2012-12-12

Advanced Computing Methods for Knowledge Discovery and Prognosis in Acoustic Emission Monitoring

Felipe Mejia
University of Miami, felimz@gmail.com

UNIVERSITY OF MIAMI

ADVANCED COMPUTING METHODS FOR KNOWLEDGE DISCOVERY AND PROGNOSIS IN ACOUSTIC EMISSION MONITORING

By

Felipe Mejia

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

December 2012

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

ADVANCED COMPUTING METHODS FOR KNOWLEDGE DISCOVERY AND PROGNOSIS IN ACOUSTIC EMISSION MONITORING

Felipe Mejia

Approved:

_____
Antonio Nanni, Ph.D.
Professor and Chair
of Civil, Architectural,
and Environmental Engineering

_____
M. Brian Blake, Ph.D.
Dean of the Graduate School

_____
Carol Hays, Ph.D.
Associate Professor
of Civil, Architectural,
and Environmental Engineering

_____
James Giancaspro, Ph.D.
Associate Professor
of Civil, Architectural,
and Environmental Engineering

_____
Mei-Ling Shyu, Ph.D.
Associate Professor
of Electrical and Computer
Engineering

_____
Daniel Berg, Ph.D.
Distinguished Research Professor
of Industrial Engineering

MEJIA, FELIPE (Ph.D., Civil Engineering)

Advanced Computing Methods for Knowledge Discovery and Prognosis in Acoustic Emission Monitoring (December 2012)

Abstract of a dissertation at the University of Miami.

Dissertation supervised by Professor Antonio Nanni.
No. of pages in text. (106)

Structural health monitoring (SHM) has gained significant popularity in the last decade. This growing interest, coupled with new sensing technologies, has resulted in an overwhelming amount of data in need of management and useful interpretation. Acoustic emission (AE) testing has been particularly fraught by the problem of growing data and is made the focus of this dissertation. The dissertation is divided into three studies, the first of which attempts to identify the computing resources necessary for the acquisition, management, and analysis of AE datasets. A computing framework capable of managing AE data is designed and implemented using the methods described in the first study. With a computing framework in place, the second study addresses the problem of unwanted signals in AE; these signals form a large part of most AE datasets and must be removed before a meaningful analysis can be performed. A semi-supervised data mining scheme for detecting and characterizing unwanted AE signals is proposed. The scheme is demonstrated on a synthetic dataset, and applications are presented for pencil lead-break and single-edge tension (SE(T)) datasets. This study suggests that underlying rules can be systematically derived from raw AE datasets. Finally, an artificial neural network (ANN) framework for crack-growth prediction is proposed. The ANN takes AE absolute energy

and crack mouth opening displacement (CMOD) data from two SE(T) specimens as an input in order to forecast future values for each of these parameters. The predicted values are then input into linear-elastic fracture mechanics (LEFM) models in order to estimate the long-term crack evolution. The study concludes that ANNs can adequately model the complex relationships between non-destructive measurement parameters and the crack-length evolution in structural members.

To my grandma,
who gave me everything
and would have given all to read this

ACKNOWLEDGEMENTS

I would like to thank all the people who made this dissertation possible and were instrumental in my growth as a person and student throughout my graduate studies. In particular, I wish to acknowledge:

- Dr. Nanni, my advisor and mentor for the past four years, for his infectious enthusiasm, his enlightening wisdom, his encouragement in good times and tough love in difficult ones, and his inspiration on all of us to become the best we can.
- Dr. Shyu, for her painstaking review of my work and her insistence on proper research practices; for her graciousness in putting up with my difficult habits; and, most importantly, for being the guiding beacon of this work.
- My doctoral committee, for deciding to undertake the responsibility of overseeing my work after having me as their student; for their invaluable input towards a quality dissertation; and for being this work's best possible audience.
- My family, for leaving everything behind so I could get where I am today; for their unconditional understanding and support; and for instilling me with the values that shape my ambitions and are the driving force of my success.
- My partners in crime, Diana, Derek, and Giovanni, for taking the time to know the person who lies beneath and making my time in the office forever memorable.

- Matteo, for being my doppelgänger in this project; for suffering with me the growing pains of research; for reminding me that we must always help the technologically-disadvantaged; and, for being my lifeline when was most needed.
- Zahra, for reminding me each day that purity of character and good intentions are still a part of our generation; for her volunteered hard work and coming to my rescue every time I needed it.
- Mariah, for being a fantastic friend despite taking the brunt of my research; for allowing me to work with who is, perhaps, the most coveted undergraduate in the land; for her patience and dedication to getting this work to the finish line when it appeared impossible.
- Emily and Julie, for blindly volunteering to my cause and being the recipient of my awkward enthusiasm for science and engineering.
- The CAE faculty who have embraced me for a long eight years; and for being teachers and counselors beyond academics.
- Maria and Lizett for their kindness and support throughout my studies; and for being the silent heroes of the department without whom it would not exist.
- The National Institute of Standards and Technology, for funding the entirety of this research; to the fellows at the University of South Carolina, the Virginia Institute of Technology, and MISTRAS, for their joint partnership in this research thrust and for providing valuable input to this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AE	acoustic emission
API	application programming interface
A / D	analog-to-digital
ANN	artificial neural network
BIC	bayesian information criterion
CMOD	crack mouth opening displacement
DAQ	data acquisition
DBMS	database management systems
DICONDE	digital imaging and communication in non-destructive evaluation
ETL	extract, transform, and load
FITS	flexible image transport system
HDF	hierarchical data format
LEFM	linear-elastic fracture mechanics
LoOP	local outlier probabilities
MAPE	mean absolute percentage error
MPI	message passing interface
MRAE	maximum relative absolute error
NetCDF	network common data format
NAS	network-attached storage
NARx	nonlinear autoregressive exogenous

NDE	non-destructive evaluation
PLB	pencil-lead-break
PLOF	probabilistic local outlier factor
RAID	redundant array of inexpensive disks
RC	reinforced concrete
SE(*T*)	single edge (tension)
SIF	stress intensity factor
SHM	structural health monitoring
SQL	structured query language
SSE	sum square error

NOMENCLATURE

a	crack size in SE(T) specimen
α	power constant in U and a LEFM model
b	plate thickness corresponding to the maximum possible SE(T) crack size
c_l	autocovariance at lag l
B	scaling constant in ΔK and $d\eta / dN$ LEFM model
B'	scaling constant in ΔK and dU / dN LEFM model
β	scaling constant in U and a LEFM model
D	scaling constant in dU/dN and da/dN LEFM model
δ	crack mouth opening displacement
$\boldsymbol{e}$	error vector used during LMA ANN training
E	sum square error used as a performance function during LMA ANN training
E'	plane strain modulus of elasticity of steel
f_r	software fraction that is parallelizable
g	power constant in ΔK and $d\eta / dN$ LEFM model
g'	power constant in ΔK and dU/dN LEFM model
$\boldsymbol{I}$	Identity matrix used during LMA ANN training
$\boldsymbol{J}$	Jacobian matrix used during LMA ANN training
k	integer denoting the number of nearest neighbors to data point o
k_p	integer defining the prediction horizon as a multiple of T

K number of clusters in K-means algorithm

ΔK stress intensity factor range

K_{max} maximum stress intensity factor

l number of lags used in autocorrelation calculation

λ significance parameter in probabilistic set distance in *LoOP* analysis

μ combination coefficient used during weight updating in LMA training

n_c total number of logical cores in a computing cluster

N number of loading cycles experienced by SE(T) specimen after t minutes

N_Σ number of neurons in hidden layer of NARx ANN

η AE counts

o data object being considered for outlierness is denoted by o

$O(.)$ algorithm efficiency

p outlier tolerance in *LoOP* analysis

P axial load on SE(T) specimen

ψ scaling parameter relating the crack extension energy and AE abs. energy

q power constant in dU/dN and da/dN LEFM model

r_l autocorrelation at lag l

r^2 coefficient of determination

r_g number of groups of logical cores acting as a single core in a computer cluster

R ratio between the minimum and maximum axial load on SE(T) specimen

S context set around a data point being considered for outlierness

σ nominal tensile stress on SE(T) specimen cross-section due to axial load

t variable denoting the time, in minutes, after crack initiation

t_r trimming percentage factor using in trimmed mean calculation

T ANN prediction resolution

u AE absolute energy of a single AE hit

$\breve{u}$ trimmed mean AE absolute energy of all recorded hits between t and $t+1$

U cumulative AE absolute energy

U_a accumulated energy released by crack extension

$\boldsymbol{W}$ ANN weight matrix used during LMA training

x exogenous variable in NARx ANN model

y autoregressive variable in NARx ANN model

Y set of objects being considered at a decision tree node

CHAPTER 1, INTRODUCTION

A large majority of the bridges constructed in the 1950s and 1960s in the United States are reaching their design end of life. The American Society of Civil Engineers' (ASCE) most recent report card of the national built infrastructure estimates that approximately 26% of all United States bridges are "deficient or functionally obsolete." These statistics should come to no surprise, as bridges are commonly built with a design-life of 50 years, and the average bridge in the United States is now 47 years old (Klotz et al. 2009). Furthermore, in steel structures, more than three-quarters of recorded failures have been related to fatigue-fracture failure modes (ASCE 1982).

Regular visual inspection methods have been federally-mandated, but studies from the Federal Highway Administration (FHWA) show that visual bridge inspections are seldom performed in conjunction with other non-destructive techniques (Moore et al. 2001a). The simplicity and cost of visual inspection makes this type of assessment particularly attractive to bridge owners; however, visual inspection is inherently dependent on the experience of the inspector and, as such, may be subject to significant variability (Moore et al. 2001b). Even though 10% of the visual inspections report crack conditions, there is clearly a growing need for dependable damage detection in civil structures. The structural health monitoring (SHM) community has emerged from these needs, and it has spurred research and development of non-destructive testing and

evaluation (NDE) technologies that will play pivotal role in the health assessment and monitoring of current and new infrastructure.

SHM

SHM has developed to signify the implementation of damage detection and assessment of "fitness for purpose" in aerospace, mechanical, or civil structures. This process generally involves the observation of a structure or mechanical system over time using periodically spaced measurements, the extraction of damage-sensitive features from these measurements, and the analysis of these features to determine the current state of system health (Farrar and Worden 2007). An SHM system may consist of sensors, data acquisition and transmission systems, databases for effective data management, and health diagnosis methodologies (including data processing, data mining, damage detection, model updating, safety evaluation, reliability analysis, and damage prognosis). The number of SHM technologies is continuously growing, but research efforts have mainly been concentrated on the areas of (1) acoustic signals, (2) electromagnetic, (3) radiography, (4) fiber optics, (5) radar and radio frequency, (6) optics, and (7) piezoelectric ceramic (Chang et al. 2003). A more detailed description of SHM can be found in (Worden and Dulieu-Barton 2004). Though several NDE techniques exist, acoustic emission (AE) testing is particularly attractive since it allows analysts to observe the dynamics of material performance in real time.

AE

The process of AE occurs when a material suddenly releases localized stress energy, thereby causing a transient elastic wave that propagates through the material. AE is usually associated with irreversible dynamic processes such as friction, fracture, impacts,

crack growth, corrosion, and other types of damage (Liptai et al. 1972). AE sensing is considered a "passive" method since it does not itself input any energy into the observed system and, as such, allows analysts to test under typical operating conditions (Bray and Stanley 1997). As an SHM technique, AE testing aims to detect, locate, and assess the intensity of damage as it evolves in the presence of applied loads or adverse environmental conditions. The AE technique has been validated using different methods such as guided waves in pitch-catch, pulse-echo, or high-frequency impedance spectrum method in different materials like steel, concrete, aluminum, and fiber reinforce polymers (Raghavan and Cesnik 2007; Wang et al. 2008).

Computing challenges in SHM

SHM remains a research topic that is still making the transition to field demonstrations and subsequent field deployment. A significant challenge facing SHM is the lack of integrated computing technologies that allow for widespread deployment of SHM systems in small-scale laboratory and field settings. This lack of integration arises primarily from the fragmented ecosystem of current SHM hardware/software, which offers solutions related to specific SHM technologies without provisions for integration with other SHM technologies or allowing for open-ended data management and analysis. The computing challenges in AE testing are emblematic of the overall roadblocks found in SHM. More specifically, AE data acquisition (DAQ) systems offer proprietary solutions for data collection and visualization with sensors, hardware, and software being inextricably tied to the vendor companies. Moreover, due to the nature of AE, datasets produced during AE monitoring have been observed to be prohibitively large. Such large datasets are often impractical to analyze, manage, or even visualize using the limited

capabilities of proprietary AE DAQ software. Enabling small-scale implementation of SHM systems such as those using AE while allowing for acquisition, management, and analysis of large datasets would arguably facilitate the adoption of SHM technologies as they enter the mainstream.

Research motivations and objectives

The overarching goal of SHM could be reframed as a condition monitoring system's ability to make automated decisions related to a structure's current and future performance. In this context, the process of SHM is arguably one of statistical pattern recognition, a paradigm which has been discussed in detail by Farrar et al. (2001). This dissertation focuses on AE as a representative SHM technology, and it attempts to advance the practice by introducing modern data mining and machine learning techniques to the analysis of AE data.

Understanding the current limitations of the SHM technology from a computational perspective is essential before undertaking new research; therefore, this dissertation devotes its first study to an overview of the present challenges in SHM. After identifying these challenges, the study outlines a methodology that allows analysts to specify computational requirements for small-scale SHM systems with large datasets in mind. The study concludes with the implementation of a computing framework that allows for the integrated acquisition, management, and analysis of large AE datasets in laboratory and field settings.

With a computing framework in place suitable for the analysis of AE data, the dissertation moves onto the second study, which attempts to address what is possibly the most pressing pattern recognition problem in AE: the presence of unwanted signals (or

"noise") in AE datasets. Removing these types of signals in AE datasets is essential to the meaningful application of the vast majority of AE techniques. The study designs and implements a data mining scheme that enhances the quality of AE datasets. More importantly, the scheme is able to produce characterization rules for both unwanted and meaningful signals. Rule extraction using this technique could lead to the finding of general AE "signatures" to particular damage mechanisms. The third and final study builds on the clean datasets obtained from the second study, and it demonstrates how meaningful information can be obtained from a properly sanitized dataset. Specifically, the study establishes a framework for fatigue-crack growth forecasting using a combination of AE and traditional fatigue-fracture measurements. Through these studies, this dissertation aims to highlight the importance of computation in SHM, while providing a snapshot of the knowledge discovery process required for meaningful interpretation of the ever-growing datasets that form part of current and future SHM applications.

CHAPTER 2, STUDY 1: DETERMINING COMPUTING RESOURCES INVOLVED IN SMALL-SCALE STRUCTURAL HEALTH MONITORING WHEN MANAGING LARGE DATASETS

Summary

Structural health monitoring (SHM) technologies have been growing in popularity, and the increased complexity of SHM systems has resulted in overwhelmingly large datasets. After a brief review of the state-of-the-art and challenges in SHM, this study attempts to outline a methodology for specifying computing resources for managing large SHM datasets and presents an example of a laboratory implementation using acoustic emission (AE) monitoring. The objective of the work is to provide a general understanding of the basic computing resources required to implement a small-scale (i.e., less than 16 sensors) SHM system in the laboratory or in the field. The outlined data management methodology is divided into three parts: data acquisition, data storage, and data analysis. Each part is addressed in a simplified manner with reference to the selected AE example.

Background

Active research in the development of real-time, integrated, and automated SHM data acquisition systems has rapidly accelerated. Over the years, SHM data acquisition systems have become increasingly capable; however, the improved sensitivity, resolution, and processing power of these systems also results in higher data collection rates. While some SHM systems are capable of performing real-time data manipulation *in situ*, the

majority of data collected during an SHM event is stored for subsequent analysis. There is no clear documentation in the literature on how to implement a basic SHM framework that can handle the large datasets produced by state-of-the-art data acquisition systems. This study is therefore intended to offer some guidance, as it presents general recommendations for specifying the computing resources that are typically involved in the management of large datasets arising from SHM in the laboratory and in the field.

An overview of the challenges in SHM and the state-of-the-art

The availability of multiple sensors, coupled with higher resolution, sampling frequencies, and real-time feature extraction by SHM data acquisition systems has enabled SHM researchers and practitioners to efficiently collect much more data in shorter periods of time (Catbas et al. 2008). However, with the increased ability to collect large datasets, there is a growing need to develop systems to store, analyze, and interpret the results rapidly and effectively. Data analysis algorithms, methods, and approaches need to be developed to retrieve and deliver critical information in a timely manner.

The conceptual problem of SHM data management has been recognized for over a decade (Farrar et al. 2001), and the flow of SHM information has been well defined. (Aktan et al. 2000) identified the SHM data problem partly as one of integrating data over complex information systems. Because such data typically exists in disconnected and nonrelational databases, the information systems will have to integrate legacy data with objective field data collected with different types of sensors. For SHM systems that are not linked or integrated, data has to be manually extracted, transferred, and merged. For these systems, data acquisition, presentation, analysis, and archival approaches do not

offer true real-time data acquisition, retrieval, organization, display, or analysis applications because of this lack of interconnectivity. Thus, SHM data management has become a two-fold challenge: that of managing increasingly large amounts of data through all of its stages, and that of merging disparate datasets arising from non-integrated data acquisition systems. Once these two challenges have been met, real-time SHM may become of age. Standards organizations have recently begun to address the issue of data management. In 2004, American Society for Testing and Materials Subcommittee E07.11 published "E2339, Practice for DICONDE," which allows wide-scale adoption of a common standard for data storage and exchange (Howard 2011). Other efforts have been made by the scientific community to provide an easy data interchange among scientists. The HDF, NetCDF, and FITS are good examples of such standards (Gray et al. 2005). While the commercial world has standardized on relational data models such as SQL, no single data storage standard or tool has reached critical mass in the scientific community. Several successful implementations of integrated wired and wireless full-scale SHM systems in have been realized both in the U.S. (Harms et al. 2010; Godinez-Azcuaga et al. 2012) and internationally (Fricker and Vogel 2007; Chae et al. 2012). The success documented in the literature is indicative that progress is indeed being made in the area of SHM, though this study attempts to make the implementation of SHM systems more intelligible to researchers and practitioners with limited access to the technologies involved.

The large majority of data management problems in small SHM applications arise from the collection of small amounts of data, which lead to bad practices that are compounded when collecting larger datasets, such as those obtained from AE systems.

Data is typically collected in portable devices without any provisions for data reliability. This oversight means that data can easily be lost to disk damage, loss, or theft. Large datasets, however, pose problems that reach beyond data management. Attempting to analyze large datasets on spreadsheet software can become problematic as performance can quickly become prohibitively slow as datasets grow. When datasets grow even larger, analysis through conventional means on desktop workstations becomes virtually impossible. Trying to amend data management as an afterthought will usually result in a poorly defined data flow and will result in inefficient data processing. Data management, therefore, should span the whole data lifecycle from collection to obsolescence when data is archived or deleted.

The main objective of this study is to provide a methodology for acquisition, storage, and analysis of large datasets as generated in small scale SHM applications in laboratory or field settings. It addresses the steps involved in basic data management, and it highlights general challenges that are encountered when working with large SHM datasets.

Through this methodology, potential entrants to the area of SHM should have a general understanding of the computing resources required to implement an SHM system. Throughout this study, computing resources will refer to the primary information technology elements that are involved in the management of SHM data. The methodology is divided into three major parts: data acquisition, which highlights the management of data up to the point of persistent storage; data storage, which details the process of readying data for analysis; and, data analysis, which provides guidance on how to specify hardware and software required for analysis. Finally, an example of this

methodology presented to the reader. Specifically, this methodology is applied to the practical use of AE systems, which have been regularly observed to produce large datasets (Beattie 1997; Lei et al. 2004). Due to the analog nature of most SHM sensors, the approach followed for AE systems may be generalized to other SHM systems that rely on waveform sampling as the primary method of data acquisition.

Methodology for data management in small-scale SHM systems

Data acquisition: data collection rates

In order to provide practical upper-bound approximations for required computing resources, the most onerous though still realistic SHM scenario should be envisioned. When choosing a sample test, it is critical to estimate the maximum number of sensors and parametrics that can potentially be used during simultaneous data collection. Any ensuing computing resources will be chosen to accommodate maximum data collection rates and, thus, will also be useful when estimating data transmission and persistent storage solutions.

A correct estimate of both the rate of data collection and the storage capacity for all datasets is the first step in SHM data management. Approximating the rate of data collection is useful when determining whether special transmission links are needed between the SHM data acquisition box and an offsite storage server due to exceedingly high throughputs. If data acquisition rates surpass the rates of data transmission, data can be bottlenecked locally, filling the data acquisition system's hard drive to capacity and losing important data.

The rate of data collection is a function of several factors that are specific to each SHM setup. There is no universal expression for estimating actual data acquisition rates,

but approximations can be made based on each individual setup. A theoretical rate ceiling can be estimated based simply on A/D converter specifications. If all channels share the same A/D converter, then the maximum data collection rate (in bits) can be obtained by multiplying the A/D converter's bitrate by its sampling rate. Maximum throughout speeds are virtually never achieved, so basing computing specifications on this number would be excessive.

A better approximation of data collection rates should be based on test conditions and the data structure of collected data. In order to perfectly predict the rate of data collection, it is necessary to know the exact data structure, so that the size of each sample is also exactly known. If the exact data structure cannot be known, each sample's size can be approximated based on the number of features gathered by the data acquisition software, plus the collection of all waveforms and parametrics corresponding to each hit—times the size in memory or disk of each of these types of data.

Transfer to persistent storage

Before putting an SHM system in place, how data will be transferred to its permanent storage location (if any) must be specified. Oftentimes, manual transfer of files from the field to a repository is the most convenient form of data transfer. There are three clear disadvantages to the manual transfer of data from a local drive in an SHM system to a storage or analysis station. First, when relying on manual transfer of data, the automation process is temporarily interrupted, leading to lack of automation and inefficiencies; second, portable storage devices rarely provide data redundancy, which makes the data vulnerable to corruption while it resides on an intermediate device; and third, the structure's owner or SHM system manager risks filling local SHM system's drive to

capacity, with the possible loss of important data. SHM data should thus be transferred via wired or wireless services, and the process should be periodic or constant, and it should be automated. When data is transferred through a network, the local SHM system's drive should be used as a buffer so that data is not lost if the communication link inadvertently goes down.

The upload bandwidth of the wired or wireless system used to transfer the data may become an issue if data is collected at a sufficiently high rate. Practical limits of data transfer technologies are well documented. For example, common copper-wired 100BASE-TX Fast Ethernet provides an upload of 100Mbit/sec, and wireless 3G WCDMA networks provide a practical upload speed of approximately 64Kbit/sec (Kara et al. 2005). Data can be transmitted using standard network protocols such as FTP or any other TCP/IP or UDP-based service. If data is to be continuously streamed from the field, the maximum sustained upload speed should be greater than the rate of data collection. This requirement is especially important when choosing wireless technologies, which tend to have slower bandwidth allowances than their wired counterparts.

