol. 7, no. 4, pp. 493 – 497, November 1992.

[109] A. Gelman and D. B. Rubin, "A single series from the Gibbs sampler provides a false sense of security," *Bayesian Statistics*, vol. 4, pp. 625 – 631, 1992.

[110] ——, "Inference from iterative simulation using multiple sequences," *Statistical Science*, vol. 7, no. 4, pp. 457 – 511, 1992.

[111] G. Casella and E. I. George, "Explaining the Gibbs sampler," *The American Statistician*, vol. 46, no. 3, pp. 167 – 174, August 1992.

[112] A. E. Gelfand and A. F. M. Smith, “Sampling-based approaches to calculating marginal densities,” *Journal of the American Statistical Association*, vol. 85, no. 410, pp. 398 – 409, June 1990.

[113] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. CRC Press, 2004.

[114] Y. W. Teh, “Dirichlet processes,” 2007, submitted to Encyclopedia of Machine Learning. [Online]. Available: www.gatsby.ucl.ac.uk/~ywteh/research/npbayes/dp.pdf

[115] H. Ishwaran and L. F. James, “Gibbs sampling methods for stick-breaking priors,” *Journal of the American Statistical Association*, vol. 96, pp. 161–173, 2001.

[116] J. Sethuraman, “A constructive definition of Dirichlet priors,” *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[117] D. Blackwell and J. B. MacQueen, “Ferguson distributions via Polyá urn schemes,” *The Annals of Statistics*, vol. 1, pp. 353–355, 1973.

[118] K. Ni, Y. Qi, and L. Carin, “Multi-aspect target classification and detection via the infinite hidden Markov model,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 2007, pp. II–433–II–436.

[119] Y. Qi, J. W. Paisley, and L. Carin, “Dirichlet process HMM mixture models with application to music analysis,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 2007, pp. II–465–II–468.

[120] M. West, P. Muller, and M. D. Escobar, “Hierarchical priors and mixture models, with applications in regression and density estimation,” in *Aspects of Uncertainty*, P. R. Freeman and A. F. Smith, Eds. John Wiley, 1994, pp. 363–386.

[121] R. M. Neal, “Markov chain sampling methods for Dirichlet process mixture models,” *Journal of Computational and Graphical Statistics*, vol. 9, pp. 249–265, 2000.

[122] S. Kullback and R. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[123] S. Kullback, “The Kullback-Leibler distance,” *The American Statistician*, vol. 41, pp. 340–341, 1987.

[124] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of the Calcutta Mathematical Society*, vol. 35, pp. 99–109, 1943, mR0010358.

[125] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology*, vol. 15, no. 1, pp. 52–60, 1967.

[126] G. L. Yang, L. Cam, and L. M., *Asymptotics in Statistics: Some Basic Concepts.* Berlin: Springer., 2000, iSBN 0-387-95036-2.

[127] D. Chakraborty, N. Kovvali, A. Papandreou-Suppappola, and A. Chattopadhyay, "Active learning data selection for adaptive online structural damage estimation," in *Proc. of SPIE, Smart Structures and Materials & Non-destructive Evaluation and Health Monitoring*, 2010.

[128] L. Kuipers and H. Niederreiter, *Uniform Distribution of Sequences*, L. Bers, P. Hilton, and H. Hochstadt, Eds. John Wiley & Sons Inc., 1974.

[129] B. Chazelle, *The Discrepancy Method.* Cambridge University Press, 2002.

[130] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, NY: Cambridge University Press, 2004.

[131] H. Niederreiter, "Discrepancy and convex programming," *Annali di Matematica Pura ed Applicata*, vol. 93, pp. 89–97, 1972.

[132] D. P. Dobkin and D. Eppstein, "Computing the discrepancy," in *Proceedings of the ninth Annual Symposium on Computational Geometry*, 1993, pp. 47–52.

[133] D. P. Dobkin, D. Eppstein, and D. P. Mitchell, "Computing the discrepancy with applications to supersampling patterns," *ACM Transactions on Graphics (TOG)*, vol. 15, pp. 354–376, 1996.

[134] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Pearson Education Inc., 2009.

[135] S. Mohanty, R. Teale, A. Chattopadhyay, P. Peralta, and C. Willhauck, "Mixed Gaussian process and state-space approach for fatigue crack growth prediction," in *Structural Health Monitoring*, F.-K. Chang, Ed. PA, USA: DEStech Publications, Inc., 2007.

