ADVANCED DATA ANALYSIS FRAMEWORK FOR DAMAGE IDENTIFICATION IN CIVIL INFRASTRUCTURE BASED ON SELF-POWERED SENSING

By

Amir Hossein Alavi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Civil Engineering- Doctor of Philosophy

2016

ABSTRACT

ADVANCED DATA ANALYSIS FRAMEWORK FOR DAMAGE IDENTIFICATION IN CIVIL INFRASTRUCTURE BASED ON SELF-POWERED SENSING

By

Amir Hossein Alavi

This interdisciplinary research proposes an advanced data analysis framework for civil infrastructure/structural health monitoring (I/SHM) based on a pioneering self-powered sensing technology. The current work characterizes the performance of a fairly new class of self-powered sensors for specific application problems with complex behavior. The proposed health monitoring systems are established through the integration of statistical, artificial intelligence and finite element methods. Different infrastructure systems with various damage types are analyzed. A new probabilistic artificial intelligence-based damage detection technique is developed that hybridizes genetic programming and logistic regression algorithms. The proposed multi-class classification system assigns probabilities to model scores to detect damage progression. A probabilistic neural network method based on Bayesian theory is further introduced to improve the damage detection accuracy. Data obtained from the finite element simulations and experimental study of hybrid sensor networks is used to calibrate the data interpretation algorithms. The network architecture comprises self-powered sensors that use the electrical energy directly harvested by piezoelectric ceramic Lead Zirconate Titanate (PZT) transducers. The beauty of this so-called self-powered monitoring system is that the operating power for the smart sensors directly comes from the signal being monitored. An advantage of using these sensors is that there is no need to directly measure the absolute value of strain in order to estimate damage. In fact, the proposed self-sustained sensing systems use the sensor output to relate the variation rate of strain distributions to the rate of damage. The proposed data analysis framework consists of multilevel strategies for structural/infrastructure damage identification through: (a) analysis of individual self-powered strain sensors, (b) data fusion in a network of self-powered strain sensors, and (c) data analysis in a hybrid network of self-powered accelerometer and strain sensors. For each of these levels, several damage indicator features are extracted upon the simulation of the compressed data stored in memory chips of the self-powered sensors. A new data fusion concept based on the effect of group of sensors, termed as "group effect", is proposed. The goal is to improve the damage detection performance through spatial measurements over structures. Moreover, combination of the data from a network of accelerometer and strain sensors results in developing an integrated global-local damage detection approach. The investigated cases are crack growth detection in steel plates under a uniaxial tension mode, distortion-induced fatigue cracking in steel bridge girders, continuous health monitoring of pavement systems, failure of simply supported beam under three-point bending, and failure of gusset plate of the I-35W highway bridge in Minneapolis, Minnesota. 3D dynamic finite element models are developed for each of the cases. The experimental studies are carried out on a steel plate subjected to an in-plane tension, a steel plate with bolted connections, and on asphalt concrete specimens in three-point bending mode. PZT-5A ceramic discs and PZT-5H bimorph accelerometers are placed on the surface of the plates to measure the delivered voltage in each damage phase. For the asphalt experiments, a new miniaturized spherical packaging system is designed and tested to protect the PZT ceramic discs embedded inside the specimen. Uncertainty analyses are performed through the contamination of the damage indicator features with different noise levels. The results indicate that the proposed I/SHM systems are efficiently capable of detecting different damage states in spite of high-level noise contamination.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisors, Dr. Nizar Lajnef and Dr. Karim Chatti, for their invaluable support, patience, and encouragement throughout my graduate studies. Without whom it was not possible. Their technical advice has taught me numberless lessons and insights on the workings of academic research.

I would like to thank the members of my committee, Dr. Neeraj Buch and Dr. Erik D. Goodman, Director of the BEACON Center for the Study of Evolution in Action, and Mr. Fred Faridazar from the U.S. Federal Highway Administration (FHWA) for their guidance, advice and support throughout the duration of my research and providing many valuable comments.

The friendship of Hassene Hasni, Wassim Borchani, and Pengcheng Jiao is much appreciated and has led to many interesting discussions relating to this research. I am also grateful to my friend Dr. Amir H. Gandomi for his invaluable comments. I would like to thank Margaret Connor and Laura Post for making the paperwork process so easy and your help with the administrative process.

Last, but not least, I would like to thank my wife Fariba for her understanding and love during the past few years. Her support and encouragement was in the end what made this dissertation possible. My parents, Mehrnaz and Mojtaba, receive my deepest gratitude and love for their dedication and the many years of support during my undergraduate studies that provided the foundation for this work.

This work is supported in part by the FHWA grants DTFH61-13-C-00015 and DTFH61-13-H-00009.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Background and State of Knowledge

Infrastructure/structural health monitoring (I/SHM) is an emerging field in civil engineering for continuous damage assessment and safety evaluation of civil infrastructure. As a result of notable developments in the signal analysis and information processing techniques, numerous I/SHM approaches are developed (Lee et al., 2005). In this context, one of the most widely-used approaches is vibration-based method (Doebling et al., 1996; Zou et al., 2000). Signal and model-based techniques are the main classes of the vibration-based method. The first category is based on defining the damage by indices and comparing the structural responses before and after damage (Huang et al., 1999; Hou et al., 2000; Quek et al., 2001). The signal-based methods are generally appropriate for detecting the damage locations (Lee et al., 2005). On the other hand, a major feature of the model-based methods is that they can detect both the damage locations and severities by improving the structural mathematical model (Shi et al., 1998; Stetson et al., 1981). An advantage of using such methods is that they require a relatively small number of sensors. Furthermore, they are highly effective for monitoring catastrophic events. However, a notable limitation of these techniques is that they are not sufficiently sensitive for detecting long term minor fatigue damage. A comprehensive review about the existing structural damage detection techniques can be found in (Farrar et al., 2006; Sohn et al., 2004; Farrar, 2001; Fan and Qiao, 2011). Another class of the damage detection methods, called nondestructive testing and evaluation (NDT&E) or nondestructive inspection has been the focus of many studies. NDT&E is the structural condition assessment without removing the individual structural components (Zong et al., 2002; Lynch, 2005). The NDT&E technology is commonly classified as a local-based damage detection approach. Some of the well-known NDT&E techniques are acoustics, emission spectroscopy, fiber-scope, hardness testing, leak testing, magnetic perturbation, X-ray, pulse-echo, and radiography (Zong et al., 2002). Different type of sensors can be used in this domain such as pressure cell, deflectometer, strain gauge, thermocouple, moisture sensor, fiber-optic sensors, etc (Lynch, 2005; Lynch et al., 2003; Lynch et al., 2004; Spencer et al., 2004; Spencer, 2003).

In the last decade, significant attention has been devoted to the utilization of new sensing technologies for instrumentation within the structural systems. A major drawback of using traditional wired sensors pertains to the difficulties in deploying and maintaining the associated wiring system. Moreover, managing huge amount of data generated by a dense array of wired sensors is very challenging and costly (Sundaram et al., 2013). To cope with these limitations, wireless sensor networks (WSNs) are increasingly utilized as alternatives to traditional structural engineering monitoring systems. The significant capability of WSNs for sensing the physical state of the structural systems has attracted considerable attention in recent years (Lynch, 2005; Lynch et al., 2003). In addition to the conventional monitoring applications of WSNs, they are autonomous data acquisition nodes providing valuable spatio-temporal information of the structure (Watters et al., 2003; Lynch and Loh, 2006). Dense arrays of low-cost smart wireless sensors can offer useful data about the structural deterioration. Such information can be efficiently used to enhance the performance of the SHM systems (Sundaram et al., 2013). Recent development and applications of smart sensors and sensing systems are comprehensively introduced by Sundaram et al. (2013) and Yun and Min (2011). However, a major concern for the application of wireless sensors is related to the difficulties of powering them. To tackle this issue, harvesting ambient energy seems to be an attractive solution (Sirohi and Chopra, 2001; Roundy et al., 2002; Rahimi et al., 2003; Elvin et al., 2001, 2003; Lajnef et al., 2011; Lajnef et al., 2008; Lajnef et al., 2013; Huang et al., 2010). Energy harvesting is the possibility of converting mechanical energy into electrical energy (Elvin et al., 2001). Among various self-powering energy sources, piezoelectric transducers are proved to be one of the most efficient choices (Elvin et al., 2001, 2003; Lajnef et al., 2011; Lajnef et al., 2008; Lajnef et al., 2013; Huang et al., 2010). For SHM, piezoelectric transducers can be used for the self-powering of wireless sensors through harvesting energy from the mechanical loading experienced by the structure (Lajnef et al., 2013).

Recently, a new class of self-powered wireless sensors (SWS) has been developed by the authors at Michigan State University (MSU) based on "smart" pebble concept (Lajnef et al., 2008; Lajnef et al., 2013; Huang et al., 2010). The "smart" pebble generally refers to a battery-less sensor having a size comparable to the grain size of the construction material. By embedding these sensors inside the structure, it is possible to monitor the localized strain statistics. The recorded information can be used for early damage detection and future condition evaluation. The designed SWS is a small size battery-less sensor. The prototype of this miniaturized strain-senor is shown in Figure 1.

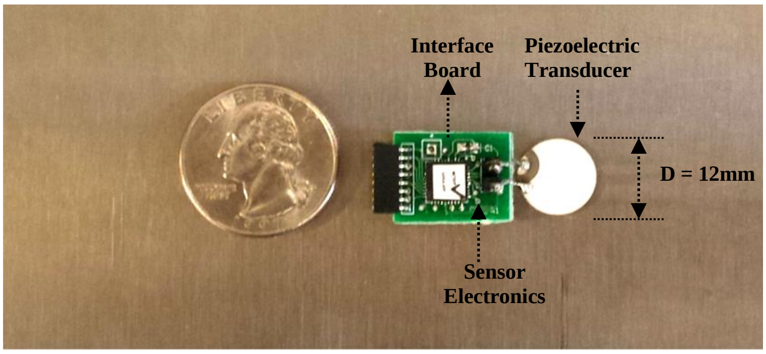


Figure 1. The prototype of the SWS system

This unique piezo-floating-gate (PFG) sensor is based on the integration of piezoelectric transducers with an array of ultra-low power floating-gate computational circuits (Lajnef et al., 2008). The sensor is fabricated using a p-channel floating-gate metaloxide-semiconductor (pMOS) transistor (Huang et al., 2010; Lajnef et al., 2011). The transistor is connected to a constant current source powered by the piezoelectric transducer. The piezoelectric transducer harvests the energy that will be used to inject electrons from the transistor channel onto the floating-gate. The impact-ionized hot-electron injection (IHEI) model can be used to derive the change/droppage of the floating-gate voltage with respect to time and drain current. Subsequently, for long-term monitoring, the cumulative duration for which the injector is operational can be obtained using the measured floating-gate voltage (Huang et al., 2010; Lajnef et al., 2011). However, the prototypes of the sensor can have floating-gates with constant and variable electron injection rates. The floating-gate injection rate is a property of the gate that controls the injection of the electrons into the gate. This parameter is correlated with the voltage droppage rate across the gate. Note that the cumulative time at specific pre-defined strain/voltage thresholds is proportional to the voltage droppage across the memory gate. The output of the memory cells can be reported in the form of a histogram where each bin represents the cumulative time of occurrences at a predetermined load level. This can be done for sensors with either constant or variable injection rates. The recorded data are stored on-board the sensor from the time it is installed, and can be periodically read using a Radio Frequency Identification (RFID) scanner (Lajnef et al., 2011; Alavi et al., 2016a,b). One of the main advantages of this sensing system is the fact that it is “response-based”. All the effects due to variations in load location, load magnitude, traffic wander, environmental effects such as temperature and moisture, and material aging and degradation, are aggregated in the strain response recorded by the sensor over time. This feature makes the sensor suitable for long-term I/SHM. Most of the other existing solutions evaluate the condition of the system at a given instant. These methods present only a snapshot at the time where the measurements are taken. Thus, the obtained results are highly influenced by the environmental conditions. Since the developed SWS records each and every event all the time, it will aggregate all these short-term fluctuations. Thus, if long-term shifts are observed in the results, they are most probably correlated with condition degradation. Herein, the whole methodology is based on relative damage as the sensor does not directly measure the absolute value of strain. The rate of variation of strain distributions is related to the rate of damage. Research in the previous Federal Highway Administration (FHWA) funded project revealed the applicability of the SWS for continuous monitoring of infrastructures (Lajnef et al., 2013). That project is basically focused on the manufacturing of the sensor electronics, and design of a packaging system to withstand loading and environmental conditions for the pavement implementation. A limited study is done on developing a method for predicting remaining fatigue life of pavement and generating missing data from a set of measurements by a classical statistical technique (Lajnef et al., 2011, 2013).