Data storage: choosing a storage device

Choosing a suitable data storage solution is an often-overlooked step of the data management process. Storage, at its core, is comprised of physical media such as hard drives, compact discs, or magnetic tapes. Persistent storage should be allocated in order to make data available to analysts for the duration of an experiment, project, or until it becomes obsolete. The data has to be readily available to the analyst, as large datasets may run the risk of being collected at a faster pace than they are analyzed. This practice would create data repositories that, in all likelihood, will never be accessed again. Thus,

stored SHM data should remain directly accessible to the analyst until it has been processed. After analysis, data may be archived or backed up to magnetic tapes so that it is retrievable when necessary. Estimating the amount of storage required for an SHM project can be difficult, due to the need of balancing the cost, capacity, and performance of the storage solution. For this reason, one of these criteria is often sacrificed at the expense of the other two more requirements.

Determining when data will become obsolete is a subjective task, so it is best to keep it in a readily-accessible form of storage throughout the life of the structure being studied. In research settings, data should arguably be readily-accessible at all times during a project. The necessary storage capacity can be estimated by adding up the anticipated runtime of each data acquisition system multiplied by the average data collection rate of each device. Note that—following acquisition—any duplication, cleansing, consolidation, compression, or transformation of or within the data structure may result in different storage needs. If raw data is to be loaded into a database management system, then considerations must be made for database overhead, potential data compression, and changes in the database structure.

As should always be the case with physical storage, data redundancy should be taken into account since a single drive failure in a non-redundant system will result in loss of data. The most common data redundancy solution is that of a redundant array of inexpensive disks (RAID). An array of disks in a RAID configuration can be arranged in a variety of different storage schemes or "levels." Storage schemes differ primarily in whether they are striped (logically-segmented across multiple drives), striped with parity (logically-segmented across multiple drives with fault tolerance), or mirrored (replicated

across separate drives). An introduction to RAID and a description of several types of RAID levels and their performance can be found in (Patterson et al. 1989) and (Chen et al. 1994). In general terms, because data mirroring can significantly decrease the effective storage space of the RAID, striping with parity is often favored.

The last few years have seen a dramatic increase in microprocessor performance, and computer system performance is doubling every 18–24 months—a phenomenon often identified as Moore's Law (Moore 1965). On the other hand, the rate of increase in storage access speed is much lower than processor performance rate. Improvement in disk access times typically faces mechanical constraints, so it has been seen to grow at a rate of less than 10% per year. The performance of storage systems is becoming a major bottleneck in computing system performance, and it will limit the speedup of sequential as well as parallel systems, as implied by the Amdahl's law: ''speedup is limited by the slowest system component'' (Amdahl 1967). In parallel systems (such as clusters), for example, it is necessary to improve I/O performance to balance increasing processor performance. Hence, providing large storage capacity with high access speed is now a critical issue to be considered in the design of computer systems. RAID levels, in addition to providing redundancy, may also increase I/O performance through the striping of data, so it is important that the storage solution chosen for an SHM system also benefits from this speedup (Cannataro et al. 2002).

Data integration and database storage

Bridge structures are often outfit with multiple data acquisition systems (Ko and Ni 2005), which in a sense act as separate data sources that are then networked together. Related data that is acquired concurrently by multiple sources must be integrated into a

unified dataset or database. Data integration has been widely studied, and several conceptual models exist (Vassiliadis et al. 2005). The process of readying data for database storage is termed *extract, transform, and load* (ETL). The first part of the ETL process is to extract data from all the separate sources. During the *extraction* process, data may have different structures, semantics, and file formats. Data sources may be flat files or other databases. The *transform* stage requires a restructuring of data in order to meet analysis needs; this stage may include joining of feature sets, sorting, and data cleansing. The *load* phase loads data into the end target, which is usually a database. Small-scale SHM systems typically save a series of data files to disk and do not interact directly with databases. Therefore, data must undergo an ETL process before it can analyzed by the user. After data has undergone this process, all queries will be able to be performed on a unified relational database, and the leftover raw data will be ready for archiving (Figure 1a).

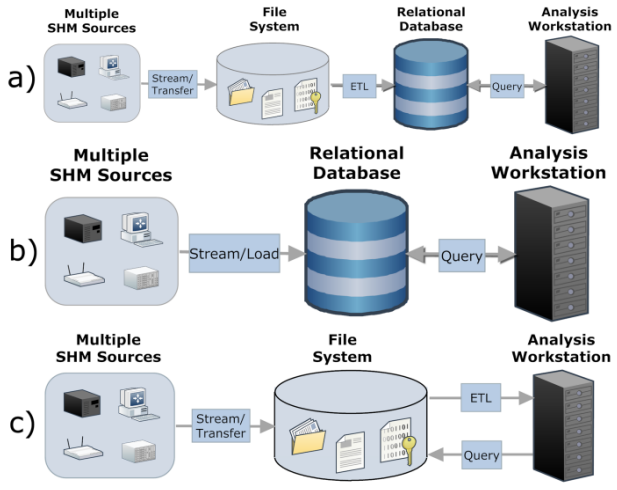

Figure 1 – Data storage models: (a) conventional DBMS with intermediate file system; (b) conventional DBMS without intermediate file system (c) data storage without a DBMS

The ETL process may be performed manually, or it can be carried out by specialized ETL software or some database management systems (DBMS). Data extraction requires knowledge of the source's data structure. ETL or DBMS applications may be capable of extracting data from known database and flat file formats. In the case of most SHM systems, data structures and data formats tend to be proprietary and not open-source. If data structures are not readily-available via SHM manufacturers' user manuals, the ETL process may prove difficult. Reverse engineering methodologies exist for understanding data-structures and can be performed on proprietary data files, though it may require significant effort (Wong et al. 1995). Once the data structure is known, ETL and DBMS software can be programmed to parse the files or databases in order to extract all necessary data. Following the ETL process, raw data may be kept, or it may be discarded if no intermediate data storage is required.

Alternatively, if the SHM project is of small scale, raw data may be left in its original format, and ETL processes can be carried out by analysis software, effectively bypassing the need for a database (Figure 1b). It must be noted that traditional DMBS software has lagged in supporting core scientific data types. Some researchers use databases for some of their work but, as a general rule, most scientists do not. Databases have been slow in adoption for several reasons including: a high learning curve; researcher and practitioner's ability to manually manage small datasets; lack of support for certain data types (arrays, spatial, temporal, etc.); slow performance; an inability to access databases by some analysis software; and costs of setup and maintenance. All of these reasons are based on experience and considerable investment, though these limitations are only still true of older systems. While modern databases have considerably improved, the research

community has been slow to migrate (Gray et al. 2005). Still, the file-FTP *modus operandi* will not work efficiently on larger datasets so, despite the lack of database use in SHM systems, the industry will have to move towards non-procedural, indexed data management systems. Ideally, data flowing from SHM sites should be directly streamed and loaded into a database as it is acquired, rendering storage of individual files unnecessary (Figure 1c). However, this functionality will not be possible until the SHM industry recognizes database integration as an essential part of data acquisition software.

Data analysis

There exist a plethora of analysis algorithms for SHM data; a summary of the most common algorithms employed in SHM is given by (Doebling et al. 1996). Due to the limited processing power available in SHM systems, the majority of data analysis must be performed after acquisition and is typically outsourced to desktop workstations with greater memory and processing power than that of the SHM data acquisition systems. In order for the data to be analyzed in a reasonable amount of time, the specifications of the workstation doing the analysis must be carefully selected. It is possible that the analysis workstations will have to be clustered so that—by taking advantage of the parallelism—data is able to be analyzed in an efficient manner.

Specifying a computing platform depends not only on the desired ability for the computer to perform a task, but on the speed at which it performs it. Before acquiring dedicated analysis hardware, the algorithms that will be applied to the data must be identified in order to better specify memory and CPU requirements. Analysis algorithms can take data as an argument and produce results based on the data that is passed to them. As the amount of data grows, the time it takes for each algorithm to complete the analysis

may change; similarly, the amount of memory utilized can vary as the quantity of data becomes larger. There are theoretical methods for analysis of algorithms such as limiting behavior analysis, which describes the asymptotic behavior of functions as inputs become very large. In this type of analysis, algorithms behavior is represented in the form

$$O(g(n)) \text{ as } n \to \infty . \tag{1}$$

For example, an algorithm that grows in a linear fashion will have the form $O(n)$, so doubling the input will, in theory, double the memory usage or runtime of the algorithm. An algorithm with $O(n^2)$ behavior will use four times more memory or take four times longer after doubling the input size. There is a considerable body of research in computer science dealing with limiting behavior of functions, where the theoretical asymptotic behavior of algorithms is studied. Theoretical analysis of SHM algorithms is only of marginal interest in small SHM applications, though it becomes important when scaling the algorithm to larger datasets. Further information on limiting behavior analysis can be found in (Knuth 1968, Bruijn 2010).

A more practical approach to algorithm analysis is that of performance analysis—also called "profiling." Profiling is achieved by instrumenting either the program source code or its executable using a tool called a *profiler*. A profiler may measure the usage of memory, the usage of particular instructions, or the frequency and duration of function calls. A profiler can be implemented on existing source code through augmentation of the code to include benchmark analysis tools (Srivastava and Eustace 1994). Third party profilers exist and are available in standalone or as part of compiler and interpreter software, such as *Microsoft® Visual Studio* and *MATLAB®*. Profiler tools will measure where a program spends most of its time and will provide a breakdown of allocated and

freed memory during execution. The two main drawbacks of utilizing profiler tools as predictors of algorithm behavior are that algorithm source code must be first be implemented in order to be tested, and that profiler benchmarks do not guarantee that algorithm behavior of larger input sizes will grow as predicted. Thus, algorithms should be thoroughly profiled on prototype datasets of varying sizes in order to better predict full-scale behavior.

Software specification

Computer architectures have changed in order to support the advent of more powerful hardware. The limitations of aging computer architectures become especially problematic in high-performance computing, where very large storage and RAM requirements must be supported by operating systems and other software. In recent years, 64-bit architectures have become more popular, as they allow for memory limits several orders of magnitude larger than those available in aging 32-bit systems, where maximum addressable memory is limited to 4GB. Most data acquisition products available commercially interface directly with 32-bit OS's, which may limit the performance of the overall SHM system. If possible, systems with 64-bit architectures should be specified in order to avoid memory and storage limits.

Hardware Specification

Once the algorithms to be used have been identified, hardware must be specified to allow for data analysis within practical time limits. Due to I/O inefficiencies associated with saving temporary data to the hard disk during analysis, it is generally preferred to have algorithms performed entirely using processor cache and available RAM. If possible, hard disk use should be limited to the storage of algorithm output, and disk

caching should be kept to a minimum. Ideally, basic data transformations should be able to be performed entirely within the CPU or RAM. In the event that some analysis algorithms require more memory than is available, temporary data will have to be saved to disk at the expense of algorithmic efficiency.

Computational runtime may also be monitored, especially when the duration of computation exceeds practical limits. In other words, how "soon" results are wanted can become the limiting factor when choosing an analysis system. In order to gain computational efficiency, algorithms can be parallelized across different cores, CPUs, or workstations. APIs such as *OpenMP* allow for multithreading, which is a form of shared memory multiprocessing; in this implementation, a master "thread" (a series of instructions executed consecutively) "forks" a specified number of slave "threads" and a task is divided among them. The threads then run concurrently, with the runtime environment allocating threads to different processors. Message passing interfaces (MPI) such as *OpenMPI* allow for processes to communicate with one another by sending and receiving messages; this protocol has become a *de facto* standard for communication among processes that model a parallel program running on a distributed memory system. Computer clusters—groups of computers linked by a communication interconnect (such as Fast Ethernet or fiber optic)—being the most common high-performance computing solution, often utilize a hybrid model of *OpenMP* and *OpenMPI* in order to take advantage of the shared memory parallelism within each compute node while allowing intranode communication. When handling large datasets, the analysis software and host operating system should take full advantage of the processor architecture.

The maximum theoretical speedup allowed by a parallel system is dictated by Amdahl's Law (Amdahl 1967). The modern version of Amdahl's law states that the computational speedup of a symmetric (identical nodes) computational system is given by:

$$Speedup(f_r, n_c, r_g) = \frac{1}{\frac{1-f_r}{perf(r_g)} + \frac{f_r \cdot r_g}{perf(r_g) \cdot n_c}}, \quad (2)$$

where f_r is the software fraction that is parallelizable, n_c is the total number of logical cores, and r_g is the number of groups of logical cores acting as a single core with reduced performance $perf(r_g)$ (typically taken to be equal to $\sqrt{r_g}$) (Hill and Marty 2008). In the case of computer clusters, it can be assumed that, in a hybrid *OpenMP/OpenMPI* mode, each node acts as a single compute core with performance $perf(r)$, where the total number of cores n_c is equal to r_g times the number of compute nodes. The specifications of the required computational system can be tweaked in order to reach the speedup needed to perform a typical analysis within the desired timeframe.

Methodology applied to AE in small laboratory and field settings

Data acquisition

The methodology described previously was employed during the implementation of laboratory and field SHM systems using AE. The most taxing operating conditions during the testing process were envisioned based on the number of sensors and parametrics.

The laboratory setup is the most demanding and consists of four reinforced concrete (RC) specimens subject to accelerated corrosion (Figure 2).

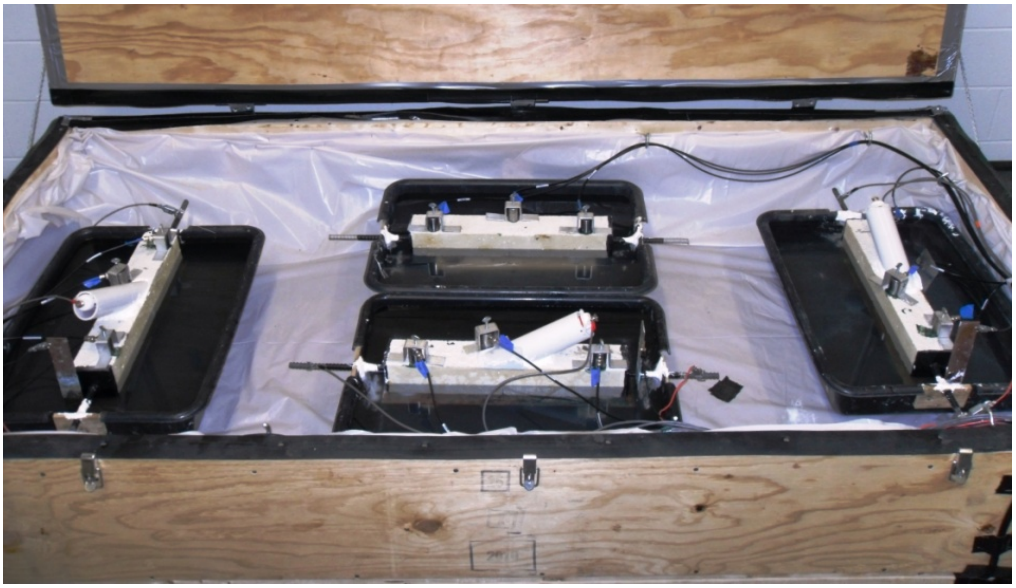

Figure 2 – AE monitoring of RC specimens subjected to accelerated corrosion

AE sensors are attached to eight *Physical Acoustics Corporation*® (*PAC*®) *PCI-2 Rev. 3* AE DAQ boards (18-bit, 40MHz A/D converter, 4x sample averaging, 10MSPS). Parametrics are connected to the primary *PCI-2* onboard parametric input (16-bit, 100Hz A/D converter, 100SPS per parametric input). In AE, sensor activity can be measured in terms of hits per second .Typical AE activity produces an average of 5hits per second per channel and a maximum of 50hits per second per channel. These hit rates are obtained by preliminary testing in the laboratory and confirmed in the literature. Parametrics are sampled at a rate of 100 samples per second per input (i.e., the maximum sampling rate allowed by the *PCI-2* board). Waveforms corresponding to each hit are sampled at a rate of 40MHz and are stored as a fixed size vector of 2048 coordinates per waveform. Since the system is fully utilized, subsequent calculations are based on a SHM setup that is collecting data using all 16 AE sensors, plus 2 parametric inputs (Figure 3).

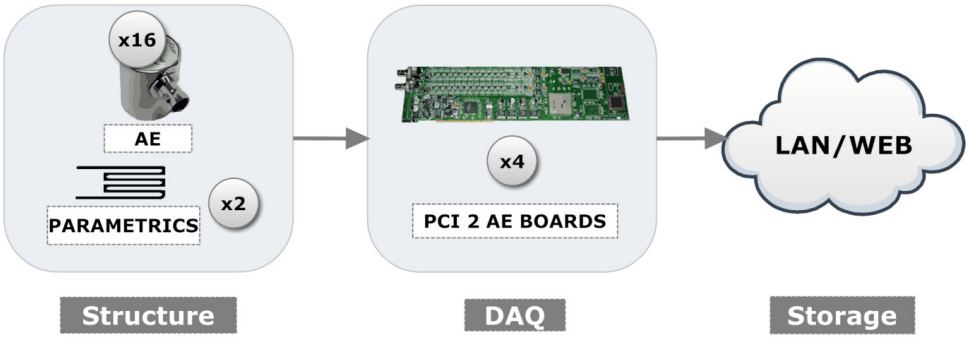

Figure 3 – Sample SHM test setup (AE equipment figures provided by PAC®)

The theoretical upper-bound data collection rate can be found by multiplying the A/D converter's bitrate by its sampling rate. In the case of the *PAC® PCI-2* AE board, the A/D can output a stream of AE data at 10MHz with a depth of 18-bits, for a maximum data collection rate of 10.73MB/s per channel. In a similar fashion, the maximum data collection rate for each parametric input can be calculated to be equal to 200 Bytes/s per channel. For all 16 channels and two parametrics, the system will collect data at a rate of 171.66MB/s. Knowing that no data is compressed or encrypted following data collection due to the limited processing power of field SHM systems, and assuming sufficient I/O headroom, then this speed can be considered to be the maximum cumulative data collection rate to be output by the A/D converters.

Certain SHM technologies, where data collection rates are proportional to the threshold sensitivity of the sensors, tend to quantify sensor activity as a hit rate. Naturally, datasets where high hit rates are observed will be inevitably larger. Additionally, any assumed data collection rates should be comfortably greater than those expected from operating conditions in order to account for hit-rate uncertainty and data transmission protocol overheads.

Average hit rate can be translated into a data rate by estimating a size on disk for each of the recorded features and parametrics. (Doebling et al. 1996) detail some of the different types of data recorded by a typical modern AE system. Specifically, the *PCI-2* board can output up to 20 features per hit sample and 7 features per time sample, in addition to waveforms and parametrics (Table 1).

Table 1 – Types of recorded AE features

Hit Data Items			**Time Data Items**	
• Time of Test • Channel • Amplitude • Energy • Counts • Duration • RMS	• ASL • Threshold • Rise Time • Counts to Peak • Average Frequency • Reverberation Frequency • Initiation Frequency • Waveforms	• Signal Strength • Absolute Energy • Cycle Counter • Partial Powers • Frequency Centroid • Peak Frequency • Parametrics	• Time of Test • Cycle Counter • Parametrics	**Per-Channel Items** • RMS • ASL • Threshold • Absolute Energy
			Conditional Data	
			• Alarm Data	• Time Mark Data

If it is assumed that all of the features computed from each sample are stored into the data file as floating point numbers (a reasonable upper-bound approximation), then each feature can be considered to have a stored length of 4 bytes. For AE boards that accept non-AE inputs, each parametric must also be accounted for, so it is also reasonable to conservatively assume that each collected parametric has a size equal to the bitrate (2 bytes in this case). For hit-based data, all parametrics are sampled once per hit; time-based parametrics, on the other hand, are sampled at regular intervals. It can be assumed that all parametric samples are also stored as a floating point numbers. Lastly, if waveforms are to be collected then (for a 2048-coordinate waveform, at an 18-bit depth) 4,608 bytes must be allocated for each waveform. By adding the contribution of each of the features and parametrics, it is possible to arrive at an estimate for the data rate of an AE system in terms of the hit rate. These types of data acquisition rates are typically

reached during high-sensitivity, high-emissivity tests and serve as a conservative upper bound.

The sample SHM system, at a hit rate of 50 hits per second per channel, would output data at a maximum rate of 3.6MB/s. It can be readily seen that waveform collection produces the greatest impact regarding data collection rates, as the same setup without waveform collection would output data at only 97.3KB/s. Maximum data collection rates are important when specifying transfer and I/O device performance. Average hit rate, though, is more closely associated with required persistent storage. In the sample setup, at a hit-rate of 5 hits per second per channel, data will be collected (including waveforms) at an average rate of 398.2KB/s.

Data collection at a rate of 398.2KB/s makes it impossible to have it transferred via a wireless 3G network, so a wired system is required. In case that a wireless connection is essential, such as in remote field locations, data collection sensitivity or data sampling rates must be decreased. If bandwidth limitations become an issue, then data can be compressed prior to transmission, or some data (such as waveforms) can be stored locally rather than transmitted. Data compression is typically not possible at the time of collection as it would require significant additional processing capability, which is relatively limited in SHM systems due to stringent power requirements (Hill and Marty 2008). Thus, most systems transmit raw data and leave any data manipulation to dedicated workstations. SHM data rates in the laboratory have been proven to be comparable to the rough estimates above; therefore, a high-bandwidth network is required. To avoid any transmission bottlenecks, all research-purposed SHM systems are connected to a local network rated at 100Mbit/sec or greater. The local SHM system's

storage drive is used as a buffer in order to prevent data loss in the case of temporary network outages. The SHM system is configured to periodically push files of 1GB through the network after they have been collected.

Data storage

The storage capacity required to store all of the data acquired throughout the life of the laboratory study is approximated by adding up all of the SHM runtime and multiplying it by the average acquisition rate. In the referenced RC corrosion study, it is expected that two identical SHM tests (each consisting of 4 specimens) will be performed concurrently for 5 days/month for a project duration of 72 months. Consequently, at the average acquisition rate of 398.2KB/s, an effective storage capacity of 23.07TB is required to accommodate all the collected data. A 16-disk, 32TB network-attached storage (NAS) server was acquired and made available to all acquisition systems and analysis workstations with the purpose of storing all data for the lifetime of the project. Because of the importance of the acquired data, a double parity array (Figure 4) was specified, which allows for up to two drives to fail at any given time without any loss of data. In order to further increase the reliability of the NAS, a standby hot spare was added to the RAID volume. Since RAID 6 requires two drives for parity, and the hot spare does not form part of the RAID volume, then three of the sixteen drives must be allocated for redundancy. This configuration limits the 32TB NAS to only 26TB of effective storage, but it still allows for all of the data to be collected. Because physical drives are typically specified as having 1000MB/GB of capacity instead of the conventional 1024MB/GB, then a 26TB volume will be reported by the operating system as having a capacity of 24.8TB. All data collection estimates should be similarly expressed in this fashion in

order to avoid a storage miscalculation. The specified NAS was benchmarked locally and it showed a 300MB/s-write and 360MB/s-read speed performance, which exceeds the throughput of the three bound 1000BASE-T LAN ports used to connect to it, limiting data I/O performance from other computers connecting to the NAS to the bandwidth of the network interconnect. Since most 32-bit commercial OSs have a limit of 16TB per storage volume, an industry standard 64-bit RedHat® Linux OS was specified for the NAS.