[136] A. Ray and R. Patankar, “Fatigue crack growth under variable-amplitude loading: Part I – Model formulation in state-space setting,” *Applied Mathematical Modelling*, vol. 25, no. 11, pp. 979–994, November 2001.

[137] ——, “Fatigue crack growth under variable-amplitude loading: Part II - Code development and model validation,” *Applied Mathematical Modelling*, vol. 25, no. 11, pp. 995–1013, November 2001.

[138] J. N. Yang and S. D. Manning, “A simple second order approximation for stochastic crack growth analysis,” *Engineering Fracture Mechanics*, vol. 53, no. 5, pp. 677–686, March 1996.

[139] S. Mohanty, R. Teale, A. Chattopadhyay, P. Peralta, and C. Willhauck, “Mixed Gaussian process and state-space approach for fatigue crack growth prediction,” in *International Workshop on Structural Heath Monitoring*, vol. 2, 2007, pp. 1108–1115.

[140] *ABAQUS*, 2007, version 6.7.1.

[141] J. L. Rose, Ed., *Ultrasonic Waves in Solid Media.* Cambridge: University Press, Cambridge, 1999.

[142] M. Abramowitz and I. Stegun, *Handbook of Mathematical Functions.* [Online]. Available: http://www.math.hkbu.edu.hk/support/aands/toc.htm

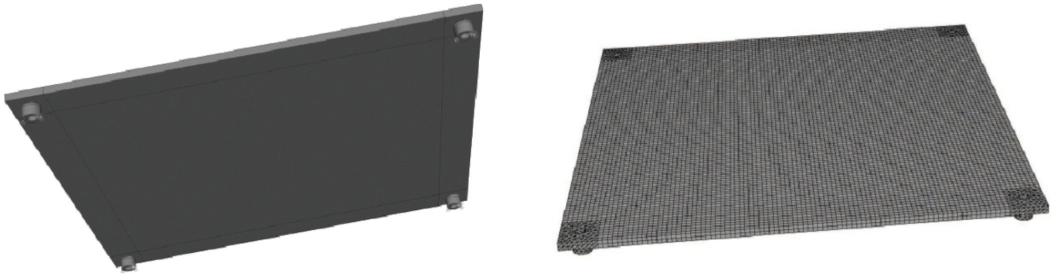
[143] F. Dominici, G. Parmigiani, and M. Clyde, “Conjugate analysis of multivariate normal data with incomplete observations,” *The Canadian Journal of Statistics*, vol. 28, no. 3, pp. 533–550, September 2000.

[144] J. Paisley and L. Carin, “Hidden Markov models with stick-breaking priors,” *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3905–3917, 2009.

[145] D. Fink, “A compendium of conjugate priors,” Montana State Univeristy, Tech. Rep., 1995.

APPENDIX A

THREE-DIMENSIONAL FINITE ELEMENT MODELING ON BOLTED PLATE



(a) Boundary condition connects and washers.

(b) Meshes.

Fig. A.1. Boundary conditions and meshes for the aluminum plate.

In practical applications with irregular boundaries, finite element analysis (FEA) methods are used with the general purpose nonlinear finite element software Abaqus [140]. The geometry of the aluminum plate and the holes and their locations (see Figure A.1) that were modeled using 3-D FEA are identical to those used in the experiments in [44]. The boundary conditions of the plate bolted at the four corners are simulated such that each bolt is supported on a linear elastic spring along three coordinate axes with spring stiffness k_1, k_2, and k_3, respectively. The varying fastener torque affects the support stiffness from the bolts and nuts, and changing the spring stiffness affects the natural frequencies of the plate. To simulate the support stiffness, a connecting washer is inserted between the nuts and the plate. The stiffness of the washer is calibrated such that the first natural frequency from the simulation closely approximates the experimental result [44]. Based on the assumption that the contact area between the nut and the washer is annular, the four supporting hexagonal nuts have been simulated as circular in shape. Hence, the plate is connected to the washers with the support stiffness, and the washers are connected to the nuts. The ends of the nuts are fixed as shown in Figure A.1(a).