Despite several advantages of using SWS, there would be a considerable loss of information. In fact, a part of the sensed information is compressed as a function of cumulative time at each load level. This drawback results in a notable difficulty in the interpretation of the data generated by SWS. However, no efficient method has yet been proposed for SHM using the valued information offered by this smart sensor technology. Accordingly, this interdisciplinary research is aimed at developing innovative data interpretation frameworks for I/SHM based on the data provided by networks of SWSs. To this aim, extensive analyses are performed utilizing statistical, AI and FE methods. Moreover, a new AI-based damage detection approach, called GPLR, is proposed that hybridizes genetic programming (GP) and logistic regression (LR) algorithms. The sensor-specific data interpretation algorithms are calibrated using the data from FE simulations and experimental studies. The self-sustained sensing systems use the sensor output to relate the variation rate of strain distributions to the rate of damage. The proposed data interpretation framework for structural/infrastructure damage identification is established upon the multilevel analysis of individual self-powered strain sensors, data fusion in networks of self-powered strain sensors, data fusion in hybrid networks of self-powered accelerometer and strain sensors, and FE model updating using strain data. Several infrastructure systems are analyzed with different type of damages. The investigated cases are continuous health monitoring of pavement systems, failure of simply supported beam under three-point bending, crack growth detection in steel plates under a uniaxial tension mode, distortion-induced fatigue cracking in steel bridge girders, and failure of gusset plate of the I-35W highway bridge in Minneapolis, Minnesota. In order to verify the robustness of the systems, uncertainty analyses are carried out through the contamination of the damage indicator features with different noise levels.

1.2 Research Objectives

Self-powered sensors use only self-generated electrical energy directly harvested by piezoelectric transducers from the structure under vibration. In fact, the sensed signal is used for both I/SHM and empowering the sensor. This concept can be referred to as “self-powered monitoring”. The sensors have a series of memory cells that store the load history profiles at a specific preset level. The output is reported in the form of a histogram where each bin represents the cumulative time of occurrences at a predetermined load level. Despite several advantages of using this type of self-powered sensors, there is a considerable loss of information. A part of the sensed information is compressed as a function of cumulative time at each load level. This drawback results in a notable difficulty in the analysis of the data generated by the self-powered sensors. One of the major obstacles in successful application and development of the self-powered sensors is lack of an advanced framework for the interpretation of the limited but valued information offered by this smart technology. The main goal of the present study is to address this gap by introducing a comprehensive data interpretation framework for the battery-less health monitoring of civil infrastructure based on the data provided by hybrid networks of self-powered sensors. The general hypothesis behind the proposed research is that damage in civil infrastructure can be identified using compressed data stored on-board the memory cells of the self-powered sensors and advanced data analysis methods. The proposed health monitoring systems are established through the integration of statistical, AI and FE methods. Advanced probabilistic algorithms are developed and evaluated during this project. The other main objective is to characterize the performance of the sensing system for specific application problems such as detecting and classifying different stages of damage in a structural/infrastructure component, tracking of damage accumulation in a fatigue sensitive detail, etc. As the sensing system can be used for different modalities of monitoring, other application cases are also considered during this project as more testing is conducted and feedback is obtained.

1.3 Research Significance

Figure 2 shows a typical Smart City sensor model. As seen in this figure, wireless sensors develop the foundation for Smart Cities. WSNs adopt a spatially distributed network of autonomous intelligent sensors to measure a variety of physical or environmental parameters. While the concept of Smart City has been recently receiving significant attention from the public authorities and private developers, there are some technical challenges to develop a successful Smart City project. In this context, two major issues are to establish data analysis frameworks for the interpretation of heterogeneous data being produced by these sensors and to develop innovative power solutions for supplying energy for WSNs.

This research work partially addresses both of these challenges through development of innovative data mining and data fusion systems that embrace state-of-the-art signal processing and NDT&E techniques, as well as an energy harvesting approach. From a data analysis perspective, the proposed systems are based on knowledge discovery in databases (KDD) and data mining processing methods to make full use of single and multi-sensor data (see Figure 3). KDD and data mining are multidisciplinary areas including several paradigms such as machine learning, pattern recognition, statistics, intelligent databases, knowledge acquisition, data visualization, high performance computing, expert systems, etc. Besides, the NDE technique used in this research is based on a pioneering energy scavenging paradigm in which the operating power of a sensor is harvested directly from the signal being sensed. Accordingly, the proposed I/SHM system not only tackles the limitations associated with the deployment and maintenance of the traditional wired sensors but also possesses major advantages over conventional energy scavenging approaches, as well as over nearly all of the commercially viable wireless sensors for I/SHM that require an external power source (either battery or solar power). On the other hand, fusion of the data provided by heterogeneous self-powered sensors leads to more informative global and local damage diagnostic information, and eventually improved safety and reduced maintenance costs of important infrastructure.

On this basis, the proposed framework is an integral part of the next generation of smart civil infrastructure that is capable of self-diagnosis of damage before the occurrence of any failure. More, such integrated sensing systems can be modified to become building blocks of future medical, mechanical, civil, transportation, and aerospace long-term sensing technologies in Smart Cities. Taking into account different aspects of the project, it will serve as an excellent educational platform to educate and train the next generation researchers and engineers in domains of data mining, data fusion, and self-powered sensing technology.

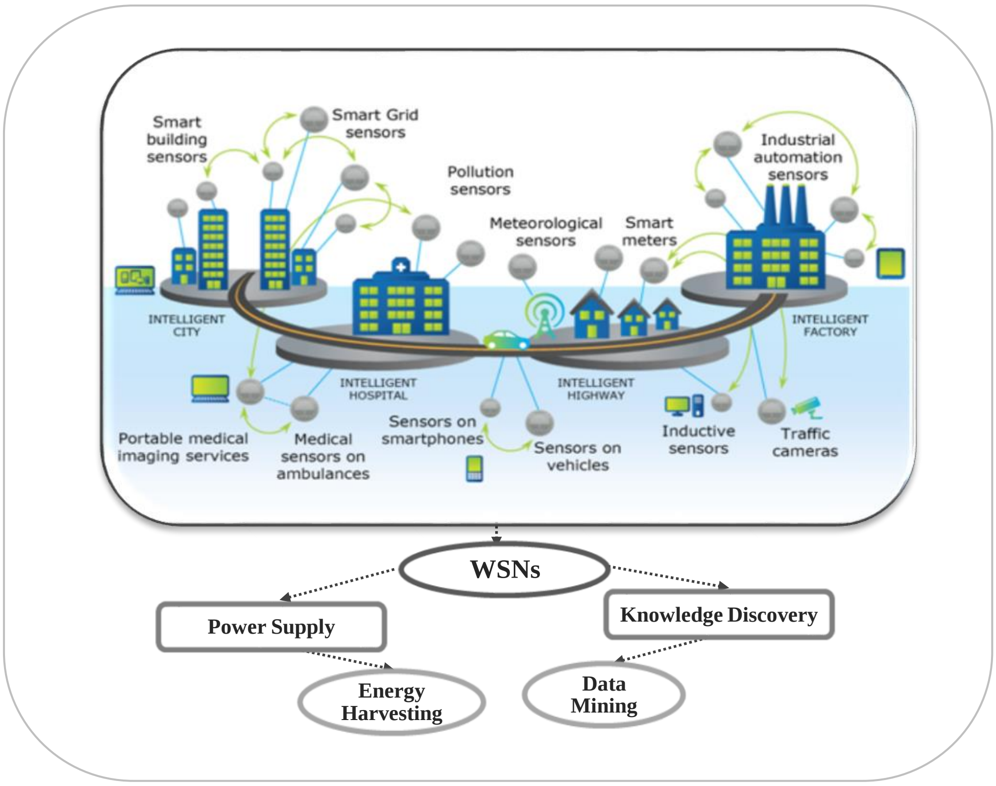


Figure 2. A Smart City sensor model

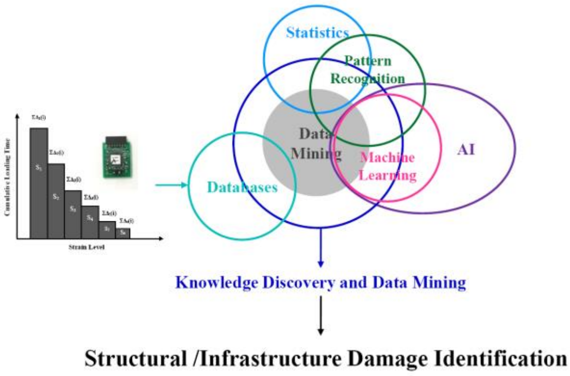


Figure 3. Knowledge discovery and data mining processing methods for civil infrastructure damage identification

1.4 Chapter Overview

The study will be guided by three major tasks for the identification of damage in civil infrastructure based on the self-powered sensing technology. An overview of each chapter is shown below, which delineates the specific objective, body, and conclusion.

Chapter II deals with the development of a data interpretation system for structural/infrastructure damage identification using point self-powered strain sensors. The goal is to discover a reasonable relationship between the damage indicator features of point sensors and damage progression. A number of probability distributions are considered to characterize the sensor output. The functionality of the proposed damage detection approach is evaluated on specific infrastructure systems with complex behavior. Extensive numerical and experimental studies are performed to achieve this goal. The cases analyzed in this chapter are continuous health monitoring of pavement systems, crack growth detection in steel plates under a uniaxial tension mode, and distortion-induced fatigue cracking in steel bridge girders. For the asphalt pavement experiments, a new installation procedure is proposed to address the major limitations of the installation of conventional H-shaped sensors. Based on the envisioned installation procedure, the sensors are embedded within a small-scale spherical packaging system that can be tossed into the paving materials during construction. However, the obtained results show that there is a sound relationship between the damage progression and features from the entire memory cells of the sensor.

A major limitation of analysis with point sensors is that they give only partial information about the health status of the structures. On this basis, Chapter III outlines the development details of a new approach defined for data fusion in a network of self-powered strain sensors. The goal is to use the information provided by a group of sensors, termed as "group effect". Statistical and AI approaches are used to fuse the data. Based on the statistical analysis, standard deviation of the extracted damage indicator features of group of sensors is positively correlated with the progression of damage. For the cases where the statistical approach fail to provide decent detections, a new probabilistic AI-based damage detection technique, called GPLR, is developed. This method hybridizes the GP and LR algorithms to assign probabilities to model scores for the detection damage progression. The application of this multi-class classification system is not limited to this study and can be extended to discover the patterns in a network of sensors or other classification problems in engineering domain. Further, a PNN analysis is performed for more verification of the results. A set of predictor features, termed as Z-functions, are defined and fed into the AI algorithms to enhance the damage detection accuracy. Besides, uncertainty analyses are carried out to assess the effect of noise and enhance the reliability of the proposed methods.

Chapter IV is focused on fusion of the data from a network of accelerometer and strain sensors to develop an integrated global-local damage detection approach. To this aim, PZT cantilever-based accelerometers and PZT strain sensors are designed. To illustrate the sensing scheme and decision fusion in a self-powered WSN, both numerical and experimental studies are conducted. An aluminum fixture with bolted connections is fabricated for the experimental study. A new type of damage is considered based on the changes in boundary condition. On this basis, different damage states are obtained by loosening (or removing) one bolt at a time from the plate. When all bolts are fully tightened, the structure is healthy. A bolt loose condition corresponds to a damaged state. Damage localization requires identification of which of the bolts is loose or missing. Based on the results, most of the PZT accelerometers are sensitive to the changes in the system and can be used to monitor changes in the characteristics of the overall structure. On the other hand, the damage details and its approximate location can be identified by the PZT strain sensors.

Finally, Chapter V summarizes the work performed under this project, outlines the main research products developed, and presents the main findings of the study. Some directions for future research are also presented.