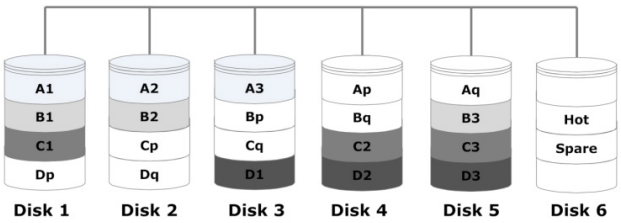

Figure 4 – RAID 6 + spare storage diagram

After a storage solution was chosen and the OS specified, a data storage model has to be adopted. Because the SHM setup is of a relatively small scale, it was determined that a database-less model was more efficient for this project (Figure 1b). This model requires data files to be parsed by analysis software such as *MATLAB*® or *Wolfram*® *Mathematica*. In this case, *MATLAB*® was chosen as the only data management tool. Prior to data analysis, all SHM files are parsed by custom MEX libraries, which perform ETL operations on the data and save it directly to RAM into temporary n-dimensional *struct* arrays with a snowflake logical schema (Figure 5).

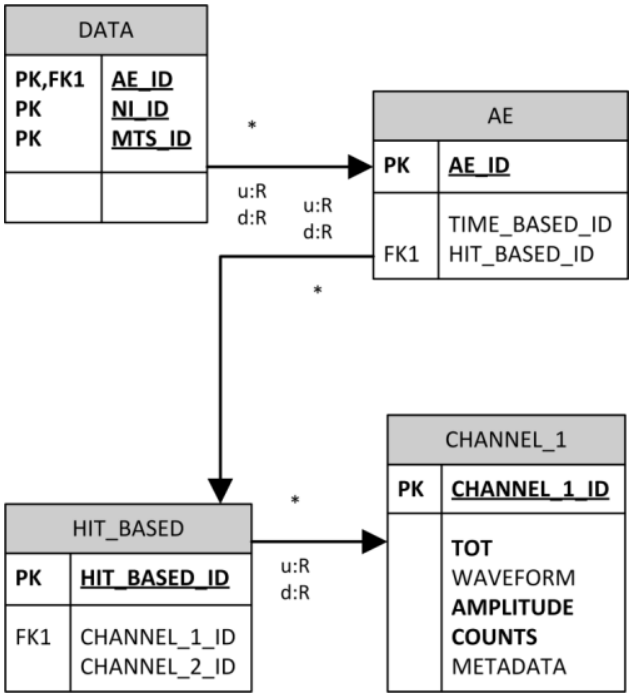

Figure 5 – Snowflake database model diagram of sample SHM dataset

This procedure is done only once per analysis and, even though it does not take advantage of the querying speed and efficiency of an integrated database, it allows the data analyst to investigate SHM data files without the need to implement a DBMS or learn the application programming interfaces (API) associated with database querying by third party software.

Data analysis

When trying to specify hardware to be used in a data analysis workstation, a representative data analysis algorithm is recommended to be used for profiling purposes. A K-means clustering algorithm with $O(n)$ time complexity, as applied to AE data by (Manson et al. 2001), was selected. (Figure 6) shows a normalized runtime profile of a k-means algorithm as input datasets are doubled. From this figure, it is possible to estimate typically required computing resources involved in data visualization by extrapolating

algorithm behavior to larger data sizes. In (Figure 6), the first k-means benchmark of a 1,419KB dataset took 0.98seconds (as measured by the profiler). A file 128 times larger (177.38MB) took 27.5 times longer (26.95 seconds). The polynomial regression of this data would predict that a K-means analysis of a 164.05GB (the average aggregate file size for one test, corresponding to a 5-day duration at a mean data acquisition rate of 398.2KB/s) dataset would require over 70 days to run on a single 2.4Ghz CPU core. The average size of the data being analyzed should always be smaller than the available addressable memory. This criterion suggests that the analysis workstation requires an aggregate RAM size greater than the amount of data being analyzed at one time. Accordingly, a system with 192GB of RAM was specified in order to, as a bare minimum, allow for the parsing, memory allocation, and basic analysis of an entire set of files for one test.

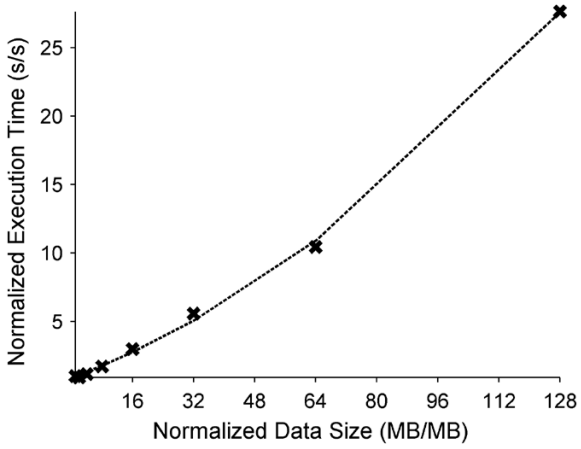

Figure 6 – Profiler benchmark of K-means clustering algorithm

Lastly, in order to reduce the time of computation from the estimated 70 days to a more manageable time-frame of 72-hours, a speedup of 23.3X is required. Assuming that the typical parallel implementation of a K-means algorithm parallelizes approximately

96% of the execution and that each compute node has two eight-core CPUs ($r_g = 16$), then the modified Amdahl's law can be used to find the number of cores that will produce this speedup. Solving for n_c yields 117 required cores, with 8 compute nodes ($n_c = 128$) being the smallest possible computer cluster configuration that would be commercially available.

Once in place, the computer cluster's speedup was tested using a profiler. In order to test theoretical speedups versus measured speedups, a K-means algorithm with approximately 96% parallelizable execution ($f_r = .96$) was run by one compute node in an embarrassingly parallel mode ($r_g = 1$ and $n_c = 16$, and a theoretical speedup of 10X). The measured speedup when utilizing all 16 cores in one node, as shown in (Figure 7), was found to be 9.4X. As expected, the observed speedup is slower than the theoretical speedup, a phenomenon that can be attritubed to communication latency inherent to the interconnect between each core or CPU.

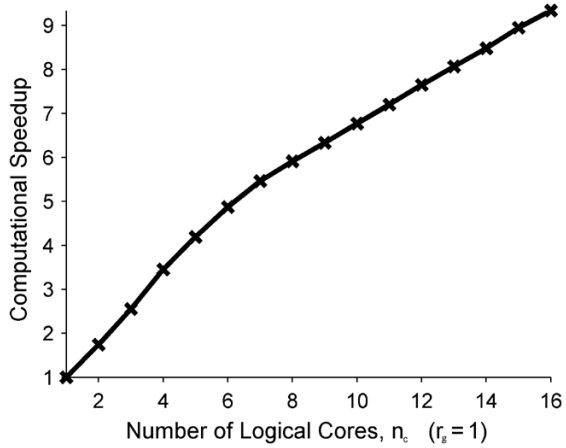

Figure 7 – Observed computational speedup of K-means algorithm

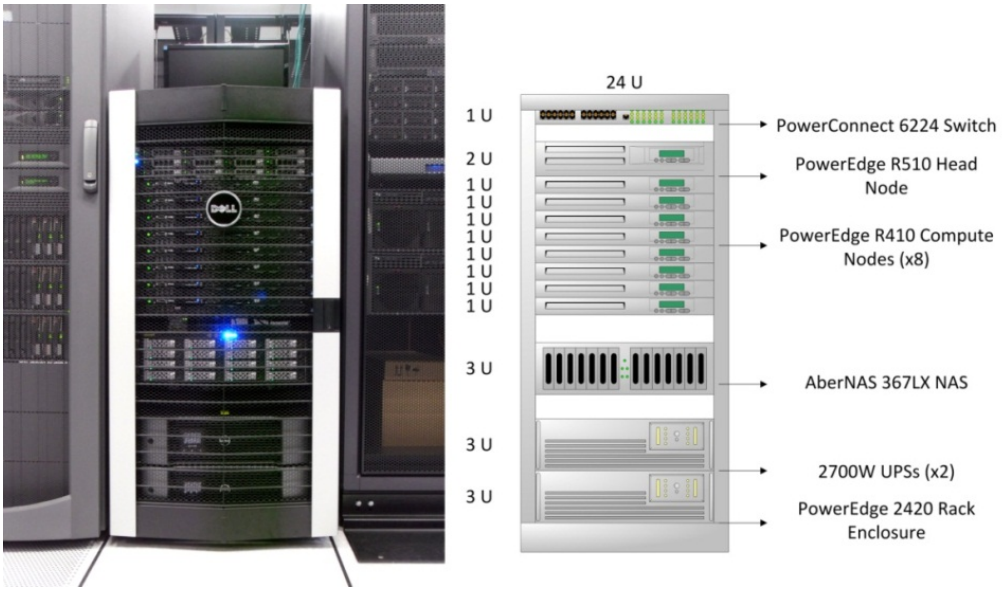

Figure 8 – Computer cluster diagram of implemented high-performance computing SHM solution

(Figure 8) provides a diagram and illustration of the final hardware configuration for the computer cluster used in analysis. To keep consistency across all of the elements in the computer cluster, a RedHat® Linux OS was also specified for all of the nodes. For analysis and computation, an instance of *MATLAB*® (also 64-bit) was chosen as the primary software solution due to its cross-platform support, multithreaded and distributed computing capabilities, and ubiquity in the scientific community.

Field SHM testing

The same SHM setup shown in (Figure 3) was adopted to monitor the reinforced concrete (RC) pier cap of a bridge slated for repair (Figure 9).

Figure 9 – Field SHM set-up of RC pier cap slated for repair

Since a wireless 3G modem was used to transfer the collected data, the AE threshold was adjusted to ensure that the data collection rate never exceeded the available bandwidth. A rate of six hits per minute per channel was found to provide adequate headroom when continuously transmitting data to the remote server. Evidently, this rate of data collection produces significantly smaller datasets. For example, continuous monitoring of this pier-cap for five years would require less than 1.2TB of storage, and analysis could be performed using a typical quad-core workstation.

Concluding remarks

The growth of SHM has led to an increased complexity of acquired datasets. The ability for engineers and researchers to be able to analyze the mounting quantities of SHM data has become a crucial challenge. The information technology demands of SHM systems have forced engineers to increasingly engage in interdisciplinary work. For this reason, high-performance computing has become a practical solution for the management and analysis of SHM datasets. Therefore, in order to allow for the continued growth of SHM—and as datasets grow even larger—these practices must become an integral part of the SHM process.

A methodology was presented that enables entrants to the field of SHM to determine required computing resources in their small-scale SHM projects. While this procedure is basic and not meant to be exhaustive, it provides a good foundation for researchers and practitioners interested in deploying larger SHM systems. A sample setup consisting of 16 AE sensors and two parametric inputs was presented with the purpose of specifying hardware and software resources required for managing datasets arising in laboratory experiments dealing with SHM of RC specimens subjected to accelerated steel corrosion. Similarly, field tests were conducted on a pier cap slated for repair.

Based on the estimated requirements, a computer cluster with 8 compute nodes—each with 16 cores and 24GB of RAM—was specified. The aggregate RAM is equal to 192GB, and the number of logical cores is equal to 128, which matches or exceeds the specifications required for performing analysis of datasets of the estimated magnitude. In summary, acquired data flows from SHM data acquisition systems to a centralized data repository via Fast Ethernet. Once there, the data to be analyzed undergoes an ETL process at the time of analysis, and it is stored in RAM until analysis has finished.

CHAPTER 3, STUDY 2: DATA QUALITY ENHANCEMENT AND KNOWLEDGE DISCOVERY FROM RELEVANT SIGNALS IN ACOUSTIC EMISSION

Summary

The increasing popularity of structural health monitoring (SHM) has brought with it a growing need for automated data management and data analysis tools. Of great importance are filters that can systematically detect unwanted signals in acoustic emission (AE) datasets. This study presents a semi-supervised data mining scheme that detects data belonging to unfamiliar distributions. This type of outlier detection scheme is useful in detecting the presence of new AE sources, given a training dataset of unwanted signals. In addition to classifying new observations (herein referred to as "outliers") within a dataset, the scheme generates a decision tree that classifies sub-clusters within the outlier context set. The obtained tree can be interpreted as a series of characterization rules for newly-observed data, and they can potentially describe the basic structure of different modes within the outlier distribution. The data mining scheme is first validated on a synthetic dataset, and an attempt is made to confirm the algorithms' ability to discriminate outlier AE sources from a controlled pencil-lead-break (PLB) experiment. Finally, the scheme is applied to data from two fatigue crack-growth steel specimens, where it is shown that extracted rules can adequately describe crack-growth related AE sources while filtering out background "noise." Results show promising performance in

filter generation, thereby allowing analysts to extract, characterize, and focus only on meaningful signals.

Background

Parametric analysis in AE

Parametric analysis of AE in metals is not a new concept. Initial analyses of AE were based on emission rates, or "hit count" (Schofield 1972). Shortly after, as more powerful circuitry became available in AE systems, basic waveform parameters such as amplitude were able to be computed. The potential of amplitude analysis as a viable characterization method of emission signals was recognized in (Pollock 1973) and became the dominant practice for several years. More recently, with the advent of waveform recording capabilities, feature extraction (e.g., frequency, time-frequency, and wavelet transforms) in post-processing has been used extensively (Ni and Iwamoto 2000, Marec et al. 2008, Khamedi et al. 2010, Dijck and Hulle 2011).

Pattern recognition, machine learning, and data mining in AE

Farrar et al. (2001) described SHM primarily as a problem of pattern recognition, and the AE research community has since adopted this paradigm. Pattern recognition has been used in AE mainly as a technique for characterizing the structure and natural "signatures" in AE datasets (Marec 2008; Gutkin et al. 2011; Li et al. 2012; Sause et al. 2012).

More recently, pattern recognition in AE has evolved towards machine learning and data mining, which have been used mainly for classification, regression, and prediction. Current implementations are implementing data mining tools for hypothesis searching, rule extraction, and decision-making. For example, artificial neural networks have been

used to classify and predict background "noise" in aerospace composites (Bhat et al. 2003); waveform classification in fiber reinforced polymer monitoring (Olivera and Marques 2008); and crack-related signal characterization in aluminum specimens (Qian et al. 2009). Finally, data mining as a rule extraction technique for classification of various AE signals has been used by (Omkar and Karanth 2008).

Proposed data mining scheme for unwanted signal detection and characterization

Because data mining techniques allow for the classification of statistically similar data, a natural application of these methodologies is to train and classify for unwanted signals. The definition of "unwanted signals," being largely subjective, requires input from a trained analyst—that is, it requires manual labeling by a human before feeding data into any kind of data mining scheme. For temporal applications where unwanted periodic signals must be removed (e.g., fatigue testing and long term monitoring), the process can be made fairly automated. For example, in the case of bridge monitoring, it can be assumed that, in a new or otherwise undamaged structure, all AE activity occurring during service loads can be characterized as "background" emissions.

If focus is made on the fatigue life of a member which is part of a structure being observed, the acoustic emissivity is expected to rise as cracks nucleate, propagate through the material, and eventually lead to uncontrolled failure. In the presence of background signals, emissions due to crack-growth can be masked by extraneous "noise", and the onset of crack-growth can be especially difficult to discern. Because the types and distributions of the AE signals in fatigue testing are largely uncharacterized, identifying data points belonging to growing damage is particularly difficult. This problem is

exacerbated by the potential presence of multi-modal otherwise overlapping distributions for different AE sources, which makes signal differentiation considerably more difficult. This study works under the assumption that the distributions of unwanted and damage-related signals are sufficiently distinct and differentiable. If this premise is accepted, then a data mining scheme that can classify new data as belonging to a specific (though arbitrary) distribution becomes especially useful. Statistical outlier methods that rely on assumptions of normality will be inadequate for this application. Therefore, an outlier test that is distribution-independent is required in AE applications, where the distributions are necessarily not Gaussian. Distribution-independent outlier detection methods exist, and some do not merely classify a single data object as being or not being an outlier but also give an outlier score or outlier factor signaling the degree to which a respective data object is an outlier (Breunig et al. 2000). A major problem with these outlier detection methods is how to interpret this outlier factor in order to decide whether or not the data object is, indeed, an outlier. A local-density-based outlier detection method providing an outlier "score" in the range [0, 1] that is directly interpretable as a probability of a data object for being an outlier is needed in order to reject new data with a certain degree of confidence.

Local outlier probabilities (LoOP)

The concept of a normalized local outlier probability (*LoOP*) in the range [0, 1] was introduced by (Kriegel et al. 2009). This section summarizes the *LoOP* procedure and lists all assumptions involved in its application. In the following, D is a set of n objects and d is a distance function used to distinguish outliers. The data object being considered for outlierness is denoted by o. A probabilistic distance of $o \in D$ relative to a context set

$S \subseteq D$ (such as a neighborhood of *k* nearest neighbors) can be referred to as $pdist(o,S)$. This distance can be interpreted as the statistical extent of the context set *S*. The reciprocal of the probabilistic distance can be seen as an estimation for the density of *S*:

$$pdens(S) = \frac{1}{pdist(o,S)}. \tag{3}$$

Estimating the density of *S* by assuming the following property:

$$\forall s \in S : P[d(o,s) \leq pdist(o,S)] \geq erf\left(\frac{\lambda}{\sqrt{2}}\right), \tag{4}$$

where erf is the Gaussian error function, represents the case of outliers being defined as objects that deviate more than a given λ times the standard deviation σ from the mean. Assuming that *o* is the center of *S*, where *s* is the vector of all objects in *S*, and the set of distances of $s \in S$ to *o* is approximately half-Gaussian, then the standard distance of *S* to *o* can be defined as:

$$\sigma_{std}(o,S) = \sqrt{\frac{\sum_{s \in S} d(o,s)^2}{|S|}}. \tag{5}$$

The context set *S*, in this study, is computed as the *k* nearest neighbor query around *o*. Based on these considerations, the probabilistic set distance of *o* to *S* with significance factor λ is defined as:

$$pdist(\lambda,o,S) := \lambda \cdot \sigma_{std}(o,S). \tag{6}$$

Intuitively, this probabilistic set distance estimates the density around *o* based on *S*. The significance parameter λ gives control over the approximation of the density and

acts as a normalization factor. This parameter, however, only affects the contrast in the resulting outlier scores, and it does not affect the outlier ranking. The assumption that the *k*-nearest distances around *o* are half-gaussian does not limit this approach for other distributions, as asymmetrical distributions around *o* will result in greater probabilistic distances and, ultimately, a larger outlier score.

A probabilistic local outlier factor (*PLOF)* can be calculated as the ratio of the probabilistic distance around *o* divided by the expected value of the probabilistic distances around each member in context set $S(o)$:

$$PLOF_{\lambda,S(o)} := \frac{pdist(\lambda, o, S(o))}{E_{s \in S(o)}[pdist(\lambda, s, S(s))]} - 1 . \tag{7}$$

Note that the calculated *PLOF* is not yet a probability, nor is it normalized. To achieve a normalization making the scaling of *PLOF* independent of the particular data distribution, the aggregate value *nPLOF*, for all objects in *D*, is obtained during *PLOF* computation:

$$nPLOF := \lambda \cdot \sqrt{E[(PLOF)^2]} . \tag{8}$$

This value can be seen as a kind of standard deviation of *PLOF* values with an assumed mean of 0. In order to convert the not yet normalized *PLOF* value into a probability value, it can be assumed that the *PLOF* values are normally distributed around 1 with a standard deviation of *nPLOF*. The value of *nPLOF* needs to only be found for the training dataset and remains unchanged when testing new objects for outlierness. With this assumption, the Gaussian Error Function can be applied to obtain a probability value indicating the probability that a point $o \in D$ is an outlier:

$$LoOP_{s(o)} = \max\left\{0, erf\left(\frac{PLOF_{\lambda,s(o)}}{nPLOF \cdot \sqrt{2}}\right)\right\}. \quad (9)$$

The *LoOP* value will be close to 0 for points within dense regions and close to 1 for density based outliers. The outlier tolerance value, *p*, is the number of allowed outliers within the training dataset. That is, after performing a *LoOP* analysis on the training dataset, λ is adjusted so that $\forall LoOP_{s(o)} : P[p \geq p_{actual}] \geq 95\%$, where p is the outlier tolerance, and p_{actual} is the actual number of outliers. This condition ensures that all outliers have an outlier probability of at least 95% for any given λ. The relationship between p and p_{actual} is expected to behave linearly until a further decrease in p no longer affects the value of p_{actual}. In order to determine a conservative value for p_{actual}, a stable value for λ can be found by varying the outlier tolerance, *p*, within the training dataset. The optimal value of λ will correspond to the first point where decreasing p does not cause an increase in the value of λ, as indicated in the sample plot of λ vs. p for the dataset examined in section 4.1 shown in Figure 10. At this point, the value of p_{actual} will also be constant for each decrease of p. Choosing the higher constant value of λ in Figure 10 will result in a smaller value of p_{actual} (and a larger enclosing minimum volume), which will possibly cover any local outliers within the training set. An unconservative choice of λ may cause the algorithm to be overly tolerant and will result in an underestimated number of outliers in both the training and testing datasets. Once λ has been estimated for the training dataset, it should remain constant during the testing phase.

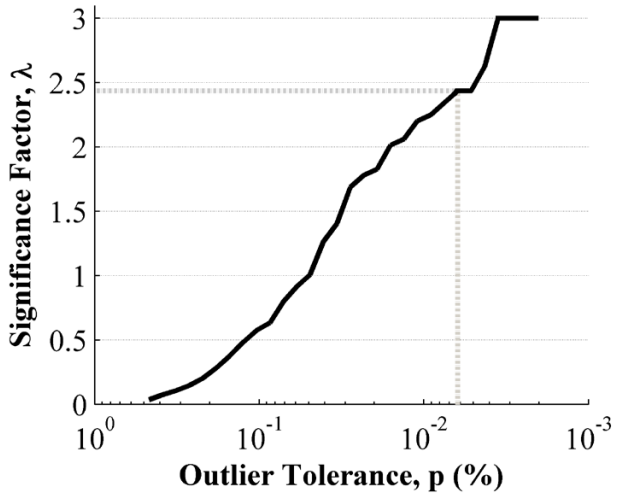

Figure 10 – Significance factor vs. outlier tolerance

Extending the *LoOP* procedure to act as a detection tool of new AE sources in testing datasets is trivial. When testing an object outside of the training dataset for outlierness, a neighborhood of k nearest neighbors, $S(o)$, must be sampled from the training dataset D when calculating $LoOP_{s(o)}$. Using the values for λ and *nPLOF* calculated during training ensures that all new objects that exceed a local outlier probability of 95% will also be classified as outliers, and that those that do not belong to the same distribution as the inliers. At this point, a binary class label c_i, where index $i = \{1, 2\}$ i represents the class cardinality, can be added to the feature set. Herein, inliers have the label $c_1 = 1$ while outliers are labeled $c_2 = -1$.

The performance of the *LoOP* algorithm depends on sufficiently populated context sets, which are generated through a search for a predetermined number of nearest neighbors. The number of nearest neighbors, k, should be sufficiently large to accurately represent the local distribution without being so large to be too computationally taxing. In small training datasets, very large values of k (e.g., k>50) can result in decreased

performance, as the ensuing local context sets may draw objects from more than one local distribution. The performance of *k* versus outlier detection accuracy in a typical AE dataset is examined in Figure 11.