Two cases are considered in the analysis. The first case involves the support stiffness of the washer being calibrated using the frequencies of all the four bolts at 100% torque. This

	Young's modulus E, GPa	Poisson's ratio ν	plate density ρ, kg/m^3
Aluminum	70	0.33	2870
Steel	200	0.32	7800
Washer at 100% torque	8	0.32	1000
Washer at 25% torque	2.5	0.32	1000

Table A.1. Material properties used for washers, plate, and nuts.

was carried out by the authors of [44] using experimental modal analysis. In the second case, three bolts were considered at 100% torque and one bolt at 25% torque. The stiffness on the three bolts at 100% torque were chosen to be the same as before but the stiffness on the 25% torque was calibrated using the experimental modal analysis (EMA) results [44], as the correct stiffness of the bolt at this torque is not known. The FEA meshes are shown in Figure A.1(b). The center part of the plate was modeled as a 20-node quadratic brick with reduced integration elements ($C_3D_{20}R$) [140]. A 10-node modified quadratic tetrahedron element ($C_3D_{10}M$) was used for the four corners, and a 4-node linear tetrahedron element (C_3D_4) was used for the nuts and washers. In total, 24,863 elements, 121,719 nodes, and 365,157 degrees of freedom were used in the modal analysis. There were two elements along the thickness of the plate, and the element length was about 3 mm.

During the calibration, a simple way to adjust the support stiffness was by altering the Young's modulus of the washer. The final material properties used in this study for the washers, plate, and nuts are listed in Table A.1. The simulated natural frequencies that lie within 1.5 kHz are listed in Tables A.2 (all bolts at 100%) and A.3 (one bolt at 25%). Note that the EMA results were not available for all the modes. Those entries in the tables are marked with a dashed line.

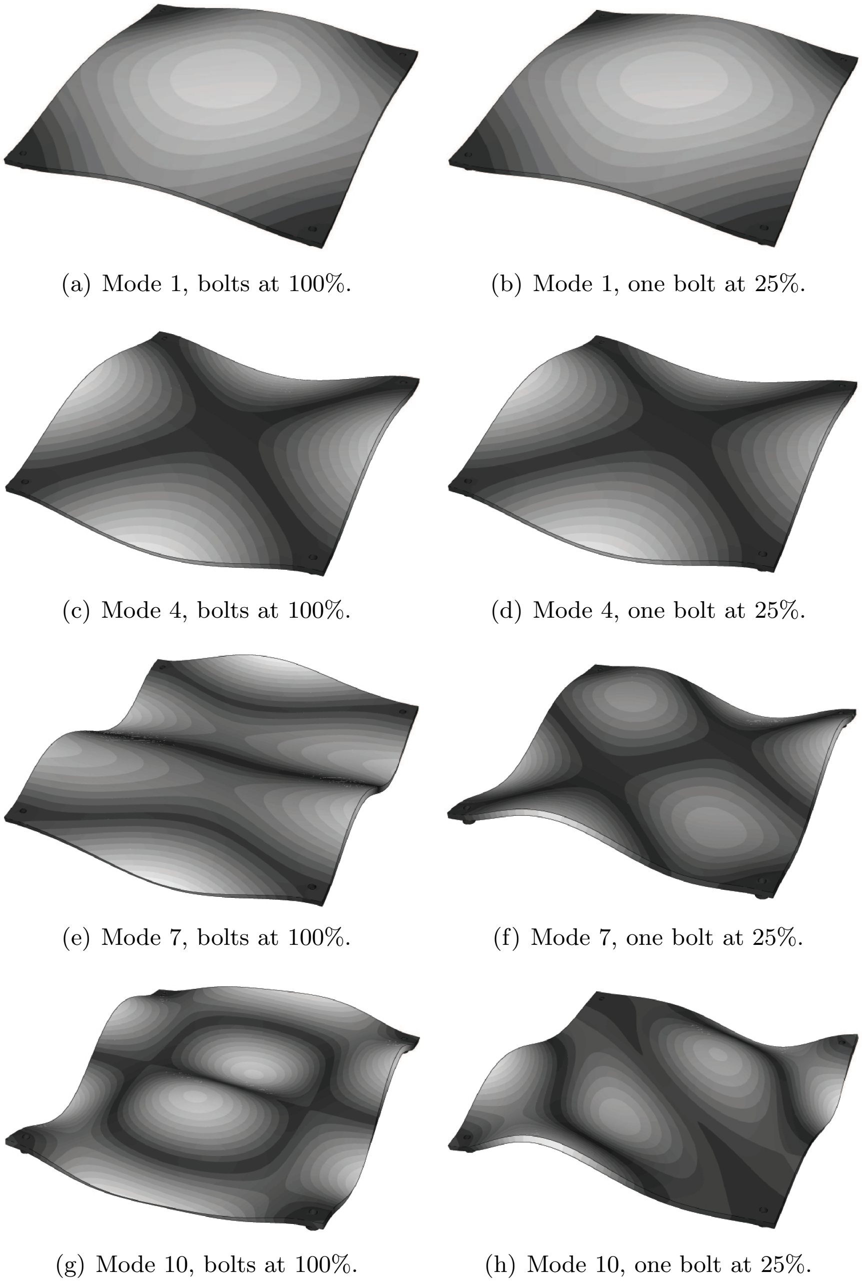
Mode	EMA (Hz)	FEA (Hz)	Difference (%)
1	185.05	185.1	0.02702
2	343.28	339.42	1.12445
3	—	339.6	—
4	388.78	382.53	1.60759
5	706.06	696.13	-1.4064
6	786.00	786.44	0.05598
7	843.37	853.7	1.224848
8	—	854.09	—
9	1119.75	1118.4	-0.12056
10	1263.84	1266.4	0.202557
11	—	1267.1	—
12	1591.14	1577.9	0.83211