CHAPTER II

DAMAGE IDENTIFICATION USING POINT SELF-POWERED STRAIN SENSORS

2.1 Characterization of the Working Mechanism of the SWS

Damage detection algorithms are one of the main parts of the I/SHM systems. These algorithms are developed to analyze raw sensor data and provide a precise diagnosis of the damage state. Apparently, quality and quantity of raw sensor data have a direct effect on the accuracy of damage detection (Singh and Joshi, 2009). The new SWS studied in this research is capable of continuously heath monitoring of the host structure based on the piezoelectric energy harvesting technology. It is known that the piezoelectric transducers have the ability to convert the mechanical applied charge to an electrical charge using the direct piezoelectricity effect. The open source voltage (V) generated across the piezoelectric ceramic Lead Zirconate Titanate (PZT) transducer material is given by the following equation:

$$V = \frac{S\,Y\,d_{31}\,h}{\varepsilon} \tag{1}$$

where S, Y, d_{31}, h and ε, are the applied strain, Young's modulus of the piezoelectric material, piezoelectric constant, thickness, and the electrical permittivity, respectively. The sensor can be economically attached to the structures either during construction, or anytime during routine maintenance operations. The communication between the sensor and a service vehicle is done using the RFID technology. The convertible electrical power levels in structures are typically less than 1 μW. The sensor also uses novel analog signal processing circuits that require less than 1 μW of power (Rhimi et al., 2012). These sensors have a series of memory cells that record the cumulative time of the voltage harvested by the piezoelectric transducer attached to the structure. In fact, they measure the duration of events when the amplitude of the input signal, coming from the piezoelectric transducer, exceeds different thresholds. Figure 2 presents a schematic representation of the level crossing cumulative time counting implemented by the SWS.

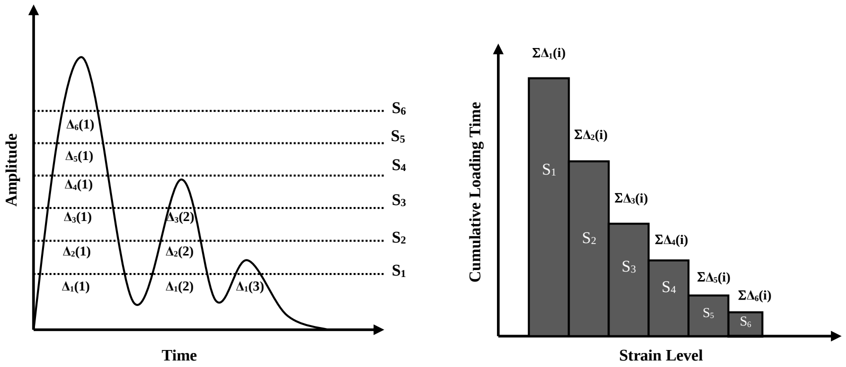


Figure 4. The level crossing cumulative time counting implemented by the SWS

The cumulative voltage time can be extracted for the sensors with either constant or variable injection rates by measuring of the voltage droppage in the memory cells (floating-gates) after a periodic excitation. The recorded time reflects the cumulative duration for which the floating-gate injector is operational. Referring to the IHEI model (Chakrabartty 2013), the change in floating-gate voltage (V_{PFG}) with respect to time and drain current (I_0) can be expressed as:

$$V_{PFG}(t) = \frac{1}{C_2}\, log(C_1 C_2 t + e^{-C_2 V_{PFG0}}) \tag{2}$$

where

V_{PFG0}: Initial floating-gate voltage

t: Total duration for which the injector is operational.

C_1 and C_2: Device parameters that can be written as:

$$C_1 = \frac{\beta I_0}{C_c}\left(\frac{I_0}{I_s}\right)^{\frac{V_T}{V_{Inj}}} \tag{3}$$

$$C_2 = \frac{\kappa}{V_{Inj}} \tag{4}$$

in which

β and V_{inj}: Constant injection parameters that are a function of the transistor size and the process parameters.

I_S: Pre-exponential current

κ: Floating-gate efficiency

V_T: Thermal voltage (26 mV at room temperature)

C_C: Total capacitance of the floating node.

Finally, for long-term monitoring, the cumulative duration can be estimated using the following equation:

$$t = \frac{e^{-C_2 V_{PFG}(t)}}{C_1 C_2} \tag{5}$$

In this context, more details on the derivation, initialization and calibration of the procedures can be found in (Lajnef et al. 2008; Huang et al. 2010; Lajnef et al. 2011). However, as can be observed from Figure 4, the only information that can be extracted from the sensor is the cumulative duration of voltage/strain events. Accordingly, the sensor does not provide information about the normally distributed strain histograms induced by the service loads every reading period. In other words, the sensor does not directly measure the absolute value of strain in order to estimate damage. The rate of variation of strain distributions is related to the rate of damage. The whole methodology is based on relative damage. That is why interpreting the data generated by SWS is a complicated task and highly desired. Different probability distributions may be used to analyze the sensor output. In order to develop a damage detection algorithm based on the interpretation of such limited data, a new approach is considered in this research. It is well-known that the service load in structures is usually defined by a Gaussian distribution and thus the induced local strain. With a Gaussian distributed load, the sensor output is the summation of the cumulative time strain distributions. Since the summation of Gaussian distributions can be assumed by a Gaussian distribution, this study rationally assumes that the sensor output can be characterized by the following cumulative density function (CDF) (Alavi et al., 2016a,b,c):

$$CDF_{\text{Gaussian}}(\varepsilon) = \frac{\alpha}{2}\left[1 - erf\left(\frac{\varepsilon-\mu}{\sigma\sqrt{2}}\right)\right] \tag{6}$$

where, μ, σ and α are mean of distribution, standard deviation accounting for the load and frequency variability, and total cumulative time of the applied strain measured by the entire gates. In fact, μ and σ can be considered the only viable tools to define the SWS output data. These parameters can be obtained by a curve fitting of the sensor output distribution collected from the entire memory cells (Alavi et al., 2016). Consequently, the damage state is logically considered to be a function of μ and σ. In order to obtain these parameters, the cumulative time of occurrences at predetermined strain levels are first determined. Then, the corresponding μ and σ values for each damage scenario are obtained through the fitting of a CDF (Eq. (6)) to the cumulative duration of strain events. Then, the CDF function is converted into probability density function (PDF) using the following equation:

$$PDF(\varepsilon) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\varepsilon-\mu)}{2\,\sigma^2}} \tag{7}$$

Figure 5 illustrates the transformation from the CDF form to the PDF form. More in-depth analyses are conducted to find the relationship between the PDFs and damage growth in the investigated structures. The major advantage of the proposed method is that it is based on relative variations in the strain response distributions. This means that there is no need to measure damage directly. The effect of damage is “sensed” by the sensors, and it is evaluated based on relative shifts of PDFs over time. This is schematically shown in Figure 6. When the sensors are deployed in a network, the sensors that are closer to critical locations will experience more prominent shifts in the distributions compared to other sensors. Knowing the exact location of these sensors gives the locations of the critical damage areas.

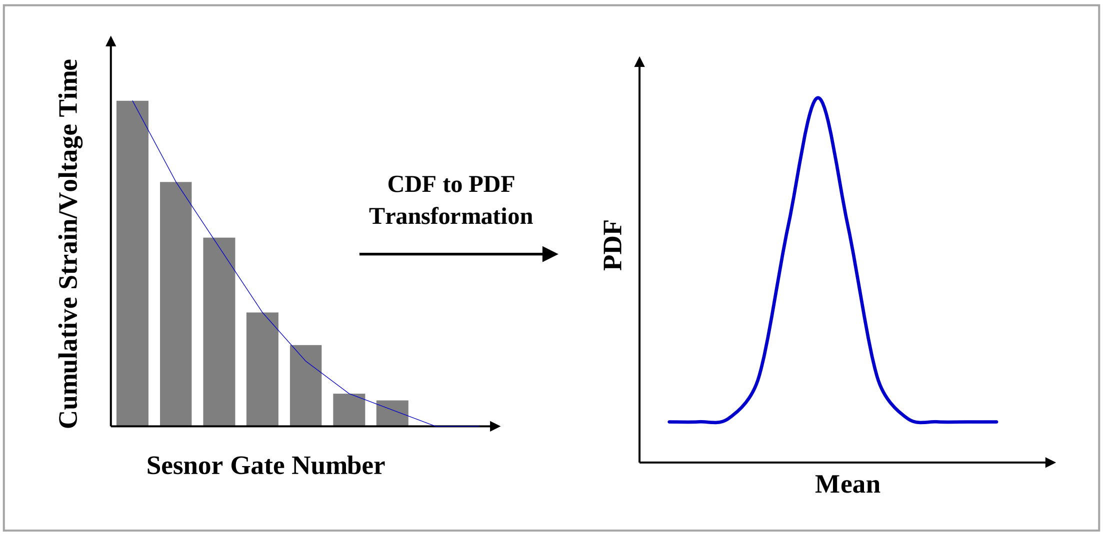


Figure 5. Transformation scheme from CDF to PDF

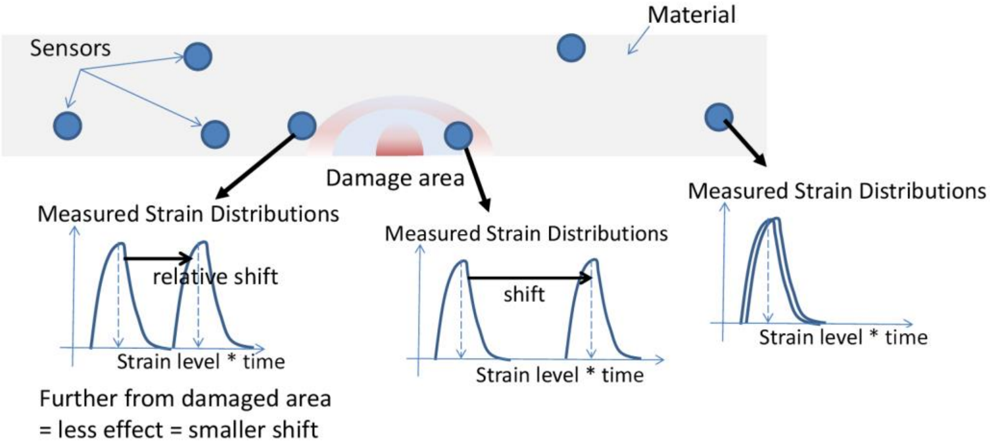


Figure 6. A schematic representation of the PDF shifts due to damage progression

It is worth mentioning that in addition to the Gaussian CDF given in Eq. (6), a number of other probability distributions are considered to characterize the sensor output as follows:

Rayleigh distribution:

$$CDF_{\text{Rayleigh}}(\varepsilon) = 1 - e^{\frac{-x^2}{2\sigma^2}} \tag{8}$$

where σ > 0, is the scale parameter of the distribution.

Shifted Gompertz distribution:

$$CDF_{\text{Gompertz}}(\varepsilon) = (1 - e^{-bx})e^{-\eta e^{-bx}} \tag{9}$$

where b > 0 is the scale parameter and η > 0 is the shape parameter of the shifted Gompertz distribution.

Weibull distribution:

$$CDF_{\text{Weibull}}\,(\varepsilon) = \begin{cases} 1 - e^{-\left(\frac{x}{\lambda}\right)^k} & x \geq 0 \\ 0 & x \geq 0 \end{cases} \tag{10}$$

where k > 0 is the shape parameter and λ > 0 is the scale parameter of the distribution.

Fréchet distribution:

$$CDF_{\text{Fréchet}}\,(\varepsilon) = e^{-\left(\frac{x-m}{s}\right)-\alpha} \tag{11}$$

where α > 0 is a shape parameter, m (the minimum) is a location parameter and s > 0 is a scale parameter.

On the basis of a series of preliminary simulations, the Gaussian CDF provides a much more better and consistent fit to the to the cumulative duration of strain events compared to the Rayleigh, Shifted Gompertz, Weibull, and Fréchet CDFs. Thus, this function is considered for the characterization of the SWS data.

2.2 Performance Evaluation of the Self-Powered Monitoring System

The performance of the proposed damage identification system is evaluated on different infrastructure systems. The performance analysis is based on extensive numerical and experimental studies. The cases investigated for this phase are crack growth detection in steel plates under a uniaxial tension mode, distortion-induced fatigue cracking in steel bridge girders, and continuous health monitoring of pavement systems The last case is studied numerically.