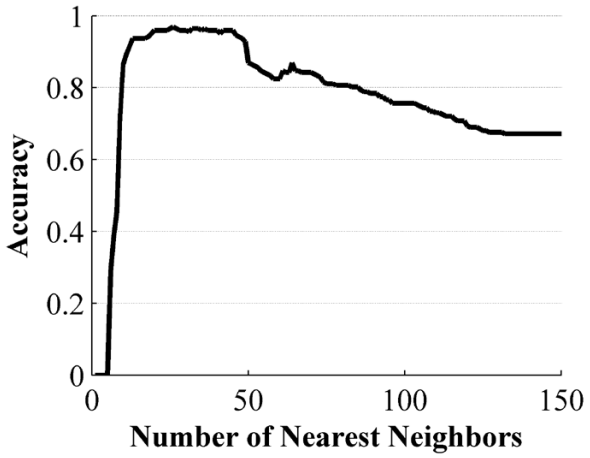

Figure 11 – Classification accuracy vs. number of nearest neighbors: AE dataset

Unsupervised clustering

As previously mentioned, the outlier detection algorithm will produce two separate classes—one positive and one negative. The positive class will, by convention, represent the inlier points within the testing dataset, and the negative class will represent the outlier data. Since the overarching goal of this data mining scheme is to extract rules for the characterization of unwanted signals, distinct "noise" sources must be reliably separated. Failing to further cluster the inlier (or outlier) context sets may result in rules that are not indicative of any particular AE source, as the rules will instead attempt to cover multiple emission sources at once. Rules that simply differentiate outliers from inliers, although possibly being more succinct, will not necessarily represent a single AE source and may fail to reveal interesting structures within the dataset.

A solution to this problem is to further cluster both of the sets found by the outlier detection algorithm, and assign a sub-cluster label $c_{i,j}$ to each outlier object, where index $j = \{1,...,K\}$ is a sub-cluster class cardinality for the inlier and outlier class *i*. Obtaining a set of mutually exclusive clusters is not a particularly difficult task. A popular algorithm such as K-means (Lloyd 1982), if carefully implemented, can yield reproducible and accurate partitions from a dataset (Maitra et al. 2010). There are several clustering algorithms that do not require previous estimation of the number of clusters, *K*, such as DBSCAN (Ester et al. 1996) and CURE (Guha et al. 2001). In this study, the X-means algorithm provides an easily implementable solution based on the original K-means, with a provision for optimal cluster determination termed the Bayesian Information Criterion (BIC).

The K-means algorithm, despite being the most prevalent clustering algorithm used in scientific and industrial applications (Berkhin 2002), has been documented to suffer from three major shortcomings: it scales poorly computationally; the number of clusters *K* has to be supplied by the user; and the search is prone to converging to local minima (the number of clusters *K* should not be confused with the number of *k* nearest neighbors used in *LoOP*). X-means tries to address these problems by introducing the BIC. If $X = \{x_1,...,x_n\}$ is the dataset to be clustered, let $X = X_1 \cup X_2 ... \cup X_K$ be a clustered dataset made up of *K* clusters. Each cluster in X can be modeled as a Gaussian distribution $N(\mu_i, \Sigma_i)$, where μ_i can be estimated as the sample mean vector and Σ_i can be estimated as the sample covariance matrix. Thus the number of free parameters *f* of

each cluster is equal to $K \cdot (m+1)$, where m is the number of features being clustered. Under these assumptions, the BIC can be computed as:

$$BIC(X_j) = \sum_{j=1}^{K} \left(|X_j| \log(|X_j|) - \frac{|X_j|}{2} \log(2\pi) - \frac{|X_j| \cdot m}{2} \log(\Sigma(X_j)) - |X_j| \log(|X|) - \frac{|X_j| - K}{2} \right) - \frac{f}{2} \log(|X|), \quad (10)$$

The main benefit of the K-means algorithm lies on its simplicity:

1. For each point x, find the centroid which is closest to x. Associate x with this centroid.
2. Re-estimate centroid locations by taking, for each centroid, the center of mass of points associated with it.

Extending this algorithm into the X-means algorithm to adaptively find new centroids involves two additional steps (Pelleg and Moore 2000):

3. Find out if and where new centroids should appear by splitting each centroid in half along a randomly chosen vector, running K-means locally on the two children and parent context sets, and comparing their BIC scores.
4. If the new $K > \sqrt{|X_j|}$ or a predefined maximum number of clusters, stop and report the best scoring model found during the search; and, if $BIC(X) > BIC(X_j)$, choose children centroids as new seeds and repeat steps 1-3 until convergence.

Running the X-means algorithm on each context set D_i, where i is the inlier or outlier class obtained from the outlier detection procedure, should reveal a series of K_j clusters per set, where j is the sub-cluster class index. Each object should be assigned a class feature $c_{i,j}$, which will be used as the training class label during rule extraction.

Rule extraction

Assuming sufficient distinction between AE sources, each class obtained through the X-means process will yield more easily interpretable rules than the set of all classes combined, which is implied by the search for statistical dissimilarity inherent in the BIC scoring process. In theory, if each class lies within mutually exclusive regions inside the Euclidean space, then each rule obtained through the rule extraction process should partition the hyperspace in such a way that each rule covers each class in its entirety.

Even though there exist a plethora of algorithms for rule extraction, in the proposed study it is preferred that the obtained rules be order independent (i.e., not a decision list), and they be comprised of binary splits in order to create rules that are physically interpretable and, more importantly, easily programmable using the primitive Boolean filters available in commercial AE software. A decision tree produces such binary classification rules for continuous data, and despite shortcomings such as subtree replication, it ensures that each AE "hit" falls under only one class or AE source. Of all the decision trees, this study employs the *J48* algorithm, which is an open source version of the popular *C4.5* algorithm introduced by (Quinlan 1993). This algorithm produces a series of nested *if-else* binary conditionals or rules, which are usually displayed in the form of a binary tree when used on continuous datasets.

The *C4.5* algorithm constructs a decision tree through a divide-and-conquer strategy. In *C4.5*, each node in a tree is associated with a set of objects. At the beginning, only the root is present and associated with the whole training set Y_{tr}. At each node, the following divide-and-conquer procedure is executed until the best local choice is found, with no backtracking allowed:

1. Compute the frequency for each class in *Y*, which is the set of objects associated at the node.
2. If all objects in *Y* belong to the same class $c_{i,j}$, or the number of objects in *Y* is less than a certain value specified by the user, then the node is turned into a leaf labeled as majority class $c_{i,j}$. The classification error of the leaf is calculated as the weighted sum of the objects in *Y* whose class is not $c_{i,j}$.
3. If *Y* contains objects belonging to two or more classes, where $|c_{i,j}| = NClass$, then the information gain ratio obtained from splitting Y into two sets Y_l should be calculated for each attribute, where,

$$gain(Y, Y_l) = \left(\sum_{l=1}^{2} \sum_{j=1}^{NClass} \frac{freq(c_{i,j}, Y_l)}{|Y_l|} \log_2 \left(\frac{freq(c_{i,j}, Y_l)}{|Y_l|} \right) \right) - \sum_{j=1}^{NClass} \frac{freq(c_{i,j}, Y)}{|Y|} \log_2 \left(\frac{freq(c_{i,j}, Y)}{|Y|} \right), \text{ and} \quad (11)$$

$$gainratio(Y, Y_l) = -gain(Y, Y_l) / \sum_{l=1}^{2} \frac{|Y_l|}{|Y|} \log_2 \left(\frac{|Y_l|}{|Y|} \right) \quad (12)$$

4. The attribute with the highest gain ratio is selected as the condition attribute to be tested at the node.
5. The threshold where to split the attribute is computed by choosing the best gain ratio from all possible splits in-between any two successive objects after sorting *Y* in ascending order by the values of that attribute.
6. Iteratively perform steps 1-5 on the new *Y* obtained after applying the rule at the previous node.
7. Calculate the classification error at each node, where the total error is the sum of the errors of each of its child nodes. If the calculated error is greater than the error

of classifying all cases in *Y* as belonging to the majority class in *Y*, then the node is turned into a leaf and all subtrees are removed.

Programmable rules (i.e., rules that can be implemented in AE applications using primitive Boolean filters) can be extracted from a decision tree by turning each node into an *if-else* statement, and nesting these conditionals until reaching a classifying leaf node. Ideally, one nested rule should be able to cover all objects in each class, but in practice, the coverage of each rule may be reduced, and several rules may be required to represent each class.

Experiments and discussion

The proposed data mining scheme will be validated in three steps. First, it will be applied to a synthetic dataset where the results are intuitive and easily interpretable. Then, it will be applied to an AE dataset where AE sources are known and labeled *a priori*. The algorithm's accuracy and rule generation under these conditions will be examined. This dataset will be used to determine a suitable number of nearest neighbors to be used when testing for outlierness and it will prove that statistically dissimilar AE sources can be systematically segregated. Finally, the scheme will be applied to AE datasets obtained from two SE(T) fatigue crack-growth steel specimens, where background "noise" is a common problem, and the results will be compared to conventional background "noise" removal practices.

Performance on a synthetic dataset

A training set of two-dimensional data consisting of two Gaussian distributions is generated synthetically. The mean and standard deviation of each distribution are chosen so that the minimum volumes enclosing each distribution do not overlap. A two-

dimensional plot of this data is shown in Figure 12a. A similar two-dimensional testing dataset draws points from the same distribution as the training dataset in addition to drawing from two separate non-overlapping distributions. This synthetic testing dataset is plotted in Figure 12b.

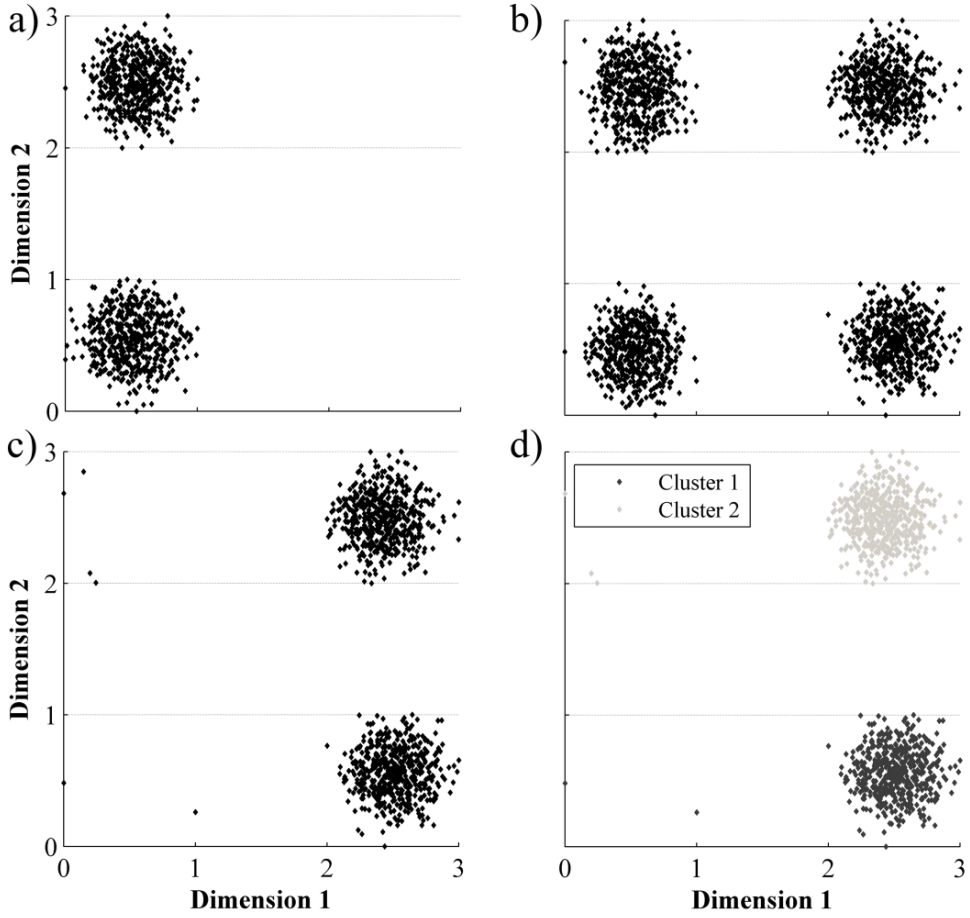

Figure 12 – Synthetic dataset: a) training data; b) testing data; c) outliers in testing data only; d) clustered outliers in testing data only

Intuitively, all points in the leftmost two distributions in the testing dataset should be classified as inliers, where all points in the rightmost two distributions should be found to be outliers. Furthermore, clustering the outlier data should reveal two separate

distributions within the outliers, and the extracted classification rules should partition both dimensions somewhere between 1 and 2 and correctly point to each class.

For the outlier detection step in the proposed data mining scheme, λ was found to be equal to 2.4 with a percentage of training outliers, p_{actual}, of 0.6%. For this dataset, the value of k when evaluating the k-nearest neighbors did not have a significant effect in the outlier detection accuracy, but it is suggested that $k \geq 30$ since it is commonly held that 30 is the minimum sample from which one should draw statistical inferences (Weiss 2012). This is particularly relevant with expected value calculations, which follow the central limit theorem that states that for $k \geq 30$, the sample mean tends to approach a normal distribution with an increasingly smaller standard deviation. The sample mean of the probabilistic distances between nearest neighbors is used in the calculation of *PLOF* and exhibits this property, thus suggesting the need for setting $k \geq 30$. Therefore, for this study, though smaller values of k could provide adequate results, the value is set to 30 to provide a more accurate representation of each local distribution without possibly choosing a k that is larger than a local distribution. The outliers obtained from the outlier detection step are shown in Figure 12c. Note that because the outlier detection algorithm is designed to be conservative when classifying objects as inliers, 0.6% inliers were incorrectly classified as outliers. This is, not coincidentally, the same percentage of outliers as in the testing dataset, since both datasets draw their inliers from identical distributions.

The X-means clustering step, also as expected, correctly distinguishes between the two different outlier distributions. Since X-means is a hard clustering algorithm, all points are assigned to one of the two found clusters. The clustered outliers are depicted in

Figure 12d. In case of extremely large testing datasets, the X-means process can be made more efficient by first constructing a K-D tree with a uniformly sampled subset of the training dataset (Pelleg and Moore 2000).

Efficiencies in the rule-extraction process can also be achieved by first discretizing the continuous dataset. Weighted proportional k-interval (WPKID) equal frequency discretization, as proposed by Yang and Webb (2003), will result in a faster runtime and may even improve the classification error for some classifiers. The final decision tree obtained from applying the *C4.5* algorithm to the clustered dataset can be seen in Figure 13.

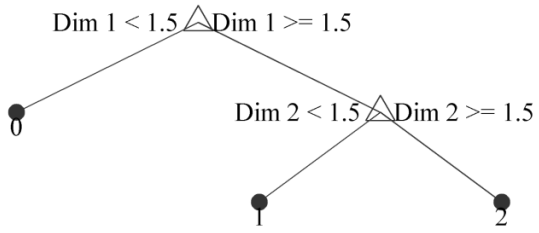

Figure 13 – Decision tree with leaf nodes as clusters: synthetic dataset

In this decision tree (as in the other decision trees in this study), the leaves are labeled "0" for inliers, "1" for *Cluster 1*, "2" for *Cluster 2*, and so on. This decision tree was found to have an overall classification accuracy of 99.7% when compared to the *LoOP* labels, though the generated rules are 100% accurate in segregating the right-most distributions in the testing dataset. As it has been shown, the resulting decision tree illustrates that the algorithm does in fact produce rules that are intuitive and accurate.

Performance on a manually classified AE dataset

AE data varies from the synthetically generated dataset in a few key ways. First, not all features measured from a single AE source are normally distributed. Second, the

number of features that are normally extracted from a typical AE "hit" is usually quite large. In this study, all AE datasets are composed of 15 unique features, all of which are assembled into a 15-dimensional matrix of *n* instances or "hits." Features that are not considered are any constant, monotonically increasing or decreasing features, or unique ID features such as *time of test*, *channel number,* and *threshold.* All features obtained by the AE data acquisition system are positive scalars.

AE monitoring data is typically plotted as a time-series of amplitude measurements also called an "*amplitude vs. time"* plot. However, because the "hit" *time-of-test* stamp is not considered in this study, the *time-of-test* dimension is not considered when finding outliers and plotting values such *amplitude vs.* time functions serves simply as a convenient presentation. Thus, points that seem to be outlying in a time series plot and are found to have very low *LoOP* scores will not be labeled as outliers.

In order to generate an AE dataset suitable for validation of whether AE data can be discriminated using the proposed data mining scheme, a simple experiment was designed. A 29x2.0x0.5in A572-G50 steel specimen was placed on top of layered wood, styrofoam, and rubber supports in order to isolate it from external "noise." The specimen was instrumented with six Physical Acoustic Corporation® (PAC®) R15I-AST resonant sensors in an identical layout to the sensor layout detailed in (Nemati 2012). The sensors were connected to a *PAC® PCI-2 Rev. 3* AE data acquisition system. At the specimen midspan, a PAC® FieldCAL™ AE pulse generator was connected as shown in Figure 14.

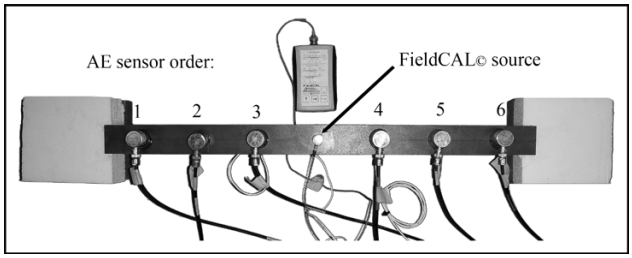

Figure 14 – PLB and FieldCAL™ signal AE test setup

A "noise" training dataset was collected by cycling the FieldCAL™ through all of its amplitude and frequency combinations for a duration of 180 seconds for each setting, as given in Figure 15a.

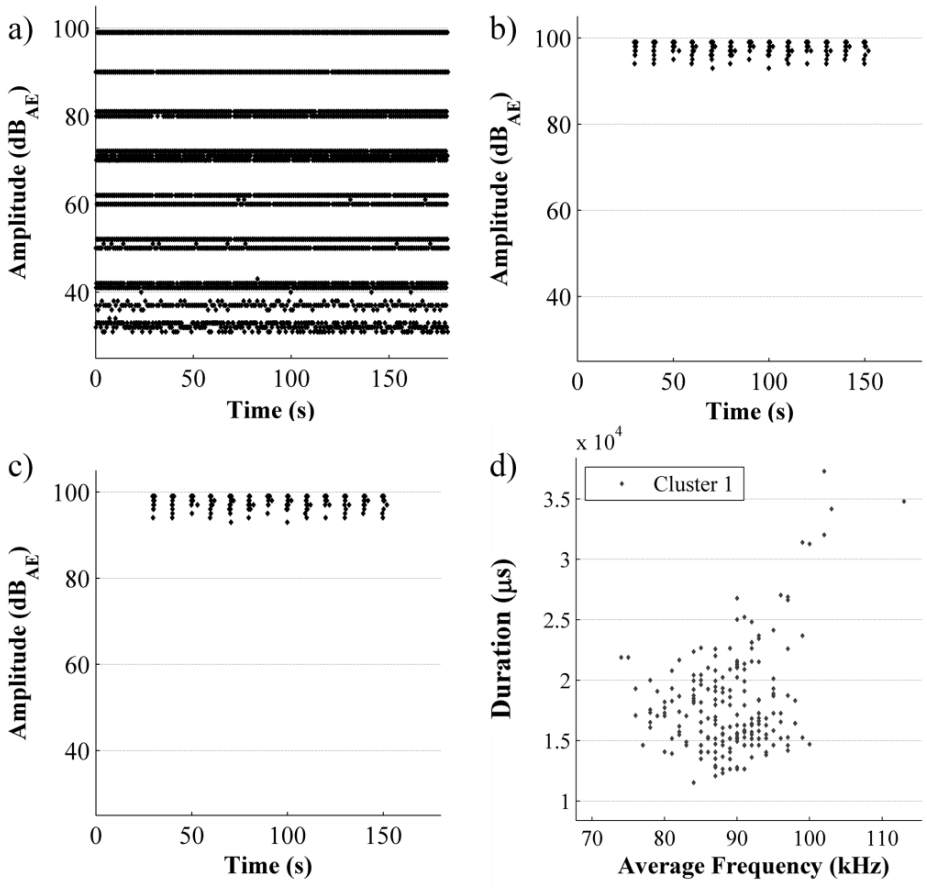

Figure 15 – AE dataset (sensor 3): a) FieldCAL™ hits used for training; b) PLB hits used for testing; c) outliers in testing data only; d) clustered outliers in testing data only

The testing dataset was generated by collecting data from 221 pencil lead breaks (PLBs) at the specimen midspan. The PLBs were performed as per American Society for Testing and Materials (ASTM) E976 (2010) guidelines, where a 0.20in diameter, 0.11in long pencil lead is broken on the material surface in order to simulate a sudden crack propagating within the material. PLBs are typically characterized by being very high amplitude hits when performed within inches of a sensor. These hits were separated manually from any reflections by selecting the highest amplitude hit immediately following each PLB and are illustrated in Figure 15b. Even though data was collected by all six sensors, the aim of this study is to provide filters and characterization rules without the need for multiple sensors. While this study focuses only on the sensor closest to the PLB source, the other sensors are used for data validation and for spatial filtering.

When running the *LoOP* algorithm on this AE dataset, the optimal value of k was found to be between 15 and 45 nearest neighbors (Figure 11). A value for k equal to 30 is used for this dataset, as it lies in this middle of the acceptable range for k and, as previously explained, it guarantees the nearest neighbor distributions are adequately represented. For the AE dataset, the value for λ is estimated to be equal to 4.0 with a percentage of training outliers equal to 0.3%. The *LoOP* algorithm mislabels 2.7% of the PLBs as belonging to the FieldCAL™ distribution, with the other 97.3% of testing points correctly labeled as outliers, as shown in Figure 15c. The ability for the *LoOP* algorithm to successfully discriminate PLBs from the large dataset of background "noise" suggests that there can be sufficient distinction between hits of similar *amplitude* and *peak frequency* provided that other AE features have sufficient contrast. Even though the PLBs are tested for outlierness versus AE-like pulses of 150kHz and $90dB_{AE}$, which is the

theoretical “signature” of PLBs, there exist enough discriminating features to separate PLBs from this background “noise”. Moreover, when using a mix of PLBs and FieldCAL™ data as the testing dataset, all FieldCAL™ hits are also correctly identified as inliers by the *LoOP* algorithm.

The clustering algorithm, not surprisingly, returns only one cluster for the set of outliers. A *duration vs. average frequency* plot, shown in Figure 15d, confirms that the PLBs are grouped into a single cluster. Other two-dimensional plots not shown reveal a similar cluster structure. In the decision tree generation phase, the *C4.5* algorithm produces a two-level decision tree with only two features being considered as presented in Figure 16.

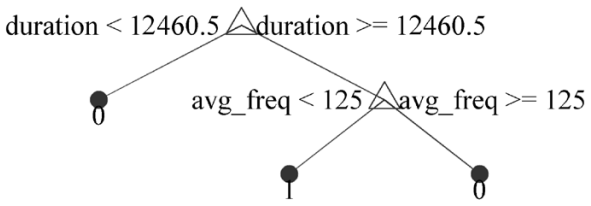

Figure 16 – Decision tree with leaf nodes as clusters: AE dataset

Interestingly, the two discriminating features are *duration* and *average frequency*, instead of more commonly used features like *amplitude* and *peak frequency*. The classification rule for PLBs characterizes them as hits with *average frequency* less than 125kHz and a *duration* greater than or equal to 12.5ms. This rule correctly classifies 98.6% of the testing dataset as outliers, and it matches the predicted outlier detection labels with 99.8% accuracy. These results suggest that the outlier detection performance of this scheme when used in AE data is satisfactory even when using the rules obtained from the decision tree.