Table A.2. Modal frequencies with all bolts at 100% (dashed lines indicate data unavailable).

Mode	EMA (Hz)	FEA (Hz)	Difference (%)
1	181.25	181.30	0.027586
2	336.80	331.67	1.52316
3	—	335.86	—
4	384.76	377.36	1.92328
5	703.81	692.31	-1.63396
6	775.27	776.46	0.153495
7	831.46	840.96	1.142568
8	—	851.38	—
9	1106.58	1106.1	-0.04338
10	1254.98	1253.3	0.13387
11	—	1268.1	—
12	1580.07	1567.6	0.78921

Table A.3. Modal frequencies with one bolt at 25% torque and three bolts at 100% torque (dashed lines indicate data unavailable).

It can seen that the simulated results are in close agreement with the experimental ones. This is expected as FEA can simulate a structure with complex geometry and complex boundary conditions. In practice, there may be differences between the EMA and the numerical analysis. For example, in the EMA, there is an experimental measuring error that depends on the instrumental precision, simplifications in the FEA analyzed model, and boundary conditions. Nevertheless, good instrumentation and well-designed modeling can lead to good agreement of the results between EMA and FEA. The verified results confirm that the model developed is accurate. Figure A.2 shows the similarities and differences between the mode shapes in the material for four chosen modes with all bolts at 100% and one bolt at 25%. It is important to note that due to the symmetry of the plate, the amount of bolt looseness does not affect the mode shape of the plate. For example, modes 1 and 4 are similar (Figures A.2(a) and A.2(d)). This implies that modal response based damage detection techniques can lead to false conclusions.



(a) Mode 1, bolts at 100%.

(b) Mode 1, one bolt at 25%.

(c) Mode 4, bolts at 100%.

(d) Mode 4, one bolt at 25%.

(e) Mode 7, bolts at 100%.

(f) Mode 7, one bolt at 25%.

(g) Mode 10, bolts at 100%.

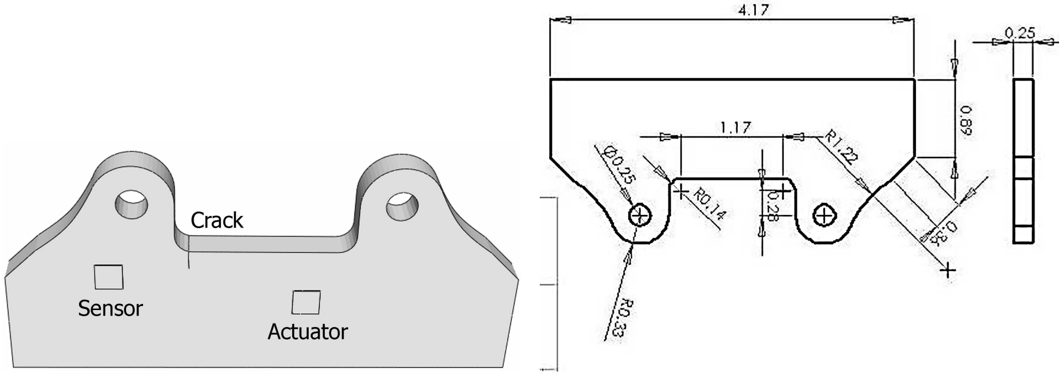
(h) Mode 10, one bolt at 25%.