2.2.1 Detection of Crack Growth in Steel Plates

According to FHWA, there are thousands of fracture-critical bridges throughout the U.S. that are susceptible to failure of their critical members. In case of failure of these certain members, the bridge will collapse, e.g., case of I-35W Bridge in Minnesota (FHWA, 2004; NTSB, 2008; Liao et al., 2011). Most of the failures that occur in bridges are related to the fracture of steel members. Fatigue cracking in steel bridges is known as the cause of the majority these failures. The fatigue cracking phenomenon occurs progressively until reaching the fracture limit of specimen. Evidently, an early detection of the damage location and size is crucial to prevent catastrophic failures. This issue has been the focus of many studies for the last decades (Malekzadeh et al., 2015). Numerous studies have used the vibration-based techniques to detect and locate damage (Doebling et al., 1996). Structural damage could be detected, located, and characterized through the examination of the change in the structural vibration response (Doebling et al., 1996). When damage occurs to a structure, the natural frequencies of the system change as the structural properties deviate from its original state. Then, the damage event can be detected based on the vibration frequencies. The information provided by the SWS is completely different with what provided by conventional vibration-based sensors. Accordingly, this phase presents the performance of the proposed system on crack growth detection in steel plates. In order to analyze the response of sensors, a series of laboratory tests are conducted on a steel plate under uniaxial tensile loading configuration. Also, different FE models of a steel plate are developed to extract the strains in different damage states. For the experimental and numerical analyses, different damage scenarios are introduced to the steel plate. The main goal is to detect the crack growth using the data provided by the SWS.

2.2.1.1 Numerical Study

The performance of the sensors is studied through a numerical study. To this aim, different 3D FE models are developed for each damage state to analyze the dynamic response of the plate under a similar cyclic loading condition to the experimental study. ABAQUS/CAE 6.11 is used for the modeling and post-processing of the results. Dynamic implicit procedure is used for the FE modeling. The length, width, span and thickness of the plate are 406.4 mm (16"), 304.8 mm (12"), 152.4 mm (6"), and 0.8 mm (1/32"), respectively. 10 sensor locations are introduced by making circular partitions with a diameter similar to PZTs (D = 12 mm). Those locations are then called by software to extract the average strain in each partition. Figure 7 shows the assembly of the plate and the meshed geometry. Damage is introduced by defining a notch at the middle of the plate. The notch size is increased in each damage state to define Intact, Damage 1, 2 and 3 modes. Cyclic displacements are applied to one edge of the plate while the other side is fixed. The plate is modeled using 19068 linear hexahedral elements of type (C3D8R) in ABAQUS. Mesh refining is adopted for the meshing strategy in order to capture high stresses and strains concentration around the crack tip. The used material for the plate is steel with elastic modulus (E), Poisson’s ratio (ν) and density equal to 200 GPa, 0.3 and 7800 kg/m^3, respectively.

Figure 8 shows the FE results for different damage states. The maximum strain at crack tip increased from 729 microstrain to 1100 microstrain between the first damage state (17 mm) and the last damage state (45 mm). As shown in Figure 8, the strain field expands around the crack region as the notch length increases. Accordingly, Sensing nodes 4-7 are subjected to higher deformations and will deliver higher voltage when damage progresses.

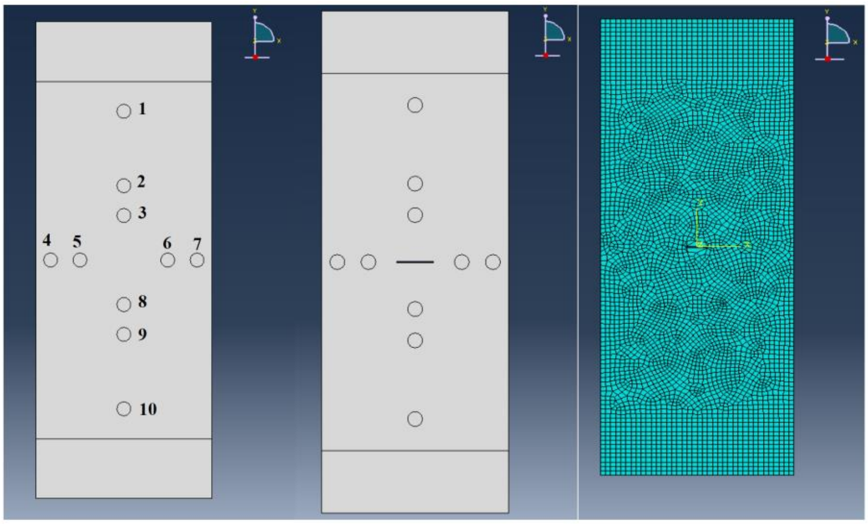
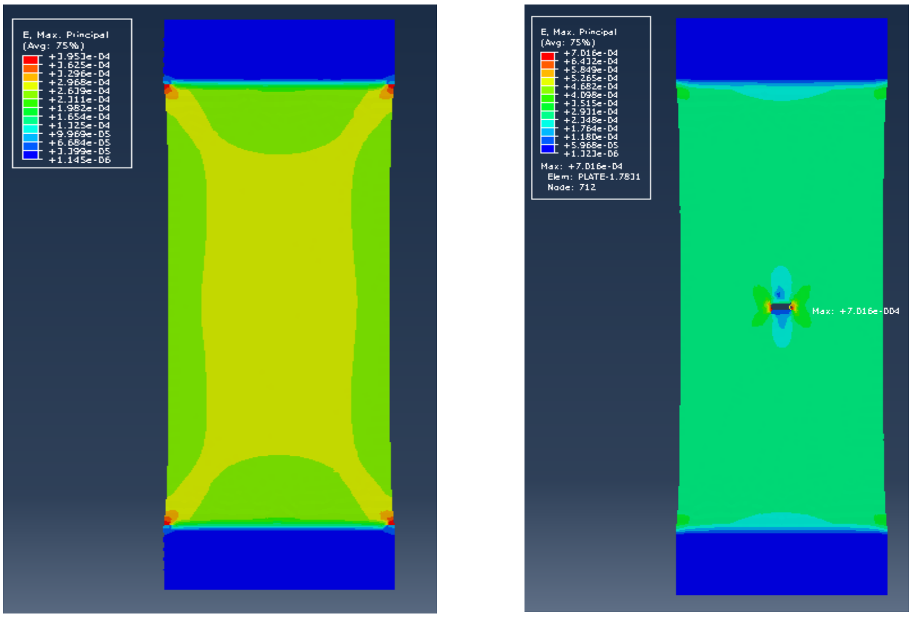
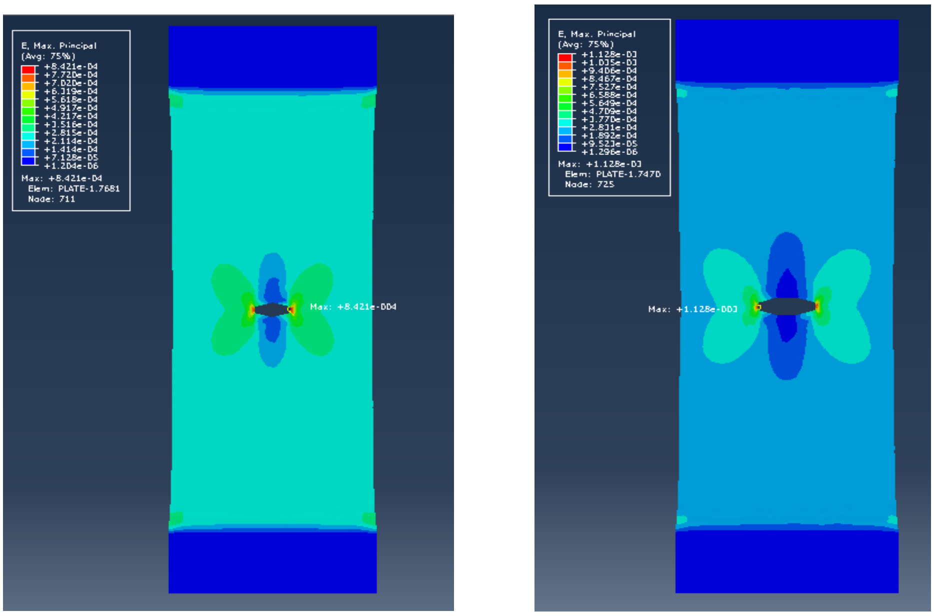


Figure 7. Assembly and meshing of the plate



(a) Intact plate (b) Damage 1



(c) Damage 2 (d) Damage 3

Figure 8. The FE results for different damage states introduced to the plate

2.2.1.2 Experimental Study

The second step of this work is focused on determination of the time history of the voltage for different damage states in order to calculate the cumulative time distribution given by an actual SWS. In this study, the specimen is tested under a uniaxial tension mode.

The experimental setup is schematically shown in Figure 9. The steel plate had 406.4 mm (16") length, 304.8 mm (12") span, 152.4 mm (6") width, and 0.8 mm thickness (1/32"). The material used is the A-32 steel grade. As seen in Figure 26, all the degrees of freedom are constrained on the top edge of the plate and displacement is applied to the lower edge. The horizontal displacements and the in-plane rotations are also constrained. In order to satisfy the boundary conditions, four thick steel plates, 50.8 mm × 152.4 mm (2" × 6"), are installed at the top and bottom of the plate. Six bolts, each 9.5mm (3/8") diameter, are used to restrain the degree the horizontal and rotational degree of freedoms (Figure 9).

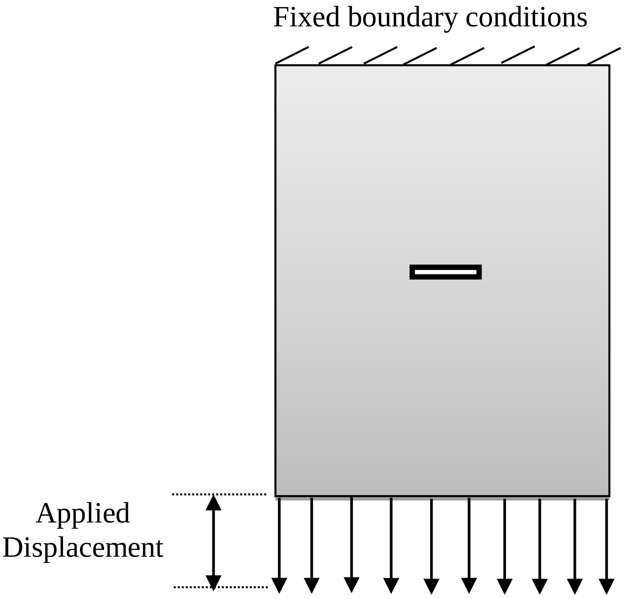


Figure 9. Uniaxial tension test setup

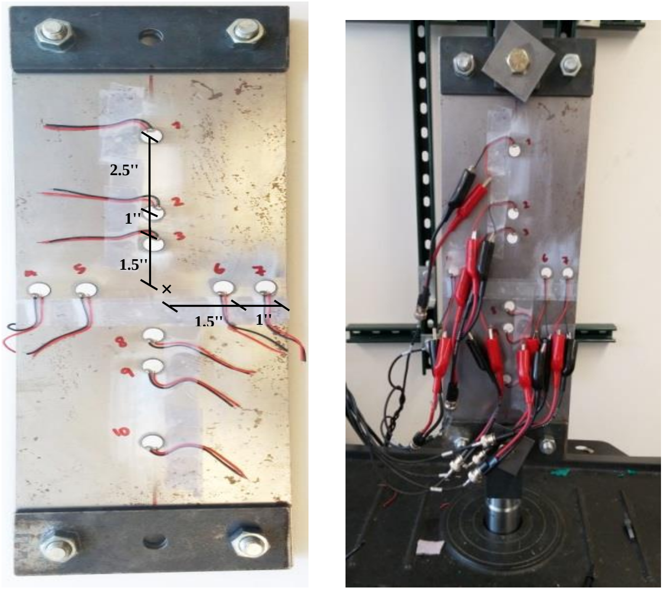
Ten PZT ceramic transducers are installed on the plate in two directions: along the crack and perpendicular to the crack direction (Figure 10). The PZT devices use the piezoelectric effect to convert the applied mechanical strain to electrical charge. The sensor locations are determined based on an FE analysis. On this basis, 3D FE models are developed to find the regions with maximum stress and strain concentration and to explore the size of the crack-induced stress zone. The FE simulations are performed using ABAQUS/CAE 6.11.