Outlier detection in SE(T) fatigue-crack growth AE datasets

With evidence that the outlier detection algorithm is able to discriminate between hits of similar amplitudes and peak frequencies provided that the other features are sufficiently dissimilar, a natural extension of the proposed data mining scheme is to utilize it as a background "noise" removal tool. Steel fatigue crack-growth testing is notoriously fraught with background "noise" problems, and "noise" damping, spatial filtering, and high amplitude thresholds are often used to obtain workable datasets (Berkovits and Fang 1995). Commercial systems such as the *PAC® PCI-2* AE data acquisition system allow for the setting of conditional front-end filters, and all rules produced in the form of decision trees can be easily implemented using these tools. Additionally, post-processing filters found in AE data analysis software such as *NOESIS®* also follow the same conditional rules, and implementation is identical to that of front-end filters within the AE data acquisition software.

Removal of unwanted signals in fatigue crack-growth testing is facilitated by the abundance of collected data, particularly during early stages of testing prior to the onset of crack-growth. AE data collected before crack growth can safely be assumed to be caused by extraneous sources. Additionally, data collected under sub-critical stress intensities can be similarly labeled as background "noise." While conducting this study, it was observed that the best training datasets are those which, when plotted as a time series, appear to be constant. Specifically, time-series plots of frequency parameters appeared to be particularly sensitive to crack-growth activity. Therefore, in the absence of a crack mouth opening displacement (CMOD) gauge or similar crack-growth measuring

device, it is advisable to use a training dataset that behaves in a constant fashion when viewed as a frequency time-series.

This study focuses on two ASTM A572-G50 steel fatigue crack-growth raw datasets collected by Nemati (2012), where SE(T) specimens were outfitted with AE sensors in the same configuration as Figure 14. In addition to these sensors, a CMOD gauge was placed on the pre-notched crack opening, and an optical microscope was used to determine the visual onset of crack-growth. Both specimens are wedge-gripped, and they are subjected to constant tensile cyclic loads.

When utilizing portions of the testing dataset as training data, it is essential that the chosen data points be relatively homogenous, and that there be enough instances for each type of hit or AE source in order to form well-represented neighborhoods. Therefore, data collected immediately following the start of a fatigue test should not be used since the fretting and friction phenomena associated with grip “noise” and frame engagement can be potentially similar to that of crack-growth activity. Grip and frame engagement-related emissions tend to subside shortly after test initiation, which can be seen as a marked drop in peak amplitude, followed by a period of constant emission rate. In this study, the period of constant AE activity detected by sensor 3 before the onset of crack-growth (as measured by the CMOD) is used as training data, as shown in Figure 17a.

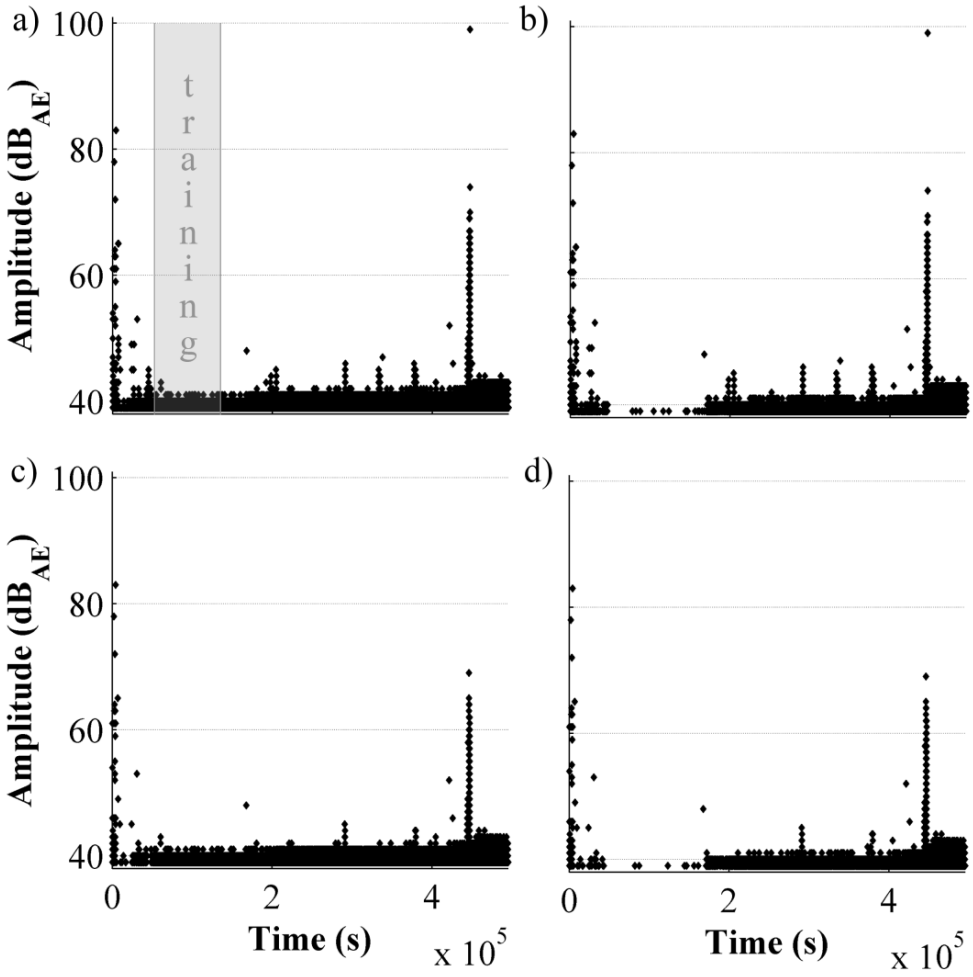

Figure 17 – SE(T) specimen S9 dataset (sensor 3): a) raw data; b) outliers in testing dataset; c) dataset filtered using guard sensors; d) dataset filtered using guard sensors and outlier detection

The entire dataset including the training portion is used for testing. Note that the training interval in the testing dataset will contain a number of outliers smaller or equal to the chosen outlier tolerance.

Intuitively, outliers found using this methodology should be directly related to crack-growth phenomena, except for early emissions due to grip and frame engagement, which are not accounted for in the training dataset. The rate of outliers is expected to increase proportionally to the crack-growth rate, and the peak emission rates and amplitudes should occur near the specimen's failure.

The outliers obtained from the *LoOP* algorithm can be seen in Figure 17b. As expected, there is an initial short period of high activity followed by a span of little to no activity. Crack-growth initiation was measured to occur at 290,000 cycles, or 1.45 x 10^5 sec., which coincides remarkably well with reappearance of significant outliers at the 1.44 x 10^5 sec. mark.

Clustering exposes three subclasses within the outliers, and further examination reveals that the X-means algorithm groups those hits with a very large *initial frequency*, which are due to waveforms that cross the amplitude threshold only once prior to the waveform peak and whose rise time is limited to the granular resolution of the AE system (1μs). This hardware-related artifact is manifested in the *counts to peak*, *rise-time*, and *initial frequency* parameters. The second cluster is a similar grouping of hits with a *duration* of 1μs (again, due to the time-resolution of the system) and one *total count*, which is equivalent to an *average frequency* of 1000kHz. Neither of these clusters should be systematically eliminated since, even though they are spuriously clustered together due to issues of resolution, they may still correspond to legitimate crack-growth events.

The rules obtained from the *C4.5* algorithm, as illustrated in Figure 18, show that both spurious clusters share the same secondary node in the rule tree, which means that they can be both be described by a single rule: *rms voltage* ≥ 0.0015mV & *duration* ≤ 1.5μs.

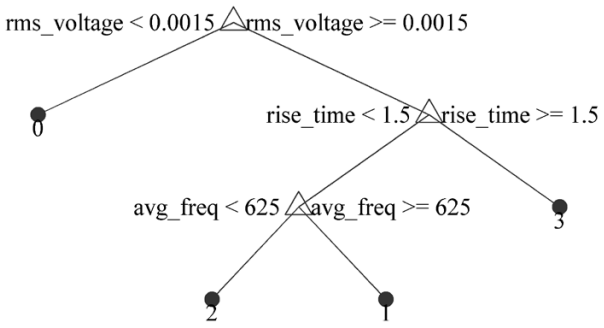

Figure 18 – Decision tree with leaf nodes as clusters: SE(T) specimen S9

The branch leading to the three outlier clusters can be represented by the root node. That is, the value of *rms voltage* can adequately describe outlying emissions. Ten-fold cross-validation of the testing dataset reports an accuracy of 99.8% when utilizing this rule to filter out background "noise." This rule is order independent and can be applied at any point of the filtering process. For example, Figure 17c shows the same dataset filtered by spatial techniques. The preferred method of background "noise" removal in fatigue testing is that of guard sensor filtering. This method consists of removing hits based on the order in which they arrive to each sensor; if sensors detect an AE event in the direction opposite to the expected propagation, then all signals associated with that event are discarded. In this study, hits within 0.5sec. of all events where guard sensors were first to detect a hit are removed. This conventional filter removes approximately 90% of all data yet, just as in the raw dataset, the onset of crack-growth cannot be clearly discerned. Applying the first rule on the decision tree results in the filtered dataset in Figure 17d. This dataset closely resembles the original outlier dataset in Figure 17b, except for the conventional filter removing three large amplitude spikes. Conventional filters, as evidenced, may remove spurious spikes in historic index and cumulative plots,

but do not affect the change in hit rate which, for the elastic portion of the crack evolution, should behave proportionally to the crack-growth rate (Yoon et al. 2000). However, an outlier filter such as the one described in this study does, in fact, expose the onset of crack-growth without removing potentially relevant emissions.

Depending on the number of clusters found by the X-means clustering algorithm, it can be seen that extracted rules may expose a structure within the data caused by segregation of frequency features derived from low resolution parameters, which tend to act like discrete attributes at certain values. Nonetheless, rule extraction can produce relevant rules that further segregate the outlier dataset.

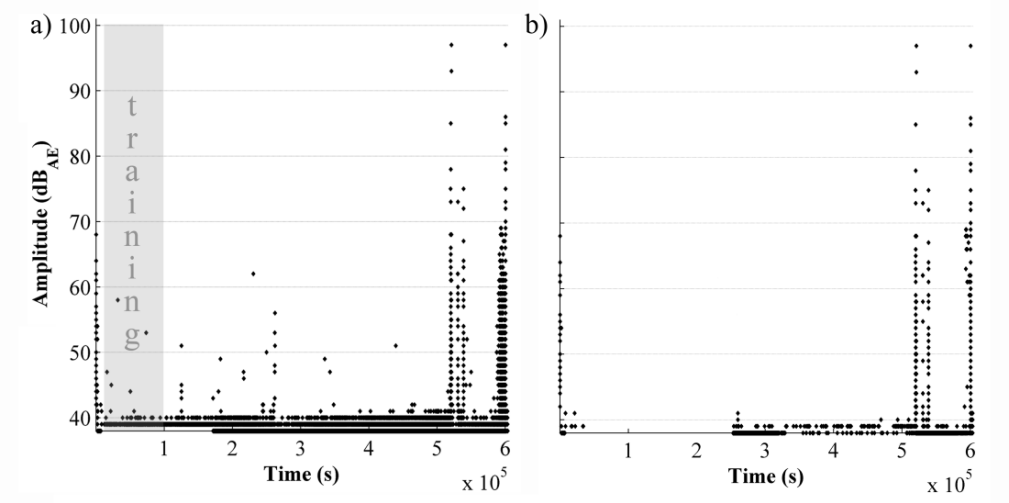

Figure 19 – SE(T) specimen S4 dataset (sensor 3): a) raw data; b) outliers in testing dataset

Figure 19a shows a raw dataset from a similar SE(T) specimen than the one used in Figure 17 (tested at an *R* ratio (i.e., the ratio of the minimum load to the maximum load in a fatigue cycle) of 0.10), though this specimen is subjected to a significantly higher *R* ratio of 0.65. For this specimen, the *amplitude* time-series shows similar trends than the previous example. The training dataset is recorded after a 3-hour initial settling period of high emissivity and is stopped after 24 hours, approximately 44 hours before the

measured onset of crack-growth. The data obtained after the outlier detection step is shown in Figure 19b. The outliers appear a few minutes after the recorded onset of crack-growth at 2.59 x 10^5sec.

The cluster structure reveals a subset of the data with a very small *rise time* of 1 μs, which is again clustered due to *initial frequency* being directly calculated from this value and being discrete for waveforms with few *counts to peak*. The bulk of the clustered emissions is characterized by having a high *absolute energy* and a low *average frequency*.

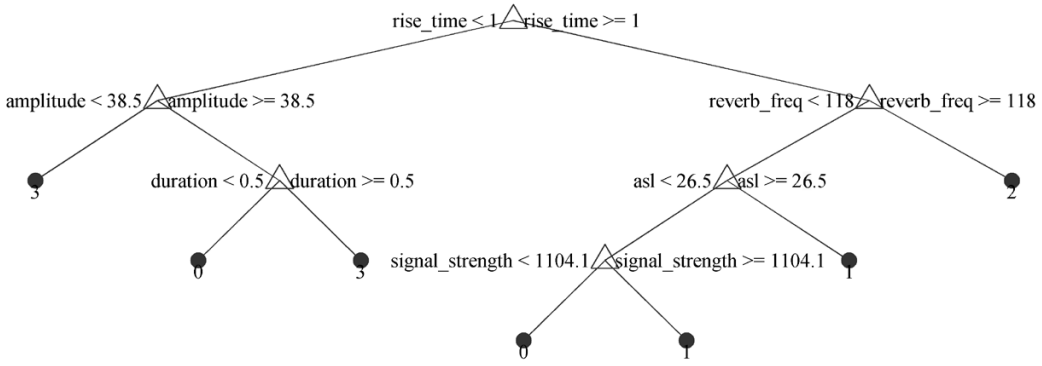

Figure 20 – Decision tree with leaf nodes as clusters: SE(T) specimen S4

Upon examining the extracted rules in Figure 20, the decision tree's left branch (composed of Cluster 2 and inliers) highlights the subset of spurious hits, while the right branch of the decision tree with a longer duration is characterized by Cluster 1 hits with a rev*erberation frequency* < 118kHz and an *average signal level* > 26.6dB$_{AE}$. Cluster 3 is characterized by a high *reverberation frequency*, low *amplitude* component and only occurs around sudden *amplitude* spikes above 45dB$_{AE}$. Ten-fold cross-validation of the testing dataset reported an accuracy of 98.4%

The rule covering most of the outlier data (Cluster 1) is of particular interest once analyzed further since it can be related to the rule obtained for the previous example. Due to the direct mathematical relationship between *average signal level* (ASL_{AE}) and *rms voltage* (*RMS*), the value for ASL_{AE} can be linearly approximated as a function of *RMS* around the values of 24 and 30 dB_{AE} by the expression

$$ASL_{AE} = 5000 \cdot RMS + 19 , \tag{13}$$

where the value of *RMS* is expressed in millivolts, and the resulting *ASL* is given in dB_{AE}. Using this expression, it can be seen that an *RMS* value of 0.0015 corresponds, approximately, to 26.5dB_{AE}. In other words, the rule leading to the outlier branch in Figure 18 is identical to the primary rule obtained for Cluster 1 in Figure 20. This similarity suggests that the extracted rules can, in fact, be generalized between similar specimens and, possibly, specific damage mechanisms for a given material. It is advised that, when trying to characterize failure mechanisms in terms of generalized rules, AE features with high resolution be used, and features directly calculated from one another (such as *average signal level* and *rms voltage*) be avoided.

Concluding remarks

The data mining scheme proposed in this study was found to perform satisfactorily, with classification accuracy for known datasets never falling below 95%. On synthetic datasets, outlier distributions were appropriately detected, and the scheme produced rules that are intuitive and correctly classify the data. In a controlled AE experiment involving PLBs and background "noise," the PLBs were correctly classified as belonging to a different distribution from those in the training dataset. Rules obtained for PLB classification performed with promising accuracy, as the rules correctly described 98% of

the dataset. Applying the scheme to fatigue crack-growth data also showed satisfactory results, with "noise" being successfully removed and exposing AE belonging to crack-related phenomena around the same time as the measured onset of crack-growth. Rules extracted from crack-related AE seemed to apply similarly across two tested specimens despite their different loading conditions. In particular, it was found that AE due to crack-related phenomena could be characterized as having an *average signal level* $\geq 26.5\text{dB}_{\text{AE}}$ for both of the specimens. While some of the extracted rules pointed towards spurious clusters commonly found as a consequence of low-resolution attributes, the rules obtained for these specimens hints at the possibility that AE "signatures" can be found for specific failure mechanisms and for different boundary conditions. Even though this study only suggests that rules can be generalized across similar specimens and boundary conditions, this methodology can be replicated when finding more general rules. If rules can, in fact, be generalized, then this scheme could not only produce efficient tests for outlierness within similar datasets but, if performed across enough specimens, could yield rules that effectively characterize the failure mechanisms that govern fatigue-fracture. Ultimately, it is hoped that rules that act as a general "signature" for AE from failure mechanisms can become part of warning systems and data filters in future SHM applications.

CHAPTER 4, STUDY 3: NEURAL NETWORK FORECASTING OF ACOUSTIC EMISSION PARAMETERS FROM FATIGUE CRACK-GROWTH DATASETS

Summary

A major role of structural health monitoring structural health monitoring (SHM) and non-destructive evaluation (NDE) efforts is the assessment of current and future damage conditions in structural members. This study presents a framework for crack-growth forecasting of structural elements subjected to cyclic loading. The framework forecasts acoustic emission (AE) and crack mouth opening displacement (CMOD) measurements. Predicted values are the input for linear-elastic fracture mechanics (LEFM) models to determine crack size. Short-term and long-term forecasts are accomplished using variations of a nonlinear autoregressive exogenous artificial neural network (NARx ANN) design. The framework is validated and tested using data collected from two single edge (tension) (SE(T)) specimens. The achieved forecasting accuracies indicate that ANNs provide an adequate way to model the complex relationships between LEFM parameters and AE measurements. Moreover, the study suggests that averaging estimated crack sizes obtained from predicted AE values yields an accurate long-term prediction of both crack size and crack-growth trends. Results also show that AE parameters can be predicted in a short-term and long-term fashion, and that crack-growth can be accurately estimated given adequate models.

Background

AE and LEFM crack size estimation

AE has been shown to occur during testing of metals in fatigue crack-growth experiments. It is well known that a portion of the energy released at the instant of crack growth is dissipated in the form of heat and elastic waves. AE is produced not only by the onset of yielding at the crack tip or crack extension, but also by the contact friction of fatigue crack surfaces due to closure (Morton 1973). Abrading surfaces produce frequent emissions, which have a slow rise time and low amplitude. AE from crack closure can occur even during tension-tension cyclic loading (Adams 1972). The premise behind AE monitoring in fatigue application is that the features extracted from waves that meet certain threshold criteria are an indication of the energy being released, which in turn can be used to estimate the current state of the crack using fracture mechanics theory. Generally, AE "hit" rates and certain extracted parameters have been shown to increase as the crack growth increases.

These relationships have been the subject of research since the early 1970s. For example, in early studies, Harris and Dunegan (1974) proposed a bilinear empirical relationship between the "hit" rate and the stress intensity factor (SIF) range. Further studies found that similar proportionalities between the "hit" rate, rate of crack-growth, and SIF range (Morton et al. 1974; Lindley et al. 1978; Williams 1982; Lee 1989; Fang and Berkovits 1995; Daniel et al. 1997; Oh et al. 2004). Of particular relevance are power-law expressions linking AE parameters to LEFM parameters.

Gong et al. (1992) proposed a relationship between the AE counts, η, and the SIF range, ΔK:

$$\frac{d\eta}{dN} = B \cdot (\Delta K)^g , \tag{14}$$

where B and p are constants for a particular material. A similar relationship between the AE abs. energy rate, U, and ΔK was introduced by (Yu et al. 2011):

, (15)

with B' and p' being material and boundary condition constants. Combining the latter expression with the Paris-Erdogan Law (Paris and Erdogan 1963) yields the following relation:

$$\frac{da}{dN} = D \cdot (\frac{dU}{dN})^q , \tag{16}$$

where a is the crack size, and D and q are material and boundary condition constants. The above expression suggests there is a relationship between the crack-growth rate and the energy release rate. The energy released by crack extension, plastic deformation, and fracture events within the plastic zone, U_a, can be expressed as follows (Dowling 2007):

$$U_a = \frac{{K_{\max}}^2 \cdot a \cdot b}{E'} , \tag{17}$$

where $K_{\max}$ is the maximum stress intensity factor, b is the plate thickness corresponding to the maximum possible crack size, and E' is equal to Young's modulus. For an existing geometry $K_{\max}$ can be expressed by:

$$K_{\max} = \sigma_{\max} \cdot \sqrt{\pi \cdot a} \cdot F(\frac{a}{b}) , \tag{18}$$

where σ is the nominal tensile stress on the SE(T) specimen cross-section due to axial load, and $F(a/b)$ includes the effects of limited thickness and gripping conditions. By combining equations 17 and 18, a reasonable assumption can be formulated that the change in crack size, Δa, and the accumulated AE energy released during crack-growth, *U,* are proportional and related by a similar power law, which can be represented by the following proposed expression:

$$\Delta a = \beta \cdot (\Delta U)^{\alpha}, \tag{19}$$

where β and α are material, geometry, and boundary condition constants. For a pre-notched specimen, the new crack size after *N* cycles, a_N, can be estimated by (as $U_0 = 0$):

$$a_N = \beta \cdot (U_N)^{\alpha} + a_0, \tag{20}$$

where a_0 is initial pre-notched crack size, and U_N is the cumulative (cum.) AE absolute (abs.) energy after N cycles. Note that these proposed relationships are only presumed to be valid for the stable elastic portion of the crack evolution.

Non-AE methods for determining crack size such as CMOD have also been developed for known materials and geometries. Nemati (2012) proposed a relationship for fixed-fixed support SE(T) specimens between the measured CMOD after *N* cycles, δ, and the ratio a/b:

$$\delta_N = \left(\frac{4 \cdot \Delta\sigma \cdot a_N}{E'} \right) \cdot V_f \left(\frac{a}{b} \right), \text{ and} \tag{21}$$

$$V_f\left(\frac{a}{b}\right)=\left(\frac{1.46+3.42\cdot\left(1-cos\frac{\pi\cdot a}{2\cdot b}\right)}{\left(cos\frac{\pi\cdot a}{2\cdot b}\right)^2}\right)\cdot\gamma\left(\frac{a}{b}\right), \text{ and} \quad (22)$$

$$\gamma\left(\frac{a}{b}\right)=1.01-0.13\cdot\left(\frac{a}{b}\right)+1.78\cdot\left(\frac{a}{b}\right)^2-8.02\cdot\left(\frac{a}{b}\right)^3+12.72\cdot\left(\frac{a}{b}\right)^4-7.68\cdot\left(\frac{a}{b}\right)^5. \quad (23)$$

This LEFM relationship is also valid only while the length of the uncracked ligament remains predominantly elastic.