Fig. A.2. Example modes under different boundary conditions.

APPENDIX B

PHYSICS BASED THREE-DIMENSIONAL FINITE ELEMENT MODELING OF LUG JOINT SAMPLE

The lug sample modeled is shown in Figure B.1(a). The considered sample was made from 0.25 in. thick Al 2024-T3 alloy plate with a polished surface finish. The dimensions of the sample are shown in Figure B.1(b). The model essentially had three parts: the lug, PZT and an adhesive layer as shown in Figure B.2(b). The lug and the adhesive layer each were modeled as isotropic, homogeneous material. The properties for alumina 2024/T351 were assigned to the lug. Continuum three-dimensional brick elements (C3D8R) were used to model both. In the lug model, the size of the mesh was different for samples with different levels of crack (notch) lengths. In this model, a crack length of 6 mm was considered. The PZT was modeled as an orthotropic material with the properties defined in the local orientation system and continuum three-dimensional piezoelectric elements were used. A Tie constraint was used to define the interaction between the lug and the adhesive layer and also between the adhesive layer and the PZTs.



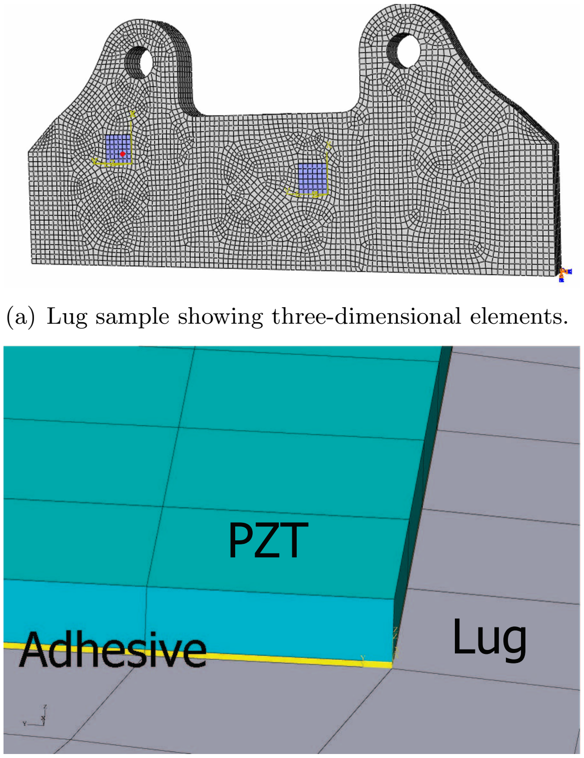
(a) Sensor placement and sight of crack.

(b) Dimensionality.

Fig. B.1. Modeled lug joint sample showing sensor location, crack and dimensions.

With these model specifications, a three-dimensional finite element modeling (FEM) was carried out for the lug sample discussed above using the commercial finite element software Abaqus/Standard (see Figure B.2(a)).

All the degrees of freedom were suppressed at one of the nodes of the lug as indicated at the bottom right corner in Figure B.2(a). A peak to peak voltage of 100 V was applied to the upper surface of the PZT while the bottom surface was maintained at zero volts. A tone burst signal with central frequency 130 KHz was used as the actuation signal.



(a) Lug sample showing three-dimensional elements.

(b) Lug, adhesive and PZT layers.

Fig. B.2. Three-dimensional FEM with boundary condition and layers.

In our specific problem, to solve the wave propagation in lug joints, the elastodynamic wave equation is solved using FEM. The specific geometry was defined in the Abaqus software. The basic elastodynamic equation in a homogeneous anisotropic solid can be written as [141]

$$C_{ijkl}(x_n)\frac{\partial^2}{\partial x_j x_l}u_k(x_n,t)+F_i(x_n,t)=\rho(x_n)\frac{\partial^2 u_i(x_n,t)}{\partial^2 t}, \tag{B.1}$$

where $C_{ijkl}(x_n)$ are the material constants, u_i are the displacement components F_i denotes the body force per unit volume and i, j, k, l and n take values 1, 2 and 3. In the subsequent formulation, all subscripts correspond to the usual index notation in three-dimensional space, and δ_{ij} is the Kronecker delta symbol. However, considering the aluminum specimen as the isotopic material used for our experiment and modeling the elastodynamic equation reduced to

$$(\lambda + 2\mu)u_{j,ij} - \mu\epsilon_{ijk}\epsilon_{kmn}u_{n,mj} + P_i f(t)\delta(x_p) = \rho\,\ddot{u}_i\ . \tag{B.2}$$