Evidently, higher voltage delivered by the PZT indicates a higher strain at the sensor location. According to the principles of fracture mechanics, the maximum stress concentration is located at the crack tip. Thus, PZTs 5 and 6 are placed along the direction of the crack. The rest of the sensors are placed in different positions from the crack area to test their resolution. The term resolution refers to the distance from the crack zone that the sensor can detect changes in voltage due to damage progression. Figure 11 shows the stress distribution on the plate for a 30 mm crack. As it is seen, the maximum stress field is located near the tip of the crack. Therefore PZTs 4, 5, 6 and 7 are expected to detect higher voltage compared to other PZTs. The rest of the PZTs are placed in other parts of the plate to check their sensitivity to damage growth.

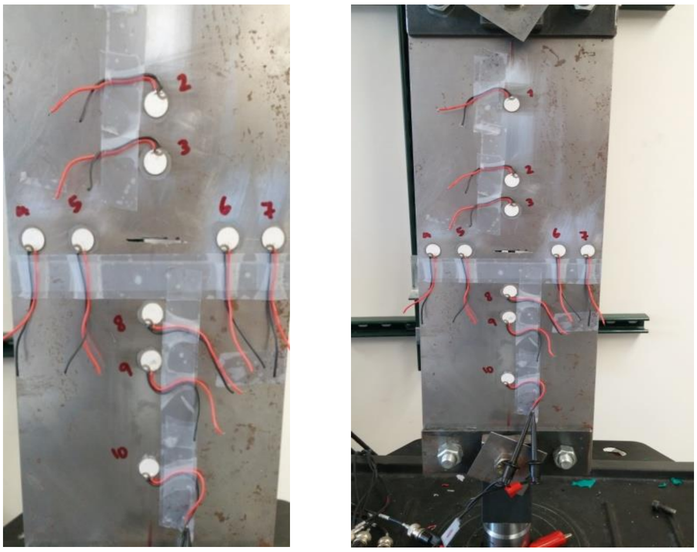
The tests are performed at 2 and 5 Hz loading frequencies for 0.05 and 0.08 mm amplitudes. Herein, the results for 2 Hz and 0.08 mm displacement are presented. For the experiments, an MTS servo hydraulic machine is used in a displacement control mode. Before starting each test, a pre-load equal to 4 kN is applied to the plate to ensure that it is seated on the fixture. Thereafter, a cyclic displacement is applied to the edge of the specimen as shown in Figure 12. The PZTs outputted voltage is read on NI 9220 data acquisition system (with 1 GΩ impedance) in parallel with a resistor with impedance similar to the SWSs. The module is controlled through a LabView program to generate the required signal.

Damage is introduced by making a notch at the middle of the plate. The damage states are defined by increasing the notch size (2a) as follows:

- Intact: 2a= 0 mm (Intact plate)
- Damage 1: 2a = 17 mm (Initial notch)
- Damage 2: 2a = 30 mm
- Damage 3: 2a = 45 mm



(a) Intact plate



(b) Notched plate

Figure 10. Test setup and sensors locations for the plate

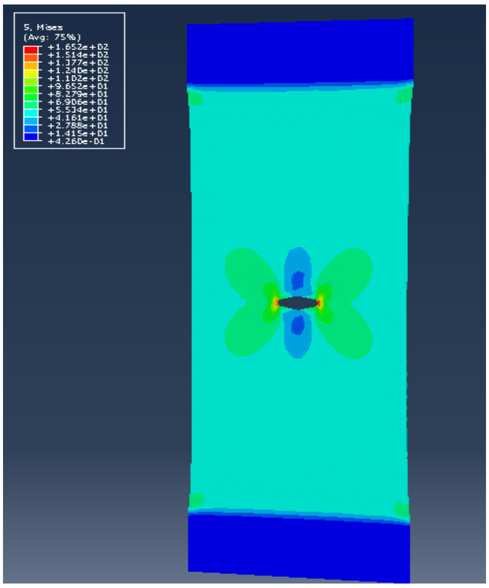


Figure 11. FE simulations for the determination of approximate sensor locations on the plate

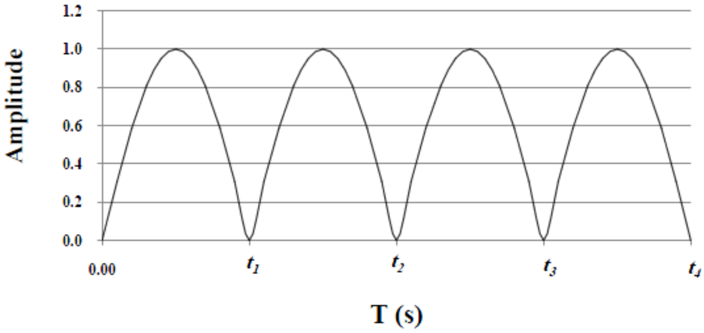


Figure 12. The applied cyclic displacement amplitude

The chosen PZTs are PZT-5A ceramic discs with a 12 mm diameter. The PZT properties are shown in Table 1. In order to attach to the PZT discs to the steel plate, different adhesive are tested. Among different adhesives tested (e.g. HBM X-60 cement paste, J-B SteelStik Epoxy, and CC-33A adhesive), CC-33A epoxy is found to perform better.

Table 2 shows the maximum voltage outputted by PZTs for different damage states. According to the results, PZTs 5 and 6 deliver the maximum voltage followed by PZTs 4 and 7. For these cases, the voltage increases as the damage progress. This is more evident for PZTs 5 and 6 that are located near the crack tip. As mentioned before, the stress concentration around the crack tip increases as the crack length increases. The voltage delivered by PZTs 1-3 and 8-10 does not have a clear trend. This is because they are far from the damage or the stress distribution at these nodes does not have a steady increasing or decreasing trend. However, the maximum voltage is delivered by PZTs 5 and 6 for Damage state 3. For these cases, the outputted voltage is equal to 10.46 volts which is the maximum voltage input range of the data acquisition system.

Table 1. The PZT properties

Parameter		Value
Diameter		12 mm
Thickness		0.6 mm
Piezo material		PZT-5A
Resonant frequency		3.4MHz±5%
Static capacitance (C_s)		2.5nF ±30%
Piezoelectric constant	d_{33}	450×10^{-12}m/v
	d_{31}	-190×10^{-12}m/v
Density		7.8g/cm3
Electromechanical coupling coefficient	K_p	0.63
	K_t	0.42
	K_{31}	0.35

Table 2. Maximum voltage delivered by the PZT discs installed on the plate for 0.08 mm displacement and 2 Hz loading frequency

	Maximum voltage (volt)			
Sensor #	Intact	Damage 1	Damage 2	Damage 3
PZT 1	6.90	6.85	6.94	6.35
PZT 2	6.68	6.92	7.47	6.09
PZT 3	6.53	6.49	6.35	4.78
PZT 4	5.58	5.92	7.16	7.25
PZT 5	7.21	8.15	9.73	10.46
PZT 6	6.40	7.04	8.93	10.46
PZT 7	5.75	6.46	7.21	7.79
PZT 8	6.85	7.19	6.36	5.30
PZT 9	5.62	5.84	6.06	5.46
PZT 10	5.48	5.41	6.33	6.42

2.2.1.3 Damage Growth Detection Based on the FE Results

The performance of the sensor is verified using the FE simulations followed by the laboratory test results. It should be noted that the voltage supplied by piezoelectric transducers is proportional to experienced strain (Sirohi and Chopra, 2001). Consequently, using voltage or corresponding strain values does not affect the damage analysis trends. For the analysis, the voltage delivered by PZTs and the strains extracted from the FE simulations are recorded and analyzed.

For the FE simulations, the results are obtained for 2 Hz and 0.08 mm displacement. A smart pebble wireless sensor has a series of memory cells (gates). Each of these gates cumulatively stores the duration of strain events at a preselected levels, experienced at the sensing node location. In general, the number of gates is dependent on the nature of the problem and the material. In this study, 7 strain levels are defined to efficiently cover the lower and upper limits of the extracted strain values. The minimum level of strains to be captured by piezoelectric transducers is about 20.00 $\mu\varepsilon$. On the other hand, the maximum of the strain value extracted from the FE simulations is about 315 $\mu\varepsilon$. Considering a reasonable value of 7 strain levels for the gates, the difference between the strain levels is 50 $\mu\varepsilon$. Table 3 presents the preselected strain levels for the plate. Using the strain histories of all elements, a script is written in MATLAB to perform the following tasks:

a) Takes the strain-time data from ABAQUS and measures the duration of events at the strain levels defined in Table 3.

b) Fits CDF given in Eq. (6) to the cumulative time of occurrences at predetermined strain levels obtained from the first step.

c) Reports the μ and σ for the data acquisition nodes for the intact and damaged models

Using the μ and σ values, the PDF plot corresponding to each sensing node is obtained and shown in Figure 13. As seen in Figure 13, the values of μ and σ, and accordingly the shape of PDFs change due to the damage progression. Sensors 5 and 6 provide the most reasonable results. In fact, the PDFs for these sensors shift to the left (μ decreases) and they expand (σ increases) as damage progresses. The same is true for Sensors 4 and 7 (Figure 14). Since these sensors are not as close as Sensors 5 and 6 to the crack zone, the change in PDF is less obvious.

Table 3. The preselected strain levels considered for the plate FE analysis

Gate number	Strain Level (με)
1	20
2	70
3	120
4	170
5	220
6	270
7	320

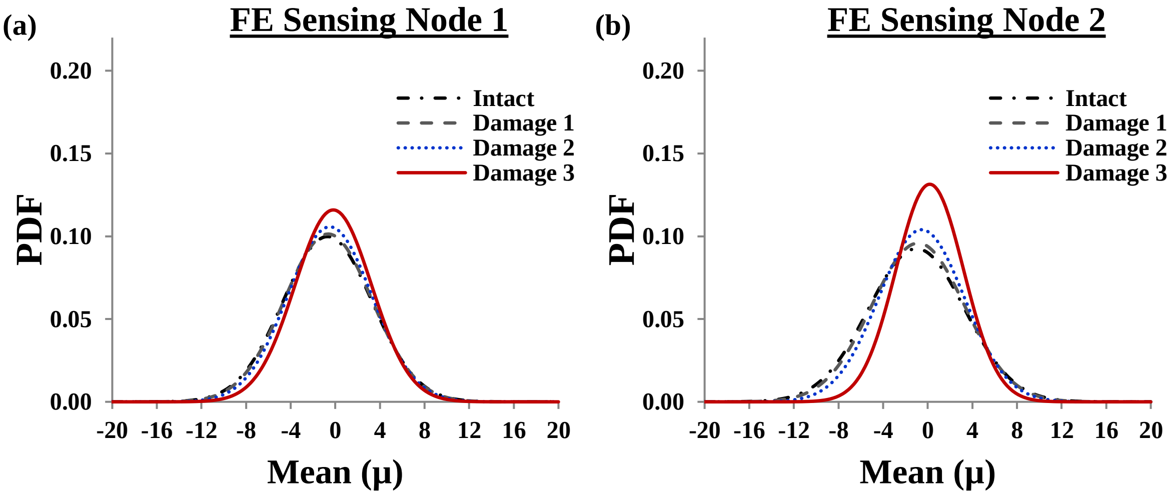


Figure 13. Change of PDFs curves due to damage progression based on the plate FE analysis

Figure 13. (cont'd)