Crack size estimation and life prediction models

Damage prognosis is typically defined as estimating a structure's remaining useful life, which is usually quantified as the number of cycles the structure can withstand before failure (Farrar et al. 2005). Unfortunately, estimating the remaining life typically requires knowledge of loading conditions, member geometry, and stress intensity factor thresholds. Several deterministic LEFM models have been proposed and are summarized by Kulkarni and Achenbach (2008). Probabilistic models based on grounds that crack-growth is an inherently stochastic process have also been developed. By modeling different sources of uncertainty such as imprecision in measurement, variability of LEFM parameters, or even systematic errors, the remaining life or other parameters can be output as ranges with a certain degree of confidence. Of particular relevance to this study are (Rabiei et al. 2009; Mohanty et al. 2011; Zárate et al. 2012a), who used, respectively, count, amplitude, and AE abs. energy measurements as part of probabilistic models for remaining life prediction. While probabilistic models are able to output probabilistic estimates of the damage evolution, they require the analyst to make simplifying assumptions in order to model the uncertainty, and they also require knowledge of a

model relating measured and predicted values. Of greater importance, however, is the computational expense associated with probabilistic models, which are populated using Monte Carlo methods that require hundreds of millions of iterations before reaching "convergence" (Zárate et al. 2012b).

An alternative to closed-form deterministic LEFM models and stochastic methods is that of ANNs, which have been used extensively in fatigue damage prognosis and have been shown to be effective in modeling material behavior, particularly when given sufficient training data (Flood and Kartam 1998). Early work by Han (1995) on the fatigue life of weldments with weldment defects found that ANNs were effective in predicting the remaining weldment life with 50% accuracy. Pleune and Chopra (2000) found ANNs useful in characterizing the fatigue lives of carbon and low alloy steels as a function of steel type and its environment. Srinivasan et al. (2003) found ANNs to be superior to analytical approaches when relationships between LEFM input and output variables are unknown while maintaining life estimation accuracy within 50%. Genel (2004) studied the applicability of ANNs to the prediction of strain-life fatigue properties compared to analytical and empirical methods and concluded that ANNs can adequately capture these relationships. More recently, Mathew et al. (2008) used ANNs to model the relationship between temperature and nitrogen-alloyed 316L stainless steel LEFM parameters and found their network could predict the fatigue life with 50% accuracy.

Steel fatigue-fracture research using ANNs with AE parameters as inputs has been particularly scarce. Hill et al (1993) used AE amplitude inputs to predict the ultimate strength of welds within 3% accuracy. Emamian et al. (2003) employed ANNs as a classification tool, using the principal components of AE fatigue data as inputs to the

network and classifying data as “crack-related” or “noise-related” with 90% accuracy. Kim et al. (2004) fed AE counts, energy, rise time, duration, and amplitude measurements averaged for every $4x10^{-3}$in. crack size increment into an ANN in order to correlate it with the SIF range; this study found that ANNs could fit experimental SIF data with a coefficient of determination, $r^2 \geq 0.62$, for all tested specimens. Lastly, Barsoum et al. (2009) utilized an ANN in the prediction of overall fatigue life using AE count data for less than half of the specimen’s life and reported prediction errors lower than 12%.

ANN design for time-series forecasting

Although all ANNs follow the same underlying principles, there are many network designs tailored to different applications. A broad definition of a practical ANN is that it is a collection of interconnected neurons that incrementally learn from data to capture essential linear and nonlinear trends and relationships in complex data, so that it provides reliable predictions for new situations containing even noisy or partial information. The fundamentals of ANNs are well understood, so this section will only aim to summarize the design and training procedure. Additional information regarding ANN fundamentals, design, and training may be found in (Samarasinghe 2007).

Although the complex relationship between AE and LEFM has been examined in the literature, little effort has been made to make use of the temporal relationships that are inherent in fatigue-fracture applications. If it is assumed that the loading conditions in a structure remain constant over time, as is the case in constant-amplitude fatigue testing, then crack-growth can be modeled as a time series. It can be postulated that under these

loading conditions, the crack size at time *t*, a_t, may be a function of one or several of its previous values so that,

$$a_t = f(a_{t-T}, a_{t-2\cdot T}, ..., a_{t-n\cdot T}), \tag{24}$$

where T is a user-defined time-step representing the prediction resolution, and $a_{t-n\cdot T}$ is the n^{th} and earliest value of *a* contributing to a_t. Such a model is considered to be autoregressive as it depends only on its past values. Similarly, it is possible to introduce an exogenous variable that also contributes to the value of *a*, so that:

$$a_t = f(a_{t-T}, a_{t-2\cdot T}, ..., a_{t-n\cdot T}, x_{t-T}, x_{t-2\cdot T}, ...x_{t-n\cdot T}), \tag{25}$$

where x is an exogenous variable sampled over time. This model, called *autoregressive exogenous*, is typically used, as the name implies, in applications where an autoregressive time series is also influenced by external inputs. Lastly, it can be assumed that a_t depends only on previous values of the exogenous variable, x, yielding the following input-output expression:

$$a_t = f(x_{t-T}, x_{t-2\cdot T}, ...x_{t-n\cdot T}). \tag{26}$$

While these relationships can serve as a basis for an ANN study directly relating crack size to itself and other exogenous inputs, in practice, an accurate crack size time history is rarely known in field or even experimental applications. Since knowledge of the crack size history is required in order to train the crack size models, in most cases, it may be difficult to train an ANN directly for crack size predictions. Instead, prediction of NDE-type measurements, such as AE or CMOD, may be more feasible and representative of practical applications where the actual crack size is unknown. If NDE

measurements are able to be predicted with accuracy, then the crack size at each time step can be estimated using the relationships presented in the background section. This hybrid ANN and LEFM model can be represented by:

$$a_t = \phi(y_t), \text{ and } y_t = f(y_{t-T}, y_{t-2\cdot T}, \ldots y_{t-n\cdot T}, x_{t-T}, x_{t-2\cdot T}, \ldots x_{t-n\cdot T}), \quad (27)$$

where $\phi(\,.\,)$ is a model relating crack size to an NDE measurement, y, and $f(\,.\,)$ is an ANN relating this NDE measurement to its time history and an exogenous NDE measurement's, x, time history.

In this study, a nonlinear autoregressive exogenous input (NARx) ANN is used to model the relationship between NDE measurements and their future values. The NARx ANN consists of one input neuron layer, one nonlinear hidden neuron layer, and one linear output neuron. The ANN design is shown in Figure 21.

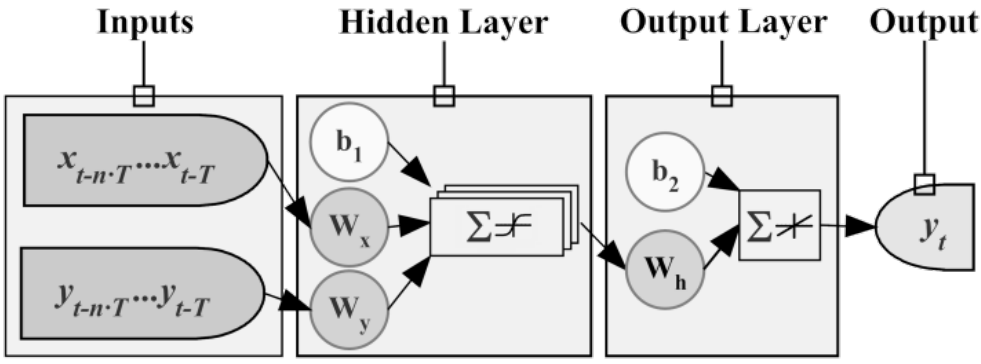

Figure 21 – Schematic for NARx ANN design for one-step-ahead prediction

This design offers the advantage of having only one hidden layer with a user-specified number of neurons, making training significantly faster. Before training this ANN, the user must fix the number of delays, n, the prediction resolution, T, and the number of neurons in the hidden layer, N_Σ,—all of which are positive integers. Prediction of a time-

step $t + k \cdot T$ can be obtained directly from a single NARx ANN by modifying the model such that:

$$a_{t+k_p \cdot T} = \phi(y_{t+k_p \cdot T}), \text{ and } y_{t+k_p \cdot T} = f(y_{t-T}, y_{t-2 \cdot T}, ...y_{t-n \cdot T}, x_{t-T}, x_{t-2 \cdot T}, ...x_{t-n \cdot T}), \quad (28)$$

where k_p is a positive integer denoting the prediction horizon. This relationship maintains the assumption that there is enough information in the variables' time histories to adequately predict the output variable this far in advance. The NARx network is able to be used without either an exogenous or autoregressive variable by simply ignoring all neurons associated with that input.

NARx ANNs, like all feed-forward ANNs, can be trained using typical back-propagation algorithms, although more efficient training can be achieved using the Lavenberg-Marquardt Algorithm (LMA) (Hagan et al., 1996). LMA is a variation of the Newton's method for minimizing functions that are sums of squares of other non-linear functions and can be written as follows:

$$\boldsymbol{W}_{n+1} = \boldsymbol{W}_n - \left(\boldsymbol{J}_n^T \boldsymbol{J}_n + \mu \boldsymbol{I} \right)^{-1} \boldsymbol{J}_n \boldsymbol{e}_n, \quad (29)$$

where n is the current iteration step, $\boldsymbol{W}$ is the network weight matrix, $\boldsymbol{J}$ is the Jacobian matrix, which is composed of the first derivatives of the network errors with respect to the weights, $\boldsymbol{I}$ is the identity matrix, and $\boldsymbol{e}$ is the error vector. The network weight matrix is randomized with small non-zero values (less than 0.5), and $\mu = 0.001$. During training, μ is decreased by a factor of 10 after each successful epoch (i.e., a reduction in the performance function) and is increased by the same factor only when a tentative step would increase this measure. In this way, the performance function will always be

reduced in each training iteration. In this study, the performance function, *E,* is chosen to be the sum square error (SSE). Lastly, the activation function chosen for the hidden layer is the sigmoid hyperbolic tangent function, and all outputs are scaled using a linear function output neuron.

Experiments and discussion

Single edge notches provide well-defined load and fatigue crack size and shape environment for estimation of LEFM parameters. However, typical ASTM E647 (2011) specimens require AE sensors to be placed on the surfaces containing the crack tip. These sensor configurations rarely represent field conditions for early crack-growth stages. To overcome this limitation, Nemati (2012) designed a small-scale specimen in order to develop uniform stresses perpendicular to the plane of crack-growth, while allowing for AE sensors to be placed in the same surfaces expected to be used in the field (Figure 22 (left)). Six narrow-band Physical Acoustics Corporation (PAC®) R15I-AST sensors are deployed around a pre-notch. Sensors are spaced with 4in. spacing in a linear array. The two innermost sensors are placed 4in. away from the notch. The final sensor arrangement is depicted in Figure 22 (right).

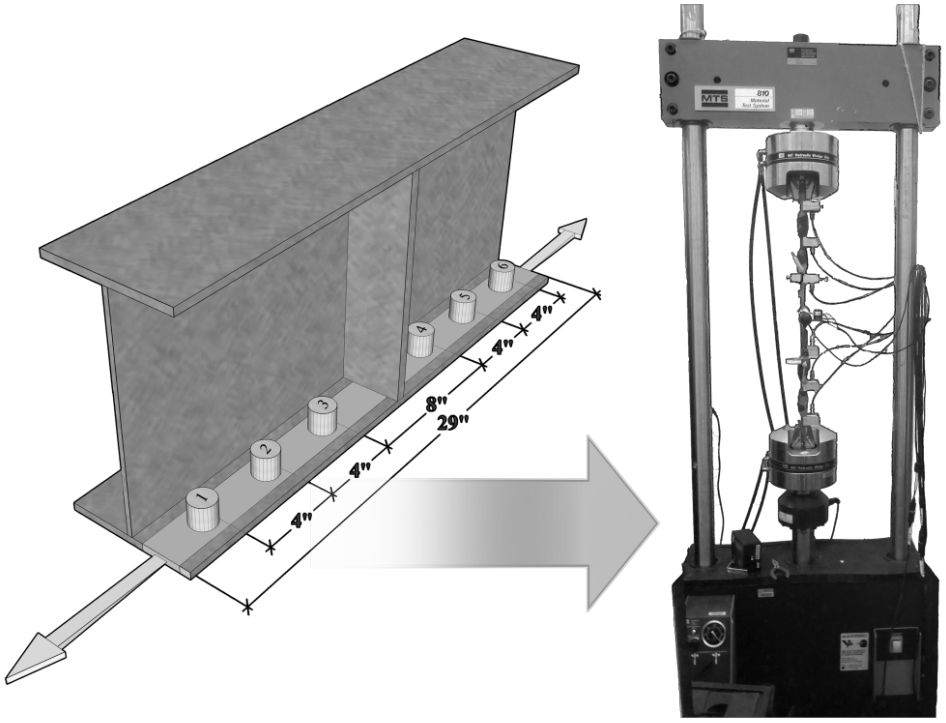

Figure 22 – Left: Schematic of field deployment of AE sensors around welded stiffener; right: Experimental setup of small-scale specimen designed from field setup

In addition to AE sensors, two 200X-magnification video cameras are mounted in order to visually monitor and measure the crack-tip evolution in real time. Crack size measurements are recorded from the magnified images using image-processing software. A displacement gauge is placed at the notch's mouth and the CMOD is measured at a sampling frequency of 50Hz. AE sensors are connected to an MTS® data acquisition system (DAQ), and AE sensors are connected to a PAC® *PCI-2 Rev. 3* AE DAQ. Specifics about experimental design and hardware setup parameters can be found in (Nemati 2012).

This study focuses on two ASTM A572-G50 SE(T) specimens with a nominal length of 29in and gross cross-sectional area of $1in^2$. The choice of dimensions and materials is meant to represent the typical design characteristics of flexural beam flanges in steel bridges. The first specimen, S4, was tested at a peak load, P_{max}, of 13kip and a minimum

load, P_{min}, of 8.5kip; thus its R ratio, $P_{\min} / P_{\max} \approx 0.65$; and, the pre-notch length, a_0, was measured to be 0.170in. The second specimen was subjected to a P_{max} of 10.5kip, a P_{min} of 1.0kip ($R \approx 0.1$), and a pre-notch $a_0 = 0.105$in. The tensile load was cycled sinusoidally at a frequency of 2Hz.

Data preprocessing

AE testing is particularly fraught by the measurement of unwanted signals (sometimes loosely referred to as "noise"), even in controlled testing environments. There are many techniques used in order to ameliorate the background "noise" experienced during metal fatigue testing. Usually, setting high-amplitude acquisition thresholds helps to remove a large portion of the background signals. Surface dampening using modeling clay was found to be particularly useful at removing a significant portion of the "noise" coming from the hydraulic pump and grips. Band-pass filters are also put in place to filter out signals with very low (<20kHz) and very high (>300kHz) frequencies. Even with these precautions, background signals are still a problem. Post-processing noise removal is accomplished using the outlier detection rules found in study 2. This methodology of noise removal is useful in removing unwanted signals unrelated to crack-growth phenomena. The AE datasets for both specimens were filtered so that only signals related to crack-growth are present. Although AE parameters can be forecast for every sensor, this study focuses only on the sensor closest to the pre-notch, as this sensor is expected to detect the highest number of crack-growth-related emissions; moreover, the choice of a single sensor is done to highlight that crack-growth forecasting can be performed in situations where sensor arrays are sparse or only one sensor is deployed.

Data collected by the CMOD gauge is preprocessed in two steps. First, peak ($\delta_{\max}$) and minimum ($\delta_{\min}$) CMOD readings for each loading cycle are each exported to a column vector. The two vectors are then subtracted from one another and the results are assigned to column vector $\boldsymbol{\delta}$. This vector represents the net CMOD as a function of the number of cycles. A locally weighted regression smoothing (LOESS) filter is applied to the ensuing vector using a span coefficient of 1%. This smoothing filter ensures that periodic fluctuations of the CMOD readings are replaced by a regressed line indicating the overall growing trend of the CMOD with increased number of cycles.

While 15 features and all six sensors in the AE dataset are used in the background "noise" filtering step, this study is only concerned with the time history of the AE abs. energy, u_{AE}, which is defined as the integral of the squared rectified voltage signal divided by the reference resistance (10kΩ) over the hit duration. AE abs. energy readings are measured in attoJoules (aJ). This measurement is assumed to be directly proportional to the total elastic energy released by crack-growth-related phenomena, U_a, such that:

$$U_N = \sum_{i=1}^{N} \boldsymbol{u}_{AE_i} = \psi \cdot U_a, \tag{30}$$

where ψ is an energy scaling parameter, and U_N is the cumulative (cum.) sum of the $\boldsymbol{u}_{AE}$ vector over N cycles. One problem of the raw U vs. N plot is that it is highly sensitive to outlier AE "hits" of very high energy. These types of AE hits are speculated to occur due to crack-surface fretting events that cause measured waveforms during this time-frame to have disproportionally high energies—sometimes orders of magnitude larger than the neighboring hit distribution. In order to address this problem, a trimmed mean outlier

removal method was employed. For a given user-specified window of time, the trimmed mean of the $\boldsymbol{u}_{AE}$ vector, $\breve{u}_{AE}$, can be defined as:

$$\breve{u}_{AE} = \frac{1}{\left|\boldsymbol{u}_{AE}\right| - 2k_r} \sum_{i=k_r+1}^{\left|\boldsymbol{u}_{AE}\right|-k_r} \boldsymbol{u}_{AE\,i:\left|\boldsymbol{u}_{AE}\right|} \text{ and } k_r = floor\left(\frac{\left|\boldsymbol{u}_{AE}\right| \cdot t_r}{2}\right), \tag{31}$$

where $\left|\,.\,\right|$ is the cardinality operator, t_r is the trimming percentage factor, and $\boldsymbol{u}_{AE}$ is sorted in ascending order, and *floor*(.) maps a number to the largest previous integer. As the name suggests, this method removes a certain proportion of the extreme order statistics from both tails of a distribution. Lehmann (1998) showed that the trimmed mean with $t_r \approx 0.1$ is a safe factor for long-tailed data, as is the case with AE abs. energy. Another example of a trimmed mean is the sample median, which is obtained by setting $t_r = \left(\left|\boldsymbol{u}_{AE}\right| - 1\right) / \left|\boldsymbol{u}_{AE}\right|$.

For each minute of AE acquisition, both the sample median, the trimmed mean ($t_r = 0.1$), and the conventional mean ($t_r = 0$) are calculated from the AE abs. energy measurements. The cum. AE abs. energy, u, for that time period is approximated by:

$$u = \breve{u}_{AE} \cdot \left|\boldsymbol{u}_{AE}\right| , \tag{32}$$

and the total cum. AE abs. energy, U_N, is calculated by summing all the values in the $\boldsymbol{u}$ vector after N cycles. Because the distribution for the AE abs. energy parameter is particularly long-tailed, higher values of t_r will result in reduced slopes in the U vs. N curves, slightly underestimating dU/dN. The final U and CMOD vectors that are passed into the ANN are sampled at a minimum resolution of 1 minute, equal to the unit of time between each trimmed mean AE abs. energy value.

The benefit of using a trimmed value for the AE abs. energy is readily seen when fitting the a vs. U curve to the $a = \beta(U)^{\alpha} + a_0$ model. The values for β and α can be estimated using a typical robust least-squares regression technique (Andersen 2009). The coefficient of determination r^2 is used as a measure of wellness of fit. The best-fit model parameters for both specimens are summarized in Table 2.

Table 2 – a vs. N best fit parameters

specimen	***ŭ* type**	***α***	***β***	r^2
S4	t_r=0	4.019e-04	0.4104	0.9901
S4	t_r=0.1	3.169e-04	0.4398	0.9923
S4	median	2.628e-04	0.4801	0.9948
S9	t_r=0	3.642e-07	0.9322	0.9527
S9	t_r=0.1	2.914e-07	0.9431	0.9551
S9	median	2.428e-07	0.9725	0.9575

In all regressions, the best fit was achieved using the sample median for the U calculation. As expected, the estimated values for α and β are, respectively, smaller and larger for each increase of t_r. It should be noted that the wellness of fit is mostly improved when using the median measure in late stages of crack-growth and the rate of energy release is most unstable. For this reason, the sample median is used as the preferred trimmed mean type in this study. A plot of the a vs. U (obtained from the sample mean AE abs. energy) curves for both specimens and the best-fit models can be seen in Figure 23.

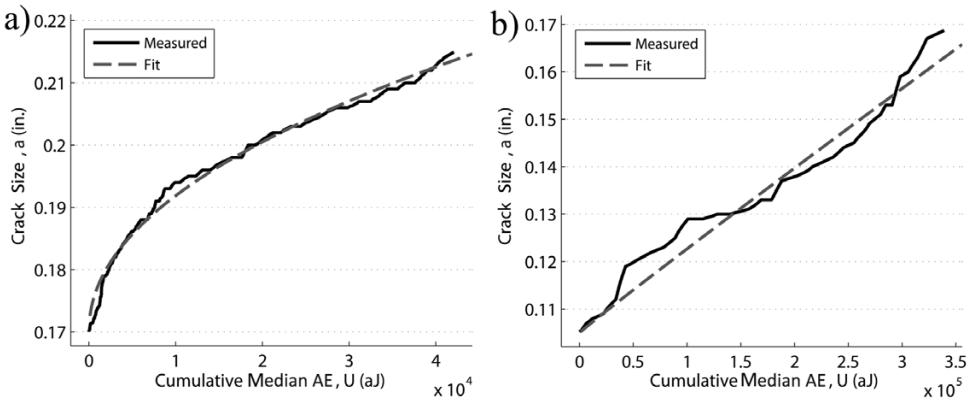

Figure 23 – Measured and best-fit crack size vs. cum. median AE abs. energy (sensor 3): a) specimen S4; b) specimen S9

As is suggested by Nemati (2012), the wellness of fit of AE abs. energy models seems to be adversely affected under low *R* ratios. The model parameters obtained from this fit are later used in the estimation of the crack size from predicted values of *U*. Similarly, predicted values for the CMOD are input into the expression relating *a* and δ presented in section 2.1 in order to obtain a crack approximation. A plot of the measured and predicted crack sizes for the valid-reign of elastic crack-growth is presented in Figure 24.

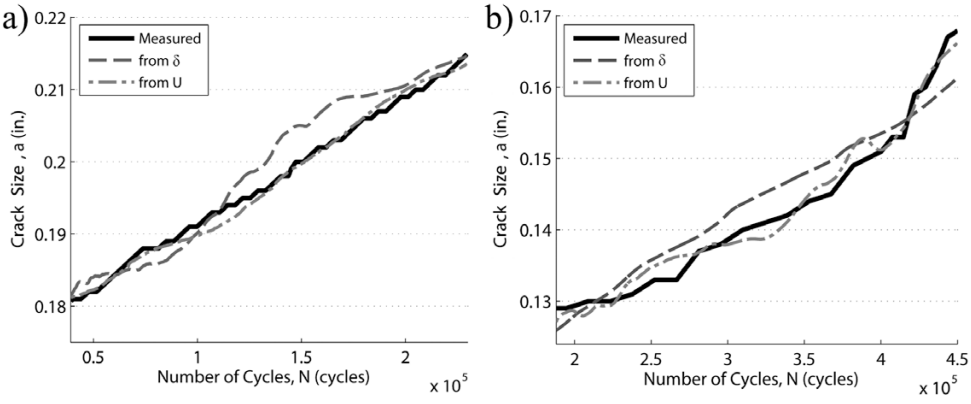

Figure 24 – Measured and estimated crack size (from measured values) vs. number of cycles: a) specimen S4; b) specimen S9

As can be seen, crack estimation using the CMOD measurements exhibits a greater error than the approximation from U. The crack size values estimated from U are fit to the measured crack size for each specimen, so they should naturally exhibit much lower errors assuming the fit is adequate.