Here λ and μ are two Lame' constants of aluminum material. P_i is the magnitude of the external force provided on the body and $f(t)$ is the time dependent unit wave packet acting on the body. Hence, the particle in motion is subjected to time dependent excitation. Please note that the above expressions are valid at a point in the body. To solve a global problem, external force is provided by an actuator with the burst wave as specified earlier. Hence, each point in the body is subjected to external time dependent force and the problem is solved by modeling the problem in Abaqus. Fixed time steps were considered to solve the problem and isoparametric elements were used to model the complete body. The traction free boundary condition was provided at the lug joint boundary.

In this example, two damage cases were modeled. One was unfatigued (with no defects) and the other was fatigued (with a 6 mm crack).

APPENDIX C

LOG-NORMAL DISTRIBUTION

C.1. Log-normal

If a random variable χ is drawn according to log-normal distribution,

$$\chi \sim \mathit{Log}\text{-}\mathcal{N}\left(\mu_\chi, \sigma_\chi\right), \tag{C.1}$$

it can be parameterized by μ_χ and σ_χ such that

$$\begin{aligned}
\mathbf{E}\left[\chi\right] &= e^{(\mu_\chi + \frac{1}{2}\sigma_\chi^2)} \\
\mathbf{Var}\left[\chi\right] &= (e^{\sigma_\chi^2} - 1)e^{(2\mu_\chi + \sigma_\chi^2)}
\end{aligned}$$

are the mean and variance respectively, of the distribution. The parameters μ_χ and σ_χ can be computed as

$$\mu_\chi = \log(\mathbf{E}\left[\chi\right]) - \frac{1}{2}\log\left(1 + \frac{\mathbf{Var}\left[\chi\right]}{\mathbf{E}\left[\chi\right]^2}\right) \tag{C.2a}$$

$$\sigma_\chi = \sqrt{\log\left(1 + \frac{\mathbf{Var}\left[\chi\right]}{\mathbf{E}\left[\chi\right]^2}\right)} \tag{C.2b}$$

for a desired mean $\mathbf{E}\left[\chi\right]$ and variance $\mathbf{Var}\left[\chi\right]$. The likelihood of χ distributed according to the log-normal distribution can be expressed as

$$P\left(\chi | \chi \sim \mathit{Log}\text{-}\mathcal{N}\left(\mu_\chi, \sigma_\chi\right)\right) = \frac{1}{\chi\sigma_\chi\sqrt{2\pi}}\left(e^{-\frac{(\log\chi - \mu_\chi)^2}{2\sigma_\chi^2}}\right) \tag{C.3}$$

C.2. Discretized Log-normal

Following (C.3), the likelihood of $\chi \in [\hat{\chi} - \Delta_\chi, \hat{\chi}]$ given $\hat{\chi} > \Delta_\chi$ is

$$\frac{1}{\sigma_\chi\sqrt{2\pi}}\int_{\hat{\chi}-\Delta_\chi}^{\hat{\chi}} \frac{1}{\chi}e^{-\frac{(\log\chi - \mu_\chi)^2}{2\sigma_\chi^2}} d\chi = \frac{1}{2\Delta_\chi}\left(\mathbf{Erf}\left[\frac{\mu_\chi - \log(\hat{\chi} - \Delta_\chi)}{\sqrt{2}\sigma_\chi}\right] - \mathbf{Erf}\left[\frac{\mu_\chi - \log(\hat{\chi})}{\sqrt{2}\sigma_\chi}\right]\right). \tag{C.4}$$

This can be used to discretize the distribution over χ and the notation

$$\Pr\left(\chi\right) = \overline{\mathit{Log}\text{-}\mathcal{N}}\left(\mu, \sigma, \Delta_\chi\right). \tag{C.5}$$

can be used. Thus, the discrete log-normal can be defined over the interval $(0, \infty)$ by partitioning it as $((i_\chi - 1)\Delta_\chi, i_\chi \Delta_\chi] \, \forall i_\chi \in \{1, 2, \ldots\}$. Note that the likelihood of any $\chi \in \mathbb{R}^{++}$ can be computed using (C.4) by estimating a $\hat{\chi}$ according to

$$\hat{\chi} = \frac{1}{2}\Delta_\chi + \min_{i_\chi} \left\| \chi - \left(i_\chi - \frac{1}{2} \right) \Delta_\chi \right\|_2 .$$

A comparison of the true log-normal and its discrete approximation is presented in Figure C.1.