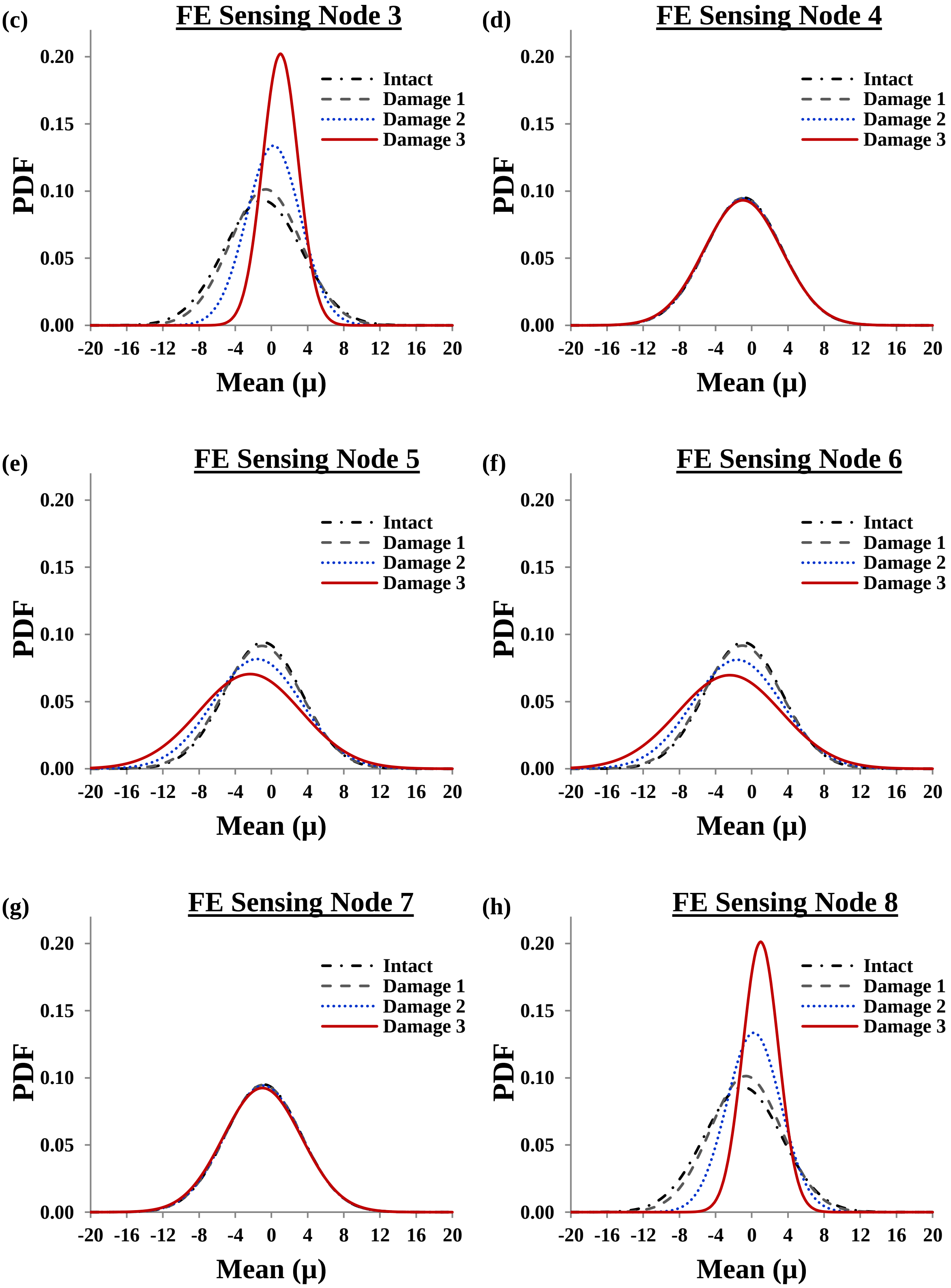
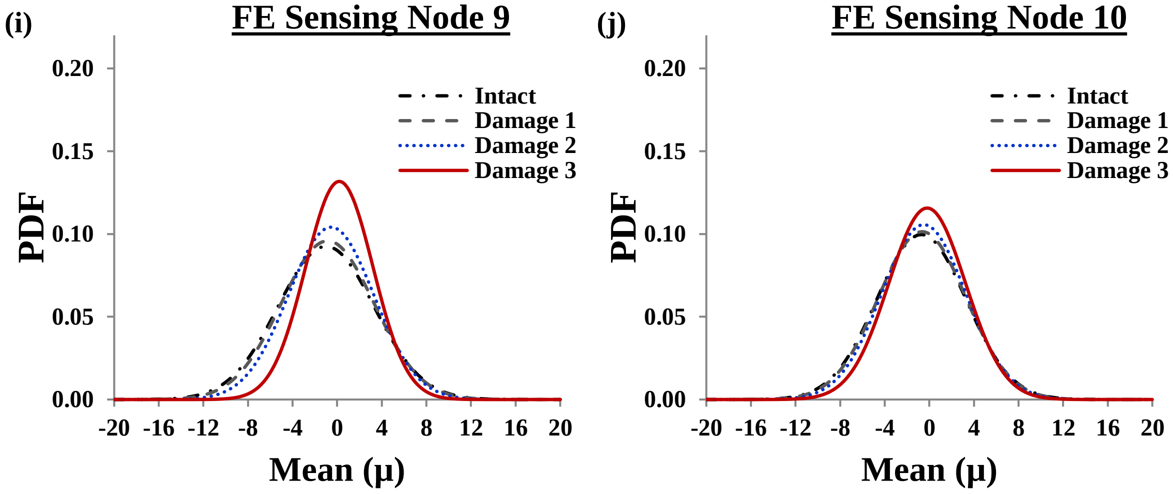
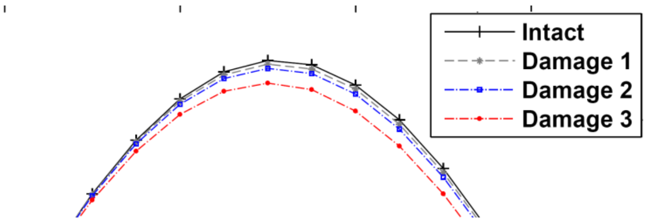
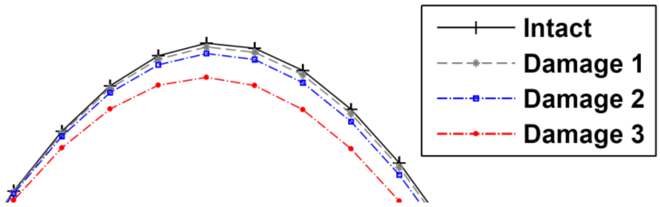


Figure 13. (cont'd)





(a) Sensor 4



(b) Sensor 7

Figure 14. High resolution illustrations of PDFs for Sensors 4 and 7 based on the plate FE analysis

2.2.1.4 Damage Growth Detection Based on the Experimental Results

By considering the 7 voltage levels of the sensor, the minimum and maximum voltage thresholds are set to 4 and 10 volts, respectively. Table 4 presents the preselected voltage levels for the plate. The PDF plot corresponding to each sensor is obtained and shown in Figure 15. As seen in this figure, the shapes of PDFs have changed due to damage progression. This indicates that damage growth can be monitored by the changes of PDFs even outside of the high stress concentration region. Referring to Figure 15, it can be seen that PZTs 4 to 7 are good indicators of the damage progression. In these cases, μ decreases and σ increases by transitioning from intact to damaged mode. Accordingly, the PDFs shift to left, their width increases and their height decreases due to damage progression. This is in agreement with the trends previously observed from the numerical study. For the sensors that are not located along the crack, the PDFs change but they do not have a clear trend.

Table 4. The preselected voltage levels for the plate experimental study

Gate number	Voltage Level (volts)
1	4.00
2	5.00
3	6.00
4	7.00
5	8.00
6	9.00
7	10.00

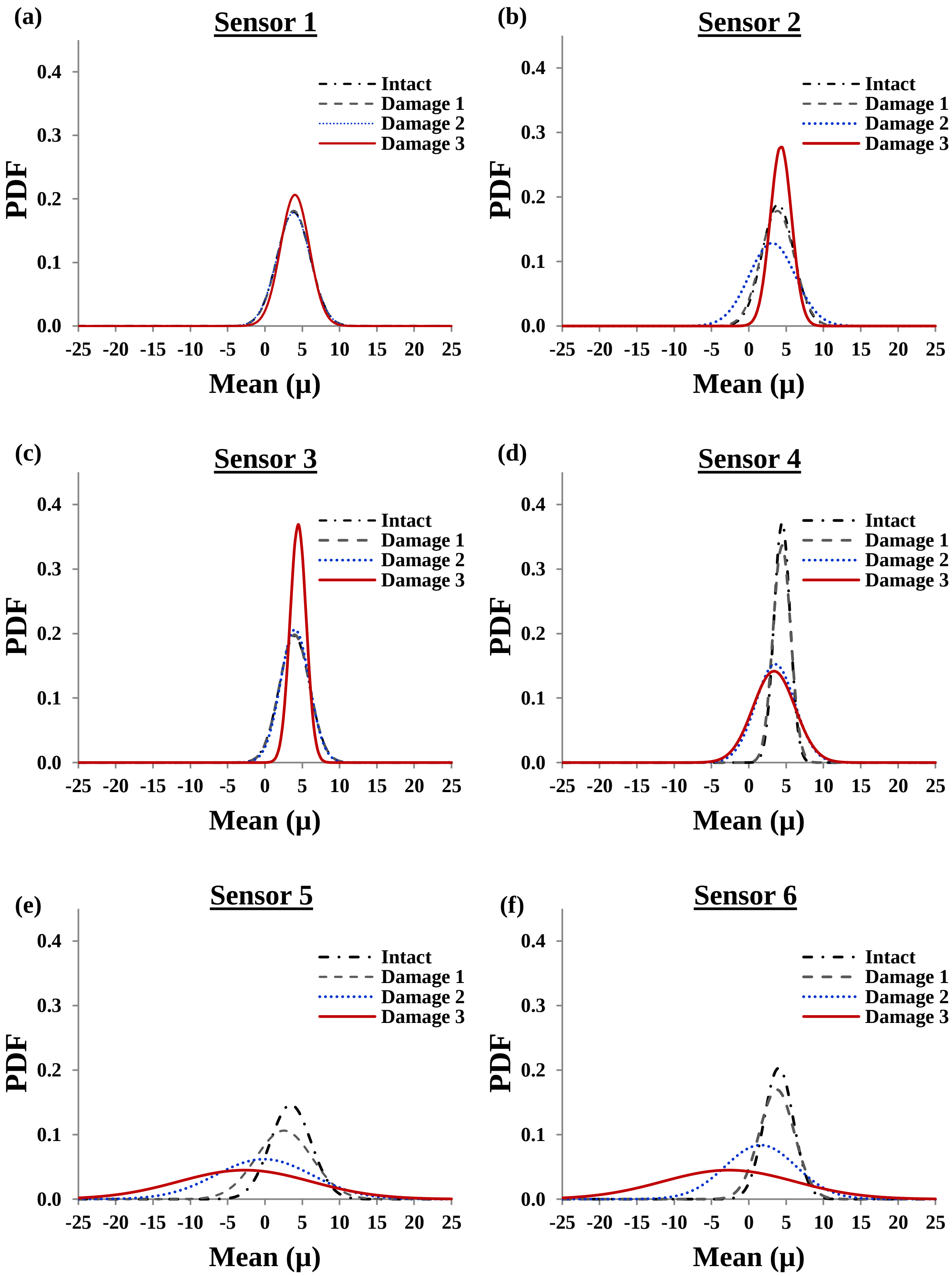
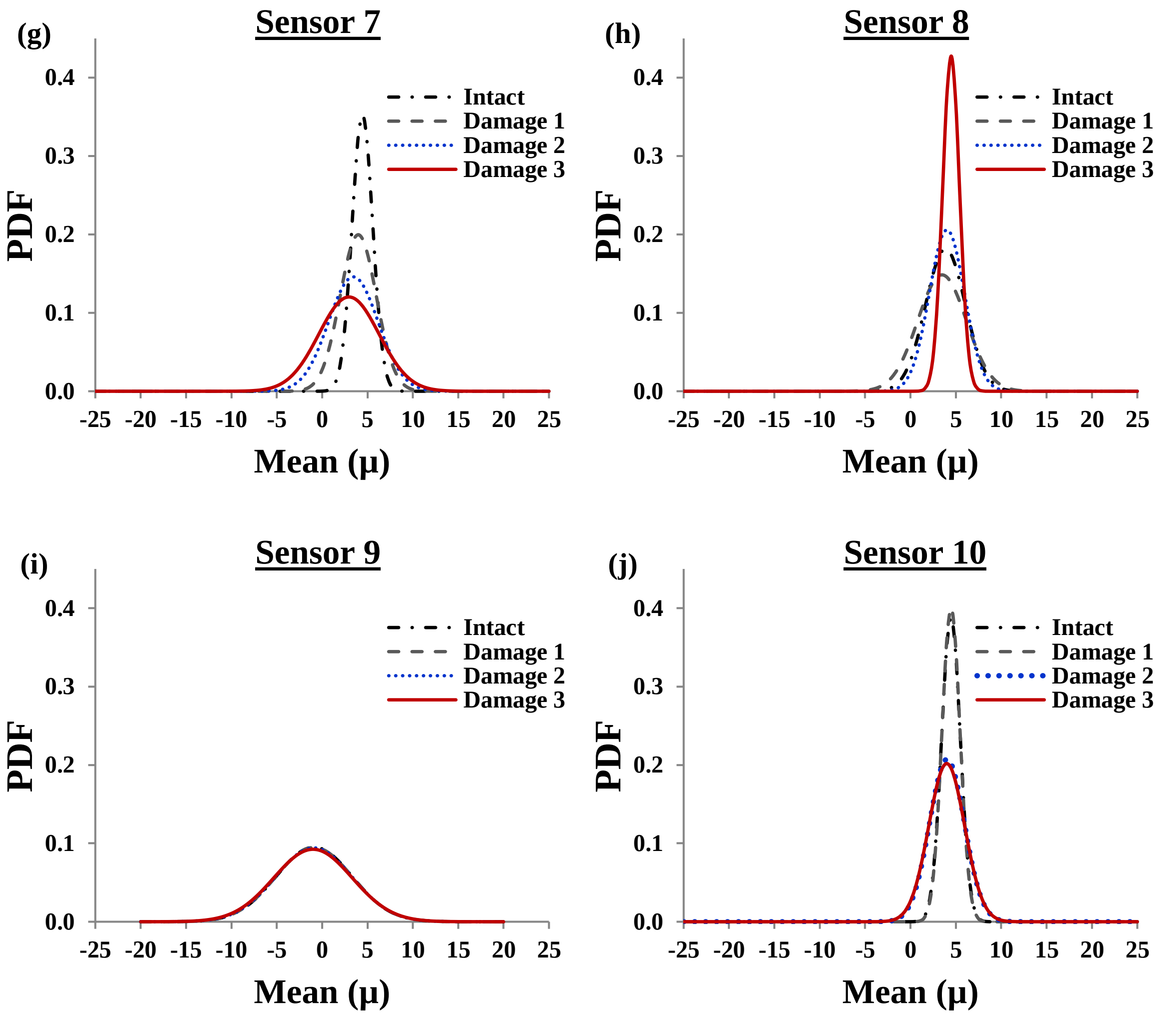