ANN training and testing

The ANN is prototyped using the *MATLAB® Neural Net Time Series* toolbox. Once the ANN design parameters are fixed, the network is implemented in *C++* using the *Heaton Research, Inc.® Encog Machine Learning Framework*. Calculations using *Encog* are parallelized using a hybrid *OpenMP/OpenMPI* approach over 128 cores. Both frameworks allow for feed-forward ANN design, leave-one-out cross-validation, and LMA training, though *Encog* was found to be appreciably more efficient.

When training the ANN, its design parameters are required to be tuned by the analyst until a design with the lowest performance function value is found. In this case, the performance function used when comparing different ANN designs is the mean abs. percentage error (*MAPE*), as it measures the average deviation from the target for all validation and testing instances. The chosen NARx ANN has three design parameters that must be varied until the best-fit model is found. The first parameter to be optimized is the number of lags represented by the tapped delay line (i.e., how far back will each input variable draw values from). Generally, the number of lags should be enough to account for autocorrelation of the input variables; however, increasing the number of lags will result in added delays and increased computational power requirements. A good estimate of the number of required lags to consider in the tapped delay line is obtained by

examining the input autocorrelation, r_l, which is simply a normalized comparison of the original and shifted time series for each input variable, given by:

$$r_l = c_l / c_0 \text{ and } c_l = \frac{1}{n_{\max}} \sum_{t=1}^{n_{\max}-l} (x_t - \bar{x})(x_{t+l} - \bar{x}) \ , \tag{33}$$

where x is the training variable being considered, $\bar{x}$ is the variable's sample mean, c_l is the autocovariance at lag l, c_0 is the sample variance, and $n_{\max}$ is the number of training objects. It is typically necessary to account for all autocorrelations outside the confidence limits, which are calculated as $\pm 2/\sqrt{n_{\max}}$. A sample autocorrelation for the CMOD input for specimen S4 can be seen in Figure 25a.

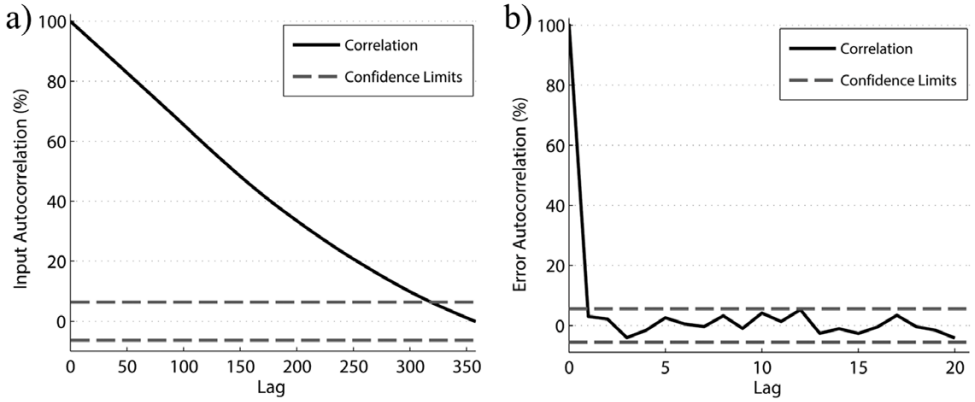

Figure 25 – Sample autocorrelation plot (specimen S4 (sensor 3)): a) input autocorrelation; b) error autocorrelation

In this autocorrelation plot, the first 320 lags are significantly positively correlated. This autocorrelation plot suggests that the ANN model should take into account delays up to 320 minutes (or greater). The most autocorrelated variable was found to be the *U* input variable for specimen S9, which showed significant autocorrelation up to 625 lags. Generally, the number of lags, which is defined as the prediction resolution, *T*, times the

number of delays, n, must be greater than the largest significantly autocorrelated lag in any of the input variables.

In order to verify the optimal number of lags, the number of neurons is fixed to 100, the prediction resolution is kept at its minimum, and the delay value is varied until the performance measure is minimized. The network is trained and validated using the entire time series (with leave-one-out cross-validation to prevent overfitting), a minimum resolution of one minute, and 100 neurons in the hidden layer. The optimal number of lags is found to be equal to 675, and the minimum *MAPE* is found to be equal to 9.4% for this design. Increasing delays to include lags past this point resulted in higher errors.

Next, the number of neurons must be optimized. This is a very important parameter in ANN design, as it dominates the predictive ability of the model and is directly related to the computational time required to train the model. As a rule of thumb, increasing the number of neurons will minimize the *MAPE* and further increases will result in diminishing returns on this measure. The prediction resolution is kept at a minimum, and the number of delays at this resolution is determined based on the optimal number of lags found above. The number of neurons is increased from 1 to 500 in 25 neuron increments. The optimal number of neurons is found to be 275, with a *MAPE* of 5.03%, and a training time of 24 seconds.

The prediction resolution, T, is the user-defined time-step between delays and predictions and, as previously mentioned, is related to the optimal number of lags. Smaller values for T will result in predictions closer in time at the expense of increasing the number of delays for each variable. A greater number of delays will cause a proportionally larger increase in training time. The optimal prediction resolution, thus,

should be a compromise between the desired granularity of the crack-prediction, the error associated with this prediction resolution, and the available computing power. In this study, a minimum prediction resolution is not required, so an optimal *T* may be found by specifying a maximum tolerated error of 5.0%, and choosing the largest value of *T* under this error. The maximum tolerated error is meant to represent a confidence threshold of 95%. With this procedure in mind, *T* is varied from 1 to 60 minutes in steps of 5 minutes, and the *MAPE* is, again, used as the performance measure. A plot of the *MAPE* vs. *T* can be seen in Figure 26.

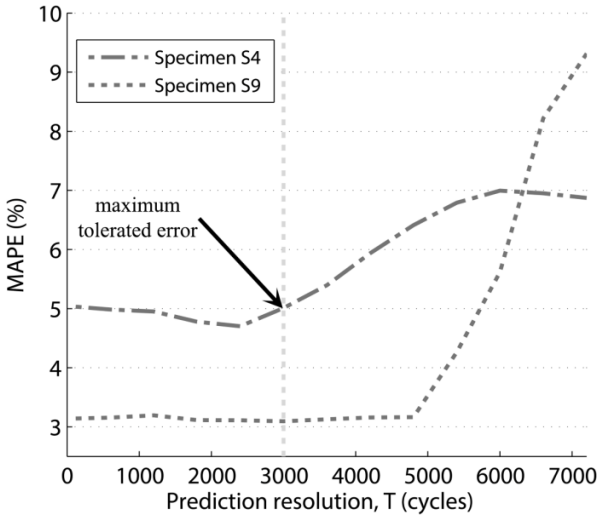

Figure 26 – *MAPE* vs. prediction resolution for specimens S4 and S9 for autoregressive exogenous model

The largest value of *T* corresponding to a $MAPE = 5.01\%$ for both specimens is found to be 25 minutes (3000 cycles).

Once all the ANN parameters are found, the error time series is checked for autocorrelation. If the error is significantly autocorrelated, the ANN design must be modified in order to ensure that the error terms at each time-step are independent of one another and systematic error due to autocorrelated terms has been removed. The error

autocorrelation for all the plots was consistently below the significance threshold for all predicted outputs. Figure 25b presents a sample plot of the error autocorrelation for specimen S4. The final network design is thus fixed at 275 neurons, 675 lags (27 delays), and a prediction resolution of 25 minutes (3000 cycles).

Once the ANN design is fixed, the network is tested for prediction using the following input-output combinations:

1) **Inputs:** Cum. AE abs. energy (U) → **Output:** CMOD (δ)
2) **Inputs:** Cum. AE abs. energy (U) and CMOD (δ) → **Output:** CMOD (δ)
3) **Input:** Cum. AE abs. energy (U)→ **Output:** Cum. AE abs. energy (U)

The NARx model is able to accommodate all three of these combinations. When the external variable is not used, the associated neurons and weights for the exogenous input are not considered in the training process. The ANN is tested using a training and validation period of 675 minutes (i.e., 81,000 cycles) (in order to accommodate all of the delays), and one instance per minute (i.e., one instance per 120 cycles). The testing horizon is increased over several time steps, and the ANN is retrained at each increase. The *MAPE* and *MRAE* errors, run times, and maximum prediction horizons for all tests are summarized in Table 3.

Table 3 – ANN testing results summary

specimen	ANN input(s)	ANN output	*MAPE (%)*	*MRAE (%)*	*pred. horizon (cycles)*	*run time (s)*
S4	U	δ	6.3	13.0	+45,000	345
S4	U, δ	δ	5.0	9.2	+45,000	748
S4	U	U	5.3	12.5	+45,000	331
S9	U	δ	7.4	23.3	+201,000	1,610
S9	U, δ	δ	3.1	12.0	+201,000	3,477
S9	U	U	4.3	11.9	+201,000	1,590

Predicted values for specimen S4 are shown in Figure 27 for each input-output combination.

A plot *MRAE vs. number of cycles* is shown on Figure 27d in order to visualize the error evolution over the multi-step prediction. Figure 28 shows similar plots for the predicted values for specimen S9.

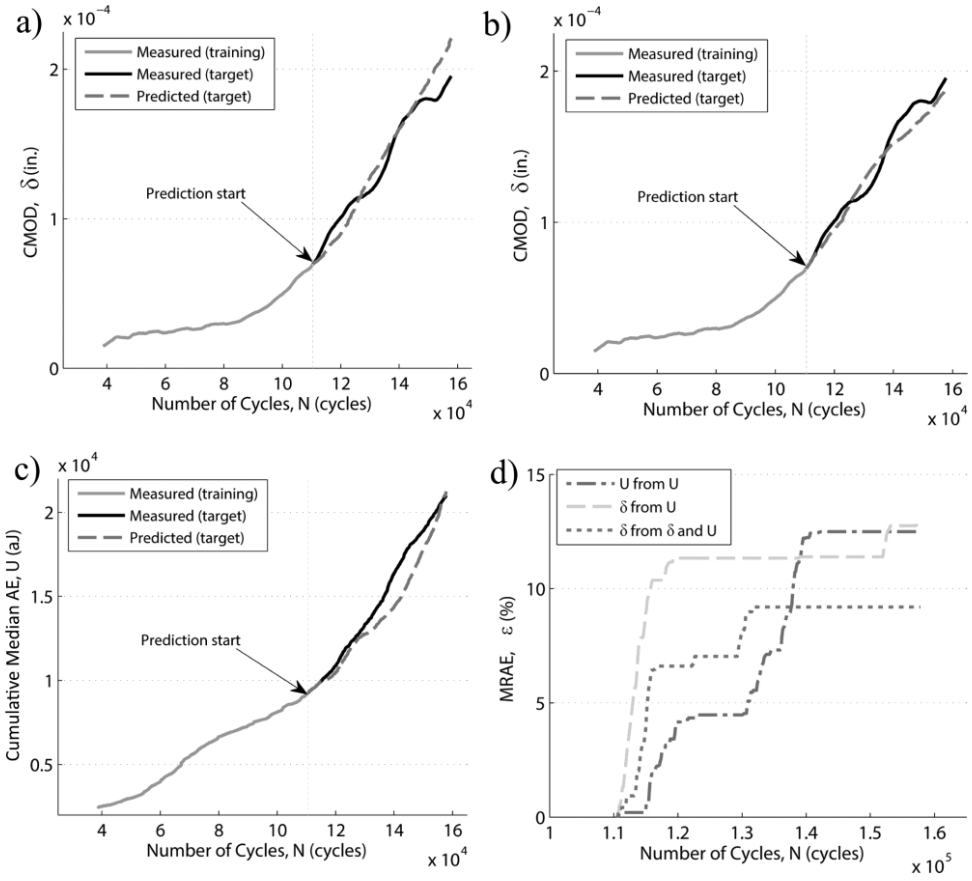

Figure 27 – Specimen S4 (sensor 3) target forecast vs. number of cycles: a) *U* input and *δ* output; b) *U* and *δ* inputs and *δ* output; c) *U* input and *U* output; d) *MRAE* for each forecast

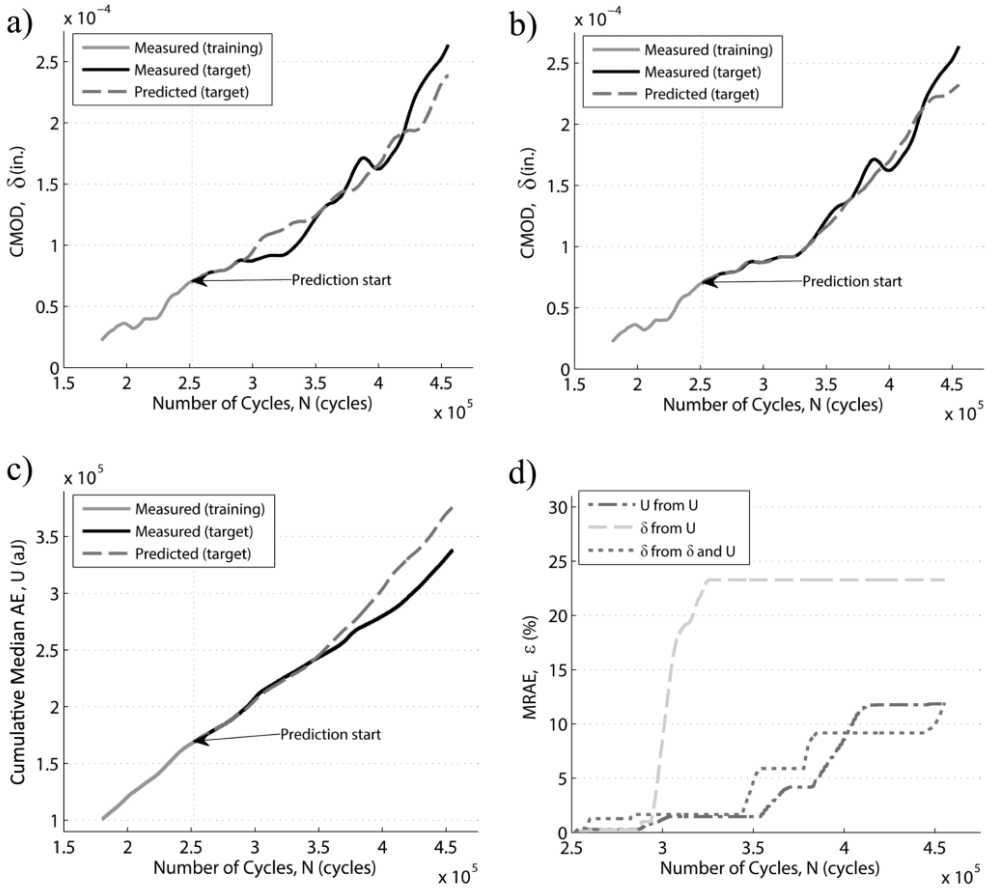

Figure 28 – Specimen S9 (sensor 3) target forecasts: a) *U* input and *δ* output; b) *U* and *δ* inputs and *δ* output; c) *U* input and *U* output; d) *MRAE* for each forecast

Once the AE and LEFM parameters are predicted, they can be applied as inputs to the LEFM models presented in the background section. The CMOD model used in this study is applicable only to SE(T) specimens under fixed-fixed conditions. The cum. AE abs. energy model is presumed to work for different geometries, loading conditions, and materials. However, the model parameters seem to be particularly sensitive to the loading conditions and, at this stage, have to be fit for each specimen. Estimation of the crack size evolution will, therefore, be dependent on the accuracy of the applied models. A summary of the *MAPE* and *MRAE* values for each specimen and input-output

combination is presented in Table 4; moreover, the *MAPE* and *MRAE* values for the average predicted crack size is calculated for each specimen.

Table 4 – LEFM model errors from predicted inputs

specimen	ANN input(s)	ANN output	LEFM model	*MAPE* (%)	*MRAE* (%)
S4	*U*	δ	$\delta \rightarrow a$	1.9	3.1
S4	*U*, δ	δ	$\delta \rightarrow a$	1.7	2.5
S4	*U*	*U*	$U \rightarrow a$	0.8	1.6
S4 (pred. avg.)				**1.0**	**1.7**
S9	*U*	δ	$\delta \rightarrow a$	3.2	6.9
S9	*U*, δ	δ	$\delta \rightarrow a$	3.5	6.4
S9	*U*	*U*	$U \rightarrow a$	2.2	4.9
S9 (pred. avg.)				**1.4**	**4.5**

Finally, a plot of all estimated *crack size* vs. *number of cycles* for specimens S4 and S9 is shown in Figure 29.

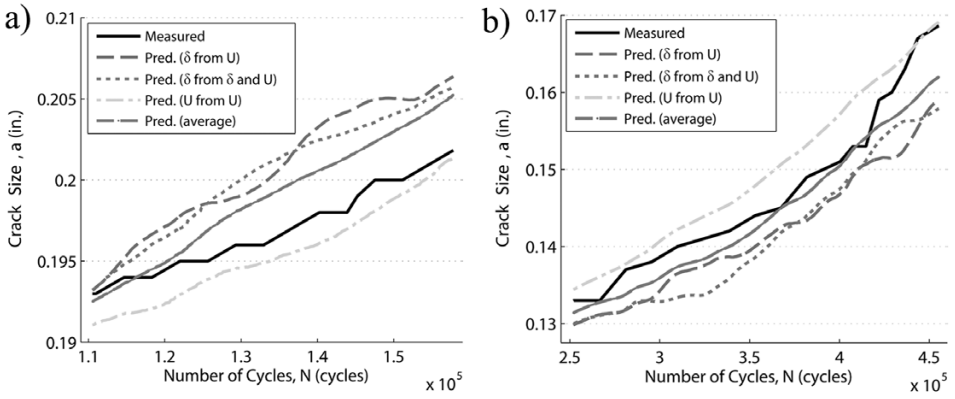

Figure 29 – Measured and estimated crack size vs. number of cycles (from predicted values): a) specimen S4; b) specimen S9

Discussion

One way of evaluating the performance of any forecast is to qualitatively examine each of the time systematic components of the predicted time series: level, trend, and periodic variation. Level describes the average value of the series, trend is the change in the series from one step to the next, and periodic variation describes the short-term cyclical behavior of the cycles. For each forecast, the model's ability to capture each of these components is examined.

The forecast values for specimen S4 adequately exhibit the level and trend of the measured target values; however, the larger periodic fluctuation of the CMOD is not present in the forecast. The model's inability to predict types and magnitudes of fluctuations not previously observed can be attributed to the relative lack of fluctuation present in the training dataset compared to the target measurements. In other words, the forecast will generally not exhibit large fluctuations if the training dataset varies only slightly around the trend line. For this specimen, the exogenous autoregressive model offers the best quantitative measures (i.e., the lowest errors) for long-term forecasting with a *MRAE* under 9.2% for up to 45,000 cycles. For short-term forecasting, the autoregressive model predicted the measured values for up to 18,000 cycles with a *MRAE* under 5.0%.

Forecast for specimen S9 was particularly accurate, exhibiting a very close adherence to the time series components of the measured values for up to 48,000 ahead with a *MRAE* under 2.5% for all 3 models. Long-term forecasting behavior exhibited similar level and growing trend, although prediction accuracy of the periodic fluctuation quickly degenerates; nonetheless, the predicted values display a fluctuation of the order of

magnitude of the training data. The forecasts for the autoregressive and exogenous autoregressive models are able to predict well into the future, with a *MRAE* smaller than 12.0% for both of these models up to 201,000 cycles ahead. For short-term forecasting, the best model was, again, the autoregressive one, with a *MRAE* smaller than 2.5% up to 102,000 cycles ahead.

Crack estimates depend on the model accuracy and, as such, the overall crack size and crack-growth rate will depend directly on the model parameters. For specimen S4, the estimated crack levels using the predicted CMOD values tend to overestimate the crack size and crack-growth rates. The values estimated from the predicted AE abs. energy values similarly overestimate the crack-growth rate, although the crack size is slightly underestimated. Crack size estimates for specimen S9 were overestimated when using the CMOD predicted values and underestimated when using predicted AE abs. energy values. The crack-growth rate for all predictions for specimen S9 agreed with the measured values. For both specimens, the average of all predicted crack sizes was found to provide both good short-term and long-term estimates. The lower *MAPE* and *MRAE* values for the estimated crack sizes suggests that using inputs with large errors in LEFM models translates into much smaller errors in the LEFM models, provided that they adequately represent the relationship between the input values and the crack size.

Concluding remarks

In this study, the formulation, design, and validation of a deterministic crack-growth forecasting framework was presented, as applicable to structural elements subjected to cyclic loading using AE measurements. This framework predicts the short-term and long-term values for cum. AE abs. energy and CMOD at different numbers of cycles during

elastic and stable crack-growth. The predicted values, then, become inputs to LEFM models in order to estimate the current and future crack sizes. The framework assumes that LEFM models for the structural element being monitored relating predicted values to the crack size are known *a-priori*. The presented LEFM model relating the cum. AE abs. energy and the crack size is found to adequately represent the crack evolution over the valid-reign of elastic crack growth. The framework is trained, validated, and tested using AE and CMOD data from two SE(T) specimens at different *R* ratios. AE and CMOD measurements are forecast using three different models for each specimen. The accuracy of the predicted values indicates that ANNs are able to adequately model autoregressive and exogenous relationships between AE and LEFM measurements and their future values. The *MAPE* for specimen S4's best model was found to be 5.0% for a 45,000 cycle-ahead prediction. For specimen S9, the lowest *MAPE* was measured to be 3.1% for a prediction 201,000 cycles ahead. In the case of long-term predictions, the autoregressive exogenous model was proven to better represent the measured values with *MRAE* values below 12.0%. For short-term predictions, the autoregressive model was found to be more effective, as it maintained a *MRAE* under 5% up to 18,000 cycles ahead. When estimating the future crack size, the average of all three models was found to be a good predictor with a *MAPE* less than 1.4% and a *MRAE* under 4.5% for all specimens.

Laboratory and field predictions of AE and LEFM measurements using this framework are immediately feasible; however, crack estimation depends entirely on choosing a model and assuming knowledge of the parameters. If parameters for an applicable model are not known for a particular field or laboratory application, they can

be estimated using a subset of previously-collected data, given that the crack history is known.

These frameworks are meant to form part of SHM applications, where knowledge of the current and future state of the structure is paramount. Once in place, systems utilizing this framework can incorporate predicted values as part of damage prognosis models. Moreover, once deployed, these frameworks can be used in warning systems in SHM systems, allowing bridge owners to be able to make repair decisions or limit traffic based on non-destructive crack-growth forecasts such as this.

CHAPTER 5, CONCLUSIONS AND RECOMMENDATIONS

Contributions of dissertation

This dissertation has been based on the premise that SHM practices are fundamental to the assessment and maintenance of current and future civil infrastructure. It has also been identified that computing practices are at the heart of modern SHM technologies, and it is essential to preface research efforts in this field by first identifying the required computing infrastructure.

The first study in this dissertation outlines the technological boundary conditions in SHM, while attempting to address the main computational challenges that must be overcome in order to allow for the acquisition, management, and analysis of SHM data. A methodology is proposed which provides a foundation for specifying computing requirements in small-scale SHM applications. The methodology is demonstrated through the implementation of a high-performance computing framework meant to handle large datasets arising from AE monitoring laboratory and field experiments. This computing framework is critical to the development of new SHM techniques, which rely primarily on knowledge discovery through high-performance computing.