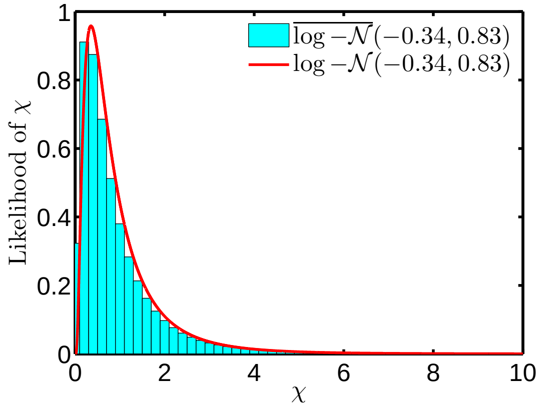


Fig. C.1. Comparing log-normal and discretized log-normal, for $\mathbf{E}[\chi] = 1$ and $\mathbf{Var}[\chi] = 1$.

APPENDIX D

NEGATIVE-BINOMIAL DISTRIBUTION

A negative=binomial distribution is a discrete distribution over non-negative integers. In the native notation, it is parameterized by say, a and b, such that $a > 0$ and $0 < b < 1$. The likelihood of a variable $\varpi \sim \textit{Neg-Bin}\left(a, b\right)$, distribution according to a negative binomial, can be computed as

$$\Pr\left(\varpi | a, b\right) = \frac{\Gamma(\varpi + a)}{\Gamma(\varpi + 1)\Gamma(a)} b^a (1-b)^{\varpi}. \tag{D.1}$$

This is a discrete distribution with mean (μ_{ϖ}), mode (ϑ_{ϖ}) and variance (σ_{ϖ}^2) given as

$$\begin{aligned} \mu_{\varpi} &= \mathbf{E}\left[\varpi\right] = a\frac{1-b}{b} && \text{(D.2a)} \\ \vartheta_{\varpi} &= \underset{\varpi}{\operatorname{argmax}} \Pr\left(\varpi | a, b\right) = \begin{cases} \lfloor \frac{(a-1)(1-b)}{b} \rfloor; & a > 1 \\ 0; & a \leq 1 \end{cases} && \text{(D.2b)} \\ \sigma_{\varpi}^2 &= \mathbf{Var}\left[\varpi\right] = a\frac{1-b}{b^2} && \text{(D.2c)} \end{aligned}$$

respectively. It must be noted that a negative binomial can be alternately defined by any two of the three attributes mentioned above. For example, $\textit{Neg-Bin}\left(\varpi; \mu_{\varpi}, \sigma_{\varpi}^2\right)$ or $\textit{Neg-Bin}\left(\varpi; \vartheta_{\varpi}, \sigma_{\varpi}^2\right)$ are sufficient notations.

It is essential to bring the computational limitations of the likelihood of negative-binomial. The gamma function $\Gamma(\cdot)$ can be extremely large for large arguments. To avoid overflow in a double precision floating point arithmetic, Stirling's approximation [142]

$$\Gamma(n) \approx \exp\left(\frac{1}{2}\log(2\pi n) + n\log(n) - n\right)$$

for large n is used. Also, a zero mean negative-binomial does not exist. As need arises, a zero-mean negative-binomial is approximated by a zero-mode distribution.

APPENDIX E

LEARNING MULTI-DIMENSIONAL GAUSSIAN MIXTURE MODEL

In certain applications the set of data $\mathbb{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$ could be D-dimensional and would be modeled as a multi-dimensional Gaussian mixture model (GMM),

$$\mathcal{G}(\mathbb{Y}|\boldsymbol{\theta}) = \sum_{m=1}^{\tilde{M}} p_m^* \mathcal{N}\left(\boldsymbol{\mu}_m^*, \Sigma_m^*\right), \tag{E.1}$$

where, $\boldsymbol{\mu}_m^*$ is the mean vector and Σ_m^* is the covariance matrix for each mixture component. To model this model in a non-parametric framework, DP could be used as the distribution for the parameter priors, with a base distribution given by a Normal-Wishart distribution,

$$\begin{aligned} P\left(\Theta_m^*\right) &= P\left(\boldsymbol{\mu}_m^*, \Sigma_m^{*-1} \middle| u_\mu, u_\tau, u_\Lambda, u_\nu\right) \\ &= G_0\left(\boldsymbol{\mu}_m^*, \Sigma_m^{*-1} \middle| u_\mu, u_\tau, u_\Lambda, u_\nu\right) \\ &= \mathcal{NW}\left(\boldsymbol{\mu}_m^*, \Sigma_m^{*-1} \middle| u_\mu, u_\tau, u_\Lambda, u_\nu\right), \end{aligned}$$

where $u_\mu, u_\tau, u_\Lambda, u_\nu$ are the hyperparameters of a D-dimensional Normal-Wishart distribution $\mathcal{NW}(\cdots)$. A Normal-Wishart distribution is a conjugate prior for the multivariate normal distribution with unknown mean and covariance [143].

A Normal-Wishart distribution is given by

$$\begin{aligned} \Sigma^*|u_\Lambda, u_\nu &\sim \mathcal{W}\left(u_\Lambda, u_\nu\right) \\ \boldsymbol{\mu}^*|\Sigma^*, u_\mu, u_\tau &\sim \mathcal{N}\left(u_\mu, u_\tau^{-1}\Sigma^*\right), \end{aligned}$$

where $\mathcal{N}(\cdots)$ and $\mathcal{W}(\cdots)$ are D-dimensional normal and Wishart distributions, respectively.

For all the data $\mathbb{Y}_m \subset \mathbb{Y}$ belonging to any cluster m, $m = 1, \ldots, M$, the distribution of data is $P\left(\mathbf{y}_m|\boldsymbol{\mu}_m^*, \Sigma_m^*\right)$. By conjugacy of prior, the posterior distribution of the Θ_m^* conditioned on the data from the same cluster is a Normal-Wishart with different hyperparameters

$$
\begin{aligned}
P\left(\Theta^*_m|\mathbb{Y}_m\right) &= P\left(\boldsymbol{\mu}^*_m, \Sigma^{*-1}_m \middle| \mathbb{Y}_m\right) \\
&\propto P\left(\mathbb{Y}_m|\boldsymbol{\mu}^*_m, \Sigma^*_m\right) P\left(\boldsymbol{\mu}^*_m, \Sigma^{*-1}_m \middle| u_\mu, u_\tau, u_\Lambda, u_\nu\right) \\
&= \mathcal{NW}\left(\boldsymbol{\mu}^*_m, \Sigma^{*-1}_m \middle| \mathfrak{u}_\mu, \mathfrak{u}_\tau, \mathfrak{u}_\Lambda, \mathfrak{u}_\nu\right),
\end{aligned}
$$

where

$$
\begin{aligned}
\mathfrak{u}_\mu &= \frac{u_\mu u_\tau + T_m \hat{\boldsymbol{\mu}}_m}{u_\tau + T_m}, \\
\mathfrak{u}_\tau &= u_\tau + T_m, \\
\mathfrak{u}_\Lambda &= \left[u_\Lambda^{-1} + \hat{\Sigma}_m + \frac{u_\tau T_m}{u_\tau + T_m}(\hat{\boldsymbol{\mu}}_m - u_\mu)(\hat{\boldsymbol{\mu}}_m - u_\mu)'\right]^{-1}, \\
\mathfrak{u}_\nu &= u_\nu + T_m,
\end{aligned}
$$

are the hyperparameters of the posterior Normal-Wishart distribution. Here, $\hat{\boldsymbol{\mu}}_m$ and $\hat{\Sigma}_m$ are the sample mean and sample covariance of the data in $\mathbb{Y}_m$ and $T_m = |\mathbb{Y}_m|$ are the number of elements in the $\mathbb{Y}_m$.

If the available data was a set of observation sequences, it could be modeled with a mixture of hidden Markov models (HMM). In that case, Dirichlet distribution would be the preferred choice for G_0 because HMM mixture and Dirichlet distribution are conjugate priors [144]. More conjugate priors are available on [143, 145].

INDEX