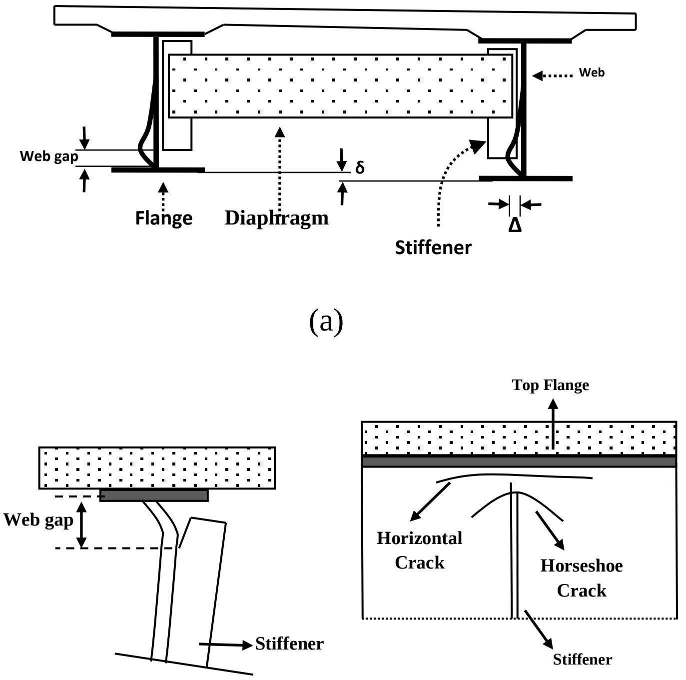
Figure 15. Change of PDFs curves due to damage progression based on the plate experimental study

Figure 15. (cont'd)



2.2.2 Distortion-Induced Fatigue Cracking in Bridge Girders

A major concern in maintenance and preservation of steel bridges is cracking of structural members. Fatigue cracking is one of the most important phenomena affecting the structural integrity and performance of welded steel bridges (Fisher et al., 1980; Fisher, 1984). Occurrence of fatigue cracks is specifically of great importance for the steel bridges built before the 1970s. This is because the fatigue design specifications are not appropriately defined until 1970s and 1980s (Zhao and Roddis 2004). In general, fatigue cracks can occur in welded steel bridges due to low fatigue resistance structural members, members with large initial defects, members subjected to out-of-plane distortion, and details at end restraints and flange terminations (Fisher and Menzemer, 1990). However, out-of-plane distortion is known as the major source of fatigue cracks leading to severe structural deficiency. Distortion-induced fatigue is more about the detailing issue and needs much fewer stress cycles to develop compared to other crack types (Zhao and Roddis, 2004). Different factors can cause out-of-plane distortion such as impact of vehicles on an expansion joint not perpendicular to the traffic flow, thermal forces on skewed and horizontally curved bridges, differential deflection of the adjacent beams, etc (Juntunen 1998). The mechanism of fatigue crack formation is schematically shown in Figure 16. A feature of the diaphragms and cross frames is primarily to distribute loads among main elements (Figure 16a). These elements are fastened to transverse stiffeners welded to the girder web. For many of the bridges designed before 1989 AASHTO Standard Specifications (AASHTO, 1996), no connection is considered between the stiffeners and the steel girder flanges. Consequently, out-of-plane fatigue cracks are developing at small web gaps at the girder flanges, webs, and stiffener plate connections due to out-of-plane bending of the girder web (Fisher, 1984). Accordingly, stress state increases in the web near the weld and fatigue cracks initiate in the heat-affected zone of the weld near its toe. Such distortion-induced fatigue cracks may occur as horizontal or horseshoe cracks at the top or bottom of girder diaphragm connections (Figure 16c). The cracks usually propagate away from the weld. The direction of the crack propagation changes as the crack grows in length (Fisher et al., 1985; Elewa, 2004).



(a)

(b) (c)

Figure 16. Distortion-Induced Fatigue Cracking: (a) web distortion due to differential displacement between two girders; (b) out-of-plane distortion; (c) fatigue cracks

Several retrofitting methods have been proposed to deal with this type of damage (Elewa, 2004). However, selection of an appropriate repair strategy is complicated and depends on many factors. On the other hand, there are no major predictive models to assess the damage caused by distortion-related cracking (Elewa, 2004). Distortion-induced fatigue cracking is a progressive phenomenon. Thus, from a SHM perspective, it is crucial to detect the damage progression at early stages so that severe damage to the bridge structure can be prevented. This study presents new results on detection of fatigue cracking of steel bridges based on the SWS data. A 3D FE analysis is performed to obtain the structural response of the girder. The fatigue life of the girder is determined based on J-integral concept and Paris Law. Several damage states are defined by extending the fatigue crack propagation lengths. Thereafter, features extracted from the strain data for individual and a group of sensing nodes are used to detect various damage scenarios.

2.2.2.1 Numerical Study

For the analysis, a girder structurally similar to an existing highway steel bridge girder (I-96/M-52) in Webberville, Michigan, is chosen. The I-96/M-52 bridge is typical of many bridges in Michigan. There are several reasons behind using the FE method for the distortion-induced fatigue analysis. While current design specifications cannot be used for the estimation of the secondary stresses over the girder web caused due to out-of-plane displacement of the web gap, FE can be considered as an efficient tool to achieve this aim. The FE numerical simulations are much less costly than performing cumbersome field or laboratory investigations. Also, it is possible to update the mathematical models in FE based on the changes of the geometry or loading environment (Bhargava and Roddis, 2007). However, a series of FE simulations are conducted for this case and the derived models are considered as representatives of the real structure. Static strain measurements at different locations are used to extract the damage detection features. Herein, the structure under consideration is assumed to be linearly elastic and is analyzed as 3D FE models using ABAQUS Version 6.12-3.

Geometry, loading, and boundary conditions: The length of the modeled girders is considered 7.62 m (300"). The girders spacing is taken 1.943 m (76"). The thickness, width and length of the stiffeners plates are 12.7 mm (0.5"), 140 mm (5.5"), and 812 mm (320"), respectively. The web gap length is taken 57 mm (2.25"). The loading is applied at the cut edge of the lower flanges. The loading is a vertical displacement of 5 mm (3/16") for the left outer girder and 15 mm (9/16") of the right outer girder. The geometry and loading of the girders are shown in Figure 17. The steel girders are modeled as simply supported beams. The translations along the x and z directions at the cut edge flange nodes and the rotation with respect to the x-axis are restrained. For the top and lower flanges, the displacement in x and z direction and the rotation around the x-axis are also fixed.

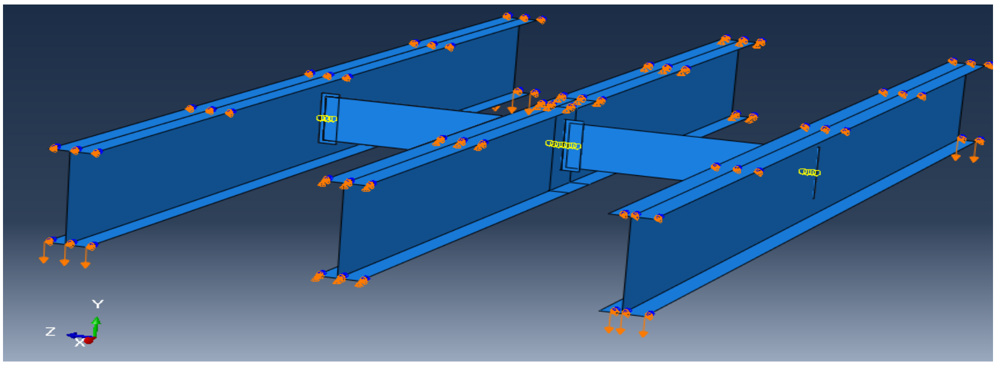


Figure 17. Geometry and loading of the girders

Material: The used material for the girders, diaphragms and stiffeners, is steel with E, v and density equal to 200 GPa, 0.3 and 7800 kg/m^3, respectively.

Model meshing: For the model developed in this work, linear quadrilateral shell elements of type S4R are used. At the web gap area, the mesh size is refined to below 4 mm to capture the effect of the stress concentration (Figure 18). Further refinement is also required to confirm the convergence of the model.

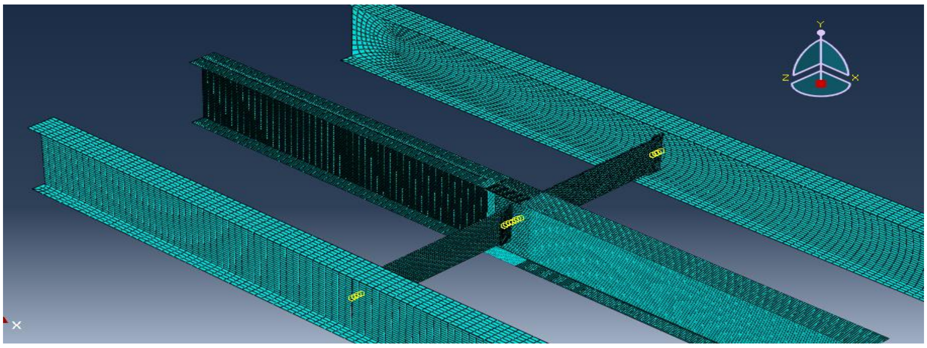
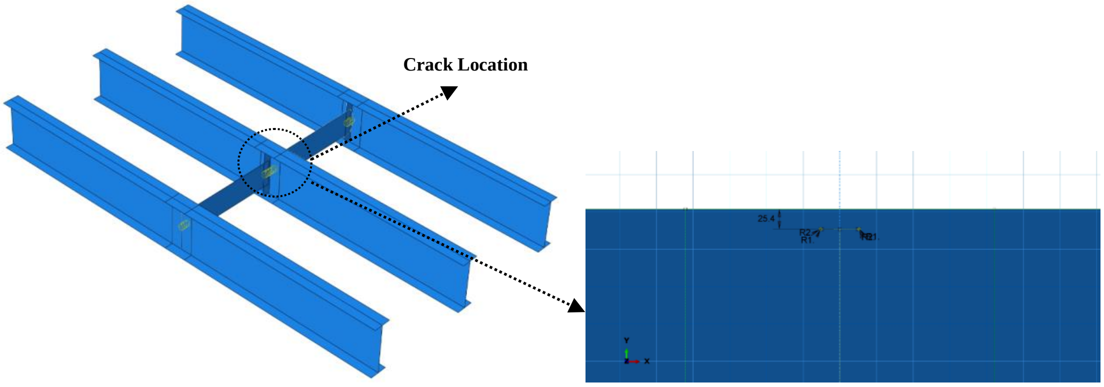


Figure 18. Meshed geometry of the girders

Determination of the Fatigue Life of the Girder: The FE results are further refined and extended to include fatigue damage accumulation, crack initiation and crack propagation. Different damage states are defined by increasing the crack length. Then, features extracted from the strain data for individual and groups of sensing nodes are used to detect various damage scenarios. Previous studies showed that the top flange web gap area is more susceptible to distortion-induced fatigue than other regions in existing welded plate girder bridges (Bhargava and Roddis, 2007). Therefore, a horizontal crack is considered in this region at a distance of 25.4 mm (1") from the upper flange of the central girder. The model assembly and crack location is given in Figure 19. The initial crack length is taken 10 mm.



(a) (b)

Figure 19. (a) Assembly of the girder model for the fatigue analysis (b) crack location and properties

The crack length is changed between 10 mm to 100 mm with 10 mm steps. The J-integral concept is used to estimate the energy release rate for each crack state (or length). The relation between crack length, energy release rate and number of cycles is given by Paris Law as follows:

$$\frac{\Delta a}{\Delta N} = c\ \Delta K^{n} \tag{12}$$

where a is the crack length and N is the number of load cycles. $\frac{\Delta a}{\Delta N}$ is known as the crack growth rate and denotes the infinitesimal crack length growth per increasing number of load cycles. C and n are material constants taken as 2.50E-12 and 3.3, respectively. ΔK is the range of the stress intensity factor, i.e., the difference between the stress intensity factor at maximum (K_{max}) and minimum (K_{min}) loading. Then, the number of cycles based on Paris Law is given by:

$$N_f - N_i = \frac{a_f - a_i}{c\,\Delta K^n} \tag{13}$$

where N_f is the remaining number of cycles to fracture, a_f is the critical crack length at which instantaneous fracture will occur, and a_i is the initial crack length at which fatigue crack growth starts for the given stress range $\Delta\sigma$. The nature of applied loading is cyclic (half sine) and therefore the R-ratio (K_{min}/K_{max}) is equal to zero because K_{min}= 0. Table 5 presents the analytical results for different crack lengths. The number of cycles to propagate the crack from one stage to another stage can be found in this table. The maximum principal strains for typical crack lengths of 20, 70 and 90 mm are presented in Figures 20-22. As it is seen, the maximum strains are located at the crack tip.

Table 5. The girder fatigue analytical results for different crack lengths

a (mm)	J_{ave}	$\Delta \boldsymbol{K}$	$\frac{\Delta \boldsymbol{a}}{\Delta \boldsymbol{N}}$	$\Delta \boldsymbol{N}$	N
10	0.580888	340.8483	6.39E-09	0	0
20	0.793691	398.4197	1.07E-08	1565338	1565338
30	0.932375	431.8274	1.39E-08	935267	2500605
40	1.03625	455.2472	1.66E-08	717028	3217633
50	1.15125	479.8437	1.98E-08	602343	3819977
60	1.277398	505.4498	2.34E-08	506326	4326303
70	1.287184	507.3823	2.37E-08	426503	4752806
80	1.414595	531.9013	2.77E-08	421166	5173973
90	1.5	547.7226	3.06E-08	360427	5534400
100	-	-	-	327197	5861597

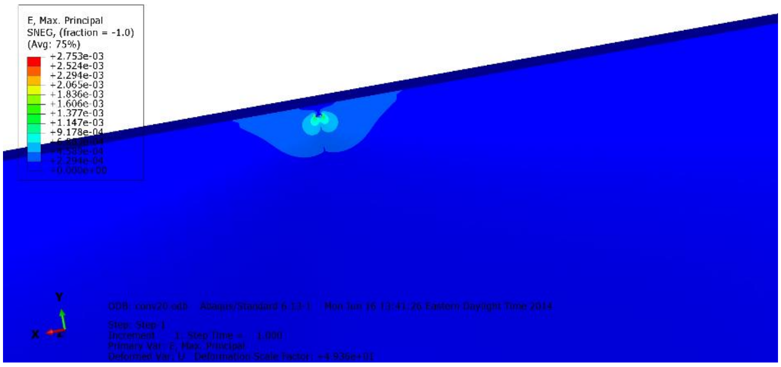


Figure 20. The FE results for the girder with 20 mm crack (Maximum principal strain = 2753 με)

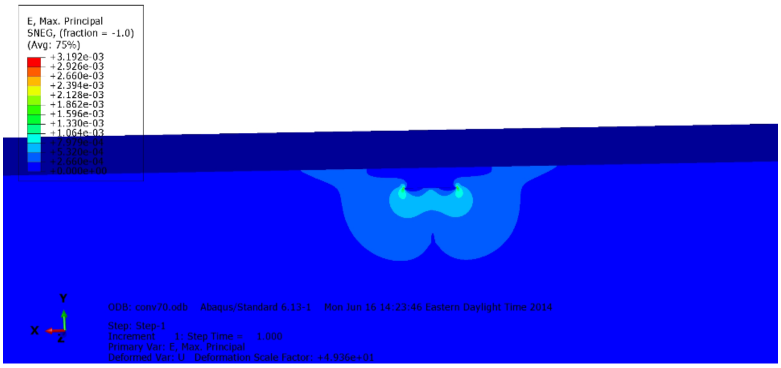


Figure 21. The FE results for the girder with 70 mm crack (Maximum principal strain = 3192 με)

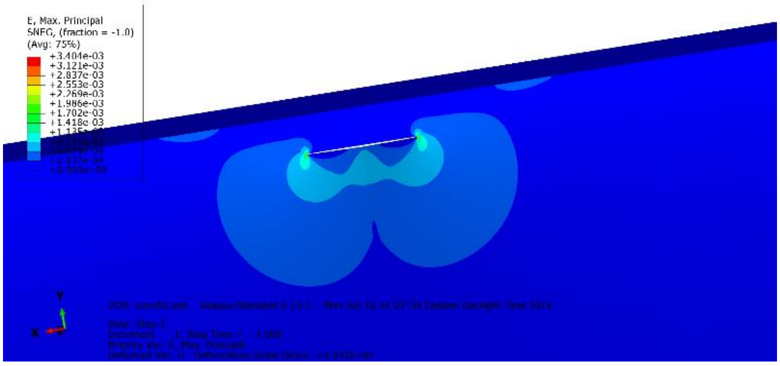


Figure 22. The FE results for the girder with 90 mm crack (Maximum principal strain = 3404 με)

2.2.2.2 Damage Detection Using the Smart Sensing Technology

For the definition of fatigue damage, 10 damage states (DS) are defined as follows: DS1 (Crack =10 mm), DS2 (Crack =20 mm), DS3 (Crack =30 mm), DS4 (Crack =40 mm), DS5 (Crack =50 mm), DS6 (Crack =60 mm), DS7 (Crack =70 mm), DS8 (Crack =80 mm), DS9 (Crack =90 mm), and DS10 (Crack =100 mm). As described before, the Paris Law and J integral concept are used to determine the number of cycles to propagate the crack from one stage to another stage (Intact to crack length equal to 100 mm) (see Table 5). Based on the results shown in Table 5, a 100 mm crack will occur after about 6 million cycles. Accordingly, the life span of the girder is divided into 6 different stages (D1 (0-1 million cycles) to D6 (5-6 million cycles)) each correspond to the condition of the girder after 1 million cycles (Figure 22). Then, the number of cycles in a specified damage state is obtained to extract the cumulative duration of strain events for that stage (Table 6). As an example, it can be observed from Table 6 that a 10 mm crack propagates to 20 mm after a 1565338 cycles. Thus, for the first million cycle shown in Figure 23, we have only damage state 1 (DS1) (Crack =10 mm). After 2 million cycles, only damage states 1 (DS1) and 2 (DS2) are occurring because 2500605 cycles are required to propagate to a 30 mm crack. Out of these 2 million cycles, 1565338 cycles correspond to DS1 and 434662 cycles to damage state DS2. These values were rounded to 565000 and 435000 as shown in Table 6. Subsequently, the duration of the strain events for 565000 cycles in DS1 and 435000 cycles in DS2 are used to extract the μ and σ parameters for the case of 1-2 million cycles (D2). This procedure is followed to obtain the features for all of the 6 million cycles.

The next stage is to define the location of data acquisition nodes (potential sensors). To this aim, 400 sensing nodes are located around the upper half of the stiffener to web contact region. Given the girder’s dimensions and at this specific resolution, 20 by 20 sensor locations with 20 mm spacing are defined in the horizontal and vertical directions. For these initial runs, the maximum number of sensing nodes is 400. The locations of the sensing nodes are illustrated as black dots in Figure 24. The diameter of the sensing nodes is equal to 10 mm. Bedsides, 10 strain levels are defined for the girder (Figure 25).

Table 6. The number of cycles corresponding to each damage state in girder fatigue analysis

	No. of Cycles in Damage State									
Cycles (Million)	**DS1 (10mm)**	**DS2 (20mm)**	**DS 3 (30mm)**	**DS4 (40mm)**	**DS5 (50mm)**	**DS6 (60mm)**	**DS7 (70mm)**	**DS8 (80mm)**	**DS9 (90mm)**	**DS10 (100mm)**
0-1 (D1)	1000000	0	0	0	0	0	0	0	0	0
1-2 (D2)	565000	435000	0	0	0	0	0	0	0	0
2-3 (D3)	0	501000	499000	0	0	0	0	0	0	0
3-4 (D4)	0	0	218000	602000	180000	0	0	0	0	0
4-5 (D5)	0	0	0	0	326000	427000	247000	0	0	0
5-6 (D6)	0	0	0	0	0	0	174000	360000	327000	138000

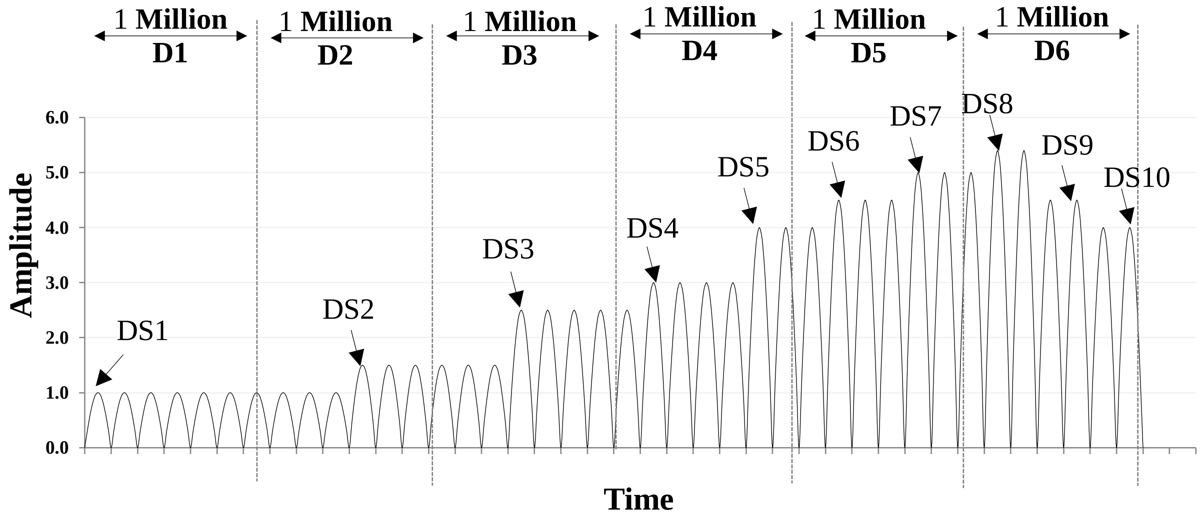