In the second study, focus shifts to the challenges present in AE testing, which is chosen in this dissertation as a technology that is representative of data-intensive SHM practices. The study proposes an unwanted signal filtering methodology as part of a data mining scheme. In addition to filtering unwanted signals, the scheme allows for the

extraction of characterization rules for both relevant and unwanted AE signals. It is intended that this data mining scheme be used as tool for finding general "signatures" of specific failure mechanisms. Once these "signatures" have been identified, they can be used in SHM systems to filter out unwanted AE activity and serve as a warning system in the presence of damage-related signals. The enhanced data quality achieved through use of this scheme may also be especially important when assessing current and expected future damage.

The third and final study in this dissertation demonstrates the importance of good quality datasets and a proper computing framework, as they allow for the accurate and computationally efficient forecast of crack-growth damage in steel elements subject to fatigue fracture. The clean datasets obtained from the second study are used as inputs to an ANN that, when used in conjunction with traditional LEFM measurements, can provide accurate predictions of crack conditions in laboratory and field applications. The proposed methodology can be used in laboratory and field settings to forecast future measurements. In the presence of adequate models, predicted measurements can be reliably linked to current and expected damage conditions.

Future work

The outcomes of this research offer an important advancement to SHM, with particular benefits to AE testing. However, further research has been identified in critical areas that could improve the methodologies in this dissertation:

- Design of a database system capable of supporting real-time loading and data mining would streamline the flow of SHM data. Identifying a data management system that integrates dissimilar datasets and allows for fast retrieval and

visualization of information is central to efforts of real-time decision-making in SHM. In the context of AE testing, allowing for the acquisition of data directly to a remote database with access to a computing engine would enable the techniques described in this research to be performed in situ.

- A performance and computing efficiency analysis of statistical, nearest neighbor, clustering, classification, and spectral decomposition-based data mining techniques could result in more accurate characterization of AE signals and produce more reliable filters. Current and improved characterization methodologies can be used in the generation of damage "signatures" for common civil engineering materials such as reinforced concrete, steel, and fiber-reinforced polymers.
- Exploration of alternative ANN forecasting strategies may lead to better damage predictions. In particular, serially-chained NARx configurations could lead to faster training times and enable multistep prediction without need for re-training. Similarly, multiple-input-multiple-output (MIMO) forecasting methods could produce more accurate forecasts in a single training step.
- The methods presented in this thesis could be easily adapted to a variety of other AE SHM applications such as: the detection of the onset of corrosion in RC, improvement of AE source location techniques, and the unsupervised assessment of structures during structural load testing.
- Robust assessment and prognosis applications should incorporate reliability and uncertainty models, though particular emphasis should be placed in efficient

computation techniques that would enable these types of analyses to be done in real time in both field and laboratory settings.

Final remarks

It is hoped that this dissertation provides entrants to the field of SHM with a good foundation for discerning the boundary conditions in data-intensive SHM applications. Furthermore, this dissertation is expected to be used as a tool for AE practitioners and provide them with advanced data quality enhancement and signal characterization tools. Finally, this work aims to serve as a demonstration that a proper computing framework, in conjunction with clean data, can be used in the implementation of knowledge discovery and prediction tools, which are expected to be fundamental in bringing SHM closer to the mainstream.

WORKS CITED

Adams, N. J. I. (1972). "Fatigue crack closure at positive stresses." *Engineering Fracture Mechanics*, 4, 543-554.

Aktan, A. E., Catbas, F. N., Grimmelsman, K. A., and Tsikos, C.J. (2000). "Issues in infrastructure health monitoring for management." *Journal of Engineering Mechanics*, 126(7), 711-724.

Amdahl, G. M. (1967). "Validity of the single processor approach to achieving large scale computing capabilities." *Proceedings of American Federation of Information Processing Societies Spring Joint Computer Conference*, Atlantic City, 483-485.

Andersen, R. (2009). "Nonparametric methods for modeling nonlinearity in regression analysis." *Annual Review of Sociology*, 35, 67-85.

ASCE. (1982)."Fatigue reliability: Introduction." *Journal of the Structural Division*, 108(1), 3–23.

ASTM Standard E976-10. (2010). *Standard guide for determining the reproducibility of acoustic emission sensor response.* American Society for Testing and Materials (ASTM) International, West Conshohocken, PA.

ASTM Standard E647-11. (2011). *Standard test method for measurement of fatigue crack growth rates.* ASTM International, West Conshohocken, PA.

Barsoum, F. F., Suleman, J., Korcak, A., and Hill, E. V. (2009). "Acoustic emission monitoring and fatigue life prediction in axially loaded notched steel specimens." *Journal of Acoustic Emission*, 27, 40-63.

Beattie, A. G. (1997). "Acoustic emission monitoring of a wind turbine blade during a fatigue test." *35th American Institute of Aeronautics and Astronautics (AIAA) Aerospace Sciences Meeting and Exhibit* , Reno, 1-10.

Berkhin, P. (2002). *Survey of clustering data mining techniques*. Accrue Software Inc., San Jose.

Berkovits, A. and Fang, D. (1995). "Study of fatigue crack characteristics by acoustic emission." *Engineering Fracture Mechanics*, 51(3), 401-416.

Bhat, C., Bhat, M. R., and Murthy, C. R. L. (2003). "Acoustic emission characterization of failure modes in composites with ANN." *Composite Structures*, 61, 213-220.

Bray, D. E. and Stanley, R. K. (1997). *Nondestructive evaluation: A tool in design, manufacturing, and service.* CRC Press, Boca Raton.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). "LOF: Identifying density-based local outliers." *Association for Computing Machinery Special Interest Group on Management of Data (ACM SIGMOD) 2000 International Conference*, Dallas, 93-104.

Bruijn, N. G. d. (2010). *Asymptotic methods in analysis.* Dover Publications, Amsterdam.

Cannataro, M., Talia, D. and Srimani, P. K. (2002). "Parallel data intensive computing in scientific and commercial applications." *Parallel Computing*, 28(5), 673-704.

Catbas, F. N., Susoy, M., and Frangopol, D. M. (2008). "Structural health monitoring and reliability estimation: Long span truss bridge application with environmental monitoring data." *Engineering Structures*, 30(9), 2347-2359.

Chae, M. J., Yoo, H. S., Kim, J. Y., and Cho, M. Y. (2012). "Development of a wireless sensor network system for suspension bridge." *Automation in Construction*, 21, 237-252.

Chang, P. C., Flatau, A., and Liu, S. C. (2003). "Review paper: Health monitoring." *Structural Health Monitoring*, 2(3), 257-267.

Chen, P. M., Lee, E. K., Gibson, G. A., Katz, R. H., and Patterson, D. A. (1994). "RAID: High-performance, reliable secondary storage." *Association of Computing and Machinery (ACM) Computing Surveys (CSUR)*, 26(2), 145-185.

Daniel, I. M., Luo, J. J., Sifiniotopoulos, C. G., and Chun, H. J. (1997). "Acoustic emission monitoring of fatigue damage in metals." *Review of Progress on Quantitative Nondestructive Evaluation*, 16, 451-458.

Dijck, G. V. and Hulle, M. M. v. (2011). "Genetic algorithm for informative basis function selection from the wavelet packet decomposition with application to corrosion identification using acoustic emission." *Chemometrics and Intelligent Laboratory Systems*, 107(2), 318-332.

Doebling, S. W., Farrar, C. R., Prime, M. B., and Shevitz, D. W. (1996). "Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review." *Los Alamos National Laboratory*, Los Alamos, New Mexico.

Dowling N.E. (2007). *Mechanical behavior of materials: engineering methods for deformation, fracture, and fatigue*. Pearson Education, NJ.

Emamian, V., Kaveh, M., Tewfik, A. H., Shi, Z., Jacobs, L. J., and Jarzynski, J. (2003). "Robust clustering of acoustic emission signals using neural networks and signal subspace projections." *European Association for Signal Processing (EURASIP) Journal on Applied Signal Processing*, 3, 276-286.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." *Proceedings of 2nd International Conference on Knowledge Discovery in Databases and Data Mining (KDD-96),* Portland, Oregon, 226-231.

Fang, D., and Berkovits, A. (1995). "Fatigue design model based on damage mechanisms revealed by acoustic emission measurements." *Journal of Engineering Materials and Technology*, 117(2), 200–208.

Farrar, C. R., Doebling, S. W. and Nix, D. A. (2001). "Vibration-based structural damage identification." *Philosophical Transactions of the Royal Society Series A,* 359(1778), 131-149.

Farrar, C. R., Lieven, N. A. J., and Bement, M. T. (2005)."An introduction to damage prognosis." *Damage Prognosis: For Aerospace, Civil and Mechanical Systems*, John Wiley & Sons, Chichester, UK, ch1.

Farrar, C. R. and Worden, K. (2007). "An introduction to structural health monitoring." *Philosophical Transactions of the Royal Society Series A,* 365(1851), 303-315.

Flood, I., and Kartam, N. (1998). *Artificial neural networks for civil engineers: Advanced features and applications*. ASCE, Reston, VA.

Fricker, S. and Vogel, T. (2007). "Site installation and testing of a continuous acoustic monitoring." *Construction and Building Materials*, 21(3), March, 501-510.

Genel, K. (2004). "Application of artificial neural network for predicting strain-life fatigue properties of steels on the basis of tensile tests." *International Journal of Fatigue*, 26(10), 1027-1035.

Godinez-Azcuaga, V. F., Farmer, J., Ziehl, P. H., Giurgiutiu, V., Nanni, A., and Inman, D. J. (2012). "Status in the development of self-powered wireless sensor node for structural health monitoring and prognosis." *International Society for Optics and Photonics (SPIE): Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security*, San Diego, 8347.

Gong, Z., Nyborg, E. O., and Oommen, G. (1992). "Acoustic emission monitoring of steel railroad bridges." *Materials Evaluation*, 50(7), 883–887.

Gray, J., Liu, D. T., Nieto-Santiesteban, M., Szalay, A., Dewitt, D. J., and Heber, G. (2005). "Scientific data management in the coming decade." *ACM SIGMOD Record*, 34(4), 34-41.

Guha, S. Rastogi, R., and Shim, K. (2001). "CURE: An efficient clustering algorithm for large databases." *Information Systems*, 26(1), 35-58.

Gutkin, R., Green, C. J., Vangrattanachai, S., Pinho, S. T., Robinson, P., and Curtis, P. T. (2011). "On acoustic emission for failure investigation in CFRP: Pattern recognition and peak frequency analyses." *Mechanical Systems and Signal Processing*, 25(4), 1393-1407.

Hagan, M. T., Demuth, H. B., and Beale, M. H. (1996). *Neural network design*. PWS Publishing, Boston, MA.

Han, Y. L. (1995). "Artificial neural network technology as a method to evaluate the fatigue life of weldments with welding defects." *International Journal of Pressure Vessels and Piping*, 63(2), 205-209.

Harms, T., Sedigh, S., and Bastianini, F. (2010). "Structural health monitoring of bridges using wireless sensor networks." *IEEE Instrumentation & Measurement Magazine*, 13(6), 14-18.

Harris, D. O. and Dunegan, H. L. (1974). "Continuous monitoring of fatigue-crack growth by acoustic-emission techniques." *Experimental Mechanics*, 14(2), 71–81.

Hill, E. v. K., Isreal, P. L., and Knotts, G. L., (1993). "Neural network prediction of aluminum-lithium weld strengths from acoustic emission amplitude data." *Materials Evaluation,* 51(9), 1040-1045, 1051.

Hill, M. D. and Marty, M. R. (2008). "Amdahl's law in the multicore era", *IEEE Computer*, 41(7), 33-38.

Howard, P. (2011). "Back to a digital future-developing standards for managing and preserving inspection data." *ASTM Standardization News*, 39(4), 40-43.

Kara, N., Issa, O., and Byette, A. (2005.) "Real 3G WCDMA networks performance analysis." *The IEEE Conference on Local Computer Networks 30th Anniversary*, Washington DC, 586-592.

Khamedi, R., Fallahi, A., and Oskouei, A. R. (2010). "Effect of martensite phase volume fraction on acoustic emission signals using wavelet packet analysis during tensile loading of dual phase steels." *Materials & Design*, 31(6), 2752-2759.

Kim, K. B., Yoon, D. J., Jeong, J C., and Lee, S. S. (2004). "Determining the stress intensity factor of a material with an artificial neural network from acoustic emission measurements." *Nondestructive Testing and Evaluation International*, 37(6), 423-429.

Klotz, D. W., Natale, P. J., Herrmann, A., Basham, D. L., Bennett, J., Brown, J., and Calhoun, C. C. (2009). *Report Card for America's Infrastructure*. American Society of Civil Engineers, Reston.

Knuth, D. E. (1968). *The art of computer programming*. Addison-Wesley, Reading, MA.

Ko, J. M. and Ni, Y. Q. (2005). "Technology developments in structural health monitoring of large-scale bridges." *Engineering Structures*, 27(12), 1715-1725.

Kriegel, H.-P., Kroger, P., Schubert, E., and Zimek, A. (2009). "LoOP: Local outlier probabilities." *Proceedings of the 18th Association for Computing and Machinery (ACM) Conference on Information and Knowledge Management*, Hong Kong, 1649-1652.

Kulkarni, S. S. and Achenbach, J. D. (2008). "Structural health monitoring and damage prognosis in fatigue." *Structural Health Monitoring,* 7(1), 37-49

Lee, K. Y. (1989). "Cyclic AE count rate and crack growth rate under low cycle fatigue fracture loading." *Engineering Fracture Mechanics,* 34(5–6), 1069–1073.

Lehmann, E. L. and Casella, G. (1998). *Theory of point estimation*. Springer-Verlag, New York, NY.

Lei, X., Masuda, K., Nishizawa, O., Jouniaux, L., Liu, L., Ma, W., Satoh, T., and Kusunose, K. (2004). "Detailed analysis of acoustic emission activity during catastrophic fracture of faults in rock." *Journal of Structural Geology*, 26(2), 247-258.

Li, J., Du, G., Jiang, C., and Jin, S. (2012). "The classification of acoustic emission signals of 304 stainless steel during stress corrosion process based on K-means clustering." *Anti-Corrosion Methods and Materials*, 59(2), 76-80.

Lindley, T. C., Palmer, I. G., and Richards, C. E. (1978). "Acoustic emission monitoring of fatigue crack growth." *Materials Science and Engineering*, 32(1), 1–15.

Liptai, R. G., Harris, D. O., and Tatro, C. A. (1972). *Acoustic emission*. ASTM, Baltimore.

Lloyd, S. P. (1982). "Least squares quantization in PCM." *IEEE Transactions On Information Theory*, 28(2), March, 129-137.

Maitra, R., Peterson, A. D., and Ghosh, A. P. (2010). "A systematic evaluation of different methods for initializing the k-means clustering algorithm." *IEEE Transactions on Knowledge and Data Engineering*.

Manson, G., Worden, K., Holford, K., and Pullin, R. (2001). "Visualisation and dimension reduction of acoustic emission data for damage detection." *Journal of Intelligent Material Systems and Structures*, 12(8), 529-536.

Mathew, M. D., Kim, D. W., and Ryu, W. S. (2008). "A neural network model to predict low cycle fatigue life of nitrogen-alloyed 316L stainless steel." *Materials Science and Engineering: A*, 474(1-2), 247-253.

Marec, A., Thomas, J.-H., and El Guerjouma, R. (2008). "Damage characterization of polymer-based composite materials: Multivariable analysis and wavelet transform for clustering acoustic emission data." *Mechanical Systems and Signal Processing*, 22(6), 1441-1464.

Mohanty, S., Chattopadhyay, A., Peralta, P., and Das, S. (2011). "Bayesian statistic based multivariate Gaussian process approach for offline/online fatigue crack growth prediction." *Experimental Mechanics,* 51(6), 833-843.

Moore, M., Phares, B., Graybeal, B., Rolander, D., and Washer, G. (2001a). "Reliability of visual inspection for highway bridges." *Report FHWARD-01-020,* FHWA, Washington, DC.

Moore, M., Rolander, D., Graybeal, B., Phares, B., and Washer, G. (2001b). "Highway bridge inspection: State-of-the-practice survey." *Report FHWARD-01-033,* FHWA McLean, VA.

Moore, G.E. (1998). "Cramming more components onto integrated circuits." *Proceedings of the IEEE*, 86(1), 82-85.

Morton, T. M., Harrington, R. M., and Bjeletich, J. G. (1973). "Acoustic emissions of fatigue crack growth." *Engineering Fracture Mechanics*, 5(3), 691-697.

Morton, T. M., Smith, S., and Harrington, R. M. (1974). "Effect of loading variables on the acoustic emissions of fatigue-crack growth." *Experimental Mechanics,* 14(5), 208-213.

Nemati, N. (2012). "Acoustic emission assessment of steel bridge details subjected to fatigue." Doctoral dissertation, University of Miami, Coral Gables, FL.

Ni, Q.-Q. and Iwamoto, M. (2002). "Wavelet transform of acoustic emission signals in failure of model composites." *Engineering Fracture Mechanics*, 69(6), 717-728.

Oh, K. H., Jung, C. K., Yang, Y. C., and Han, K.S. (2004). "Acoustic emission behavior during fatigue crack propagation in 304 stainless steel." *Key Engineering Materials 261*, 1325–1330.

Oliveira, R.d. and Marques, A.T. (2008). "Health monitoring of FRP using acoustic emission and artificial neural networks." *Computers and Structures*, 86(3-5), 367-373.

Omkar, S. N. and Karanth, R. (2008). "Rule extraction for classification of acoustic emission signals using Ant Colony Optimisation." *Engineering Applications of Artificial Intelligence*, 21(8), 1381-1388.

Patterson, D. A., Chen, P., Gibson, G., and Katz, R. H. (1989). "Introduction to redundant arrays of inexpensive disks (RAID)." *COMPCON Spring '89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage, Digest of Papers.,* New York, NY, 112-117.

Paris, P. C., and Erdogan, F. (1963). "A critical analysis of crack propagation laws." *Journal of Basic Engineering*, 85, 528–534.

Pelleg, D. and Moore, A. (2000). "X-means: Extending k-means with efficient estimation of the number of clusters." *Proceedings of the 17th International Conference on Machine Learning,* 1, 727-734.

Pleune, T. T. and Chopra, O. K. (2000). "Using artificial neural networks to predict the fatigue life of carbon and low-alloy steels." *Nuclear Engineering and Design,* 197(1), 1-12.

Pollock, A. A. (1973). "Acoustic emission amplitudes." *Non-Destructive Testing*, 6(5), 264-269.

Qian, W., Xie, L., Huang, D., and Yin, X. (2009). "Pattern recognition of fatigue damage acoustic emission signal." *IEEE International Conference: Mechatronics and Automation (IEEE-ICMA)*, Changchun, China, 4371-4375.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo.

Rabiei, M., Modarres, M. and Hoffman, P. (2009)."Probabilistic structural health monitoring using acoustic emission." *Annual Conference of the Prognostics and Health Management Society.*

Raghavan, A. and Cesnik, C. E. (2007). "Review of guided-wave structural health monitoring." *Shock and Vibration Digest,* 39(2), 91–116.

Reid, R. (2008). "The infrastructure crisis." *Civil Engineering Magazine*, 78(1), 40-65.

Sause, M. G. R., Gribov, A., Unwin, A. R. and Horn, S. (2012). "Pattern recognition approach to identify natural clusters of acoustic emission signals." *Pattern Recognition Letters*, 33(1), 17-23.

Schofield, B. H. (1972). "Research on the sources and characteristics of acoustic emission." *ASTM Selected Technical Papers (ASTM STP): Acoustic Emission,* 11-19.

Samarasinghe, S. (2007). *Neural networks for applied sciences and engineering: From fundamentals to complex pattern recognition*. Auerbach Publications.

Srinivasan, V. S., Valsan, M., Bhanu Sankara Rao, K., Mannan, S. L., and Raj, B. (2003). "Low cycle fatigue and creep–fatigue interaction behavior of 316L (N) stainless steel and life prediction by artificial neural network approach." *International Journal of Fatigue,* 25(12), 1327-1338.

Srivastava, A. and Eustace, A. (1994). "Atom: A system for building customized program analysis tools." *Proceedings of the Association for Computing Machinery's Special Interest Group on Programming Languages (ACM SIGPLAN) 1994 Conference on Programming Language Design and Implementation*, 29(6), 196-205.

Vassiliadis, P., Simitsis, A., Geaogantas, P., Terrovitis, M., and Skiadopoulos, S. (2005). "A generic and customizable framework for the design of ETL scenarios." *Information Systems*, 30(7), 492-525.

Wang, X., Lu, Y. and Tang, J. (2008). "Damage detection using piezoelectric transducers and the Lamb wave approach: I. System analysis." *Smart Materials and Structures*, 17(2), 025033.

Weiss, N. A. and Weiss, C. A. (2012). *Introductory statistics*. Addison-Wesley, Reading, MA.

Williams, J. H. (1982). "Correlations of acoustic emission with fracture mechanics parameters in structural bridge steels during fatigue." *Materials Evaluation*, 40(11), 1184–1189.

Wong, K., Tilley, S. R., Muller, H. A., and Storey, M.-A. D. (1995). "Structural redocumentation: A case study." *IEEE Software*, 12(1), 46-54.

Worden, K. and Dulieu-Barton, J. M. (2004). "An overview of intelligent fault detection in systems and structures." *Structural Health Moniotring*, 3(1), 85-98.

Yang, Y. and Webb, G. I. (2003). "Weighted proportional k-interval discretization for naive-bayes classifiers." *Advances in Knowledge Discovery and Data Mining,* 2637, 565.

Yoon, D. J., Jung, J. C., Park, P., and Lee, S. S. (2000). "AE characteristics for monitoring fatigue crack in steel bridge members." *SPIE 5th Annual International Symposium on Nondestructive Evaluation and Health Monitoring of Aging Infrastructure*, Newport Beach, 153-162.

Yu, J., Ziehl, P., Zárate, B., and Caicedo, J. (2011). "Prediction of fatigue crack growth in steel bridge components using acoustic emission." *Journal of Constructional Steel Research,* 67(8), 1254-1260.

Zárate, B. A., Caicedo, J. M., Yu, J., and Ziehl, P. (2012a). "Deterministic and probabilistic fatigue prognosis of cracked specimens using acoustic emissions." *Journal of Constructional Steel Research,* 76, 68-74.

Zárate, B. A., Caicedo, J. M., Yu, J., and Ziehl, P. (2012b). "Probabilistic prognosis of fatigue crack growth using acoustic emission data." *Journal of Engineering Mechanics,* 138(9) 1101-1111.

VITA

Felipe Mejia was born in Pereira, Colombia, on November 8, 1986. His parents are Carlos H. Mejia, a civil engineer, and Marta L. Zuluaga, a child psychologist; and his sister is Monica Mejia, a doctoral candidate in Integrative Biology/Neuroscience. Felipe received his elementary education at the Calasanz School in Pereira, Colombia, and his secondary education at Marjory Stoneman Douglas High School in Parkland, Florida. In August 2005 he entered the University of Miami from which he was graduated with the B.S. degree in December 2008. Immediately following his undergraduate studies, Felipe entered the Ph.D. program in Civil Engineering at the University of Miami and, after being granted the M.S. degree in December 2011, was conferred the Ph.D. degree in December 2012.

Permanent Address: 11721 W. Atlantic Blvd. Apt. 7-26, Coral Springs, Florida 33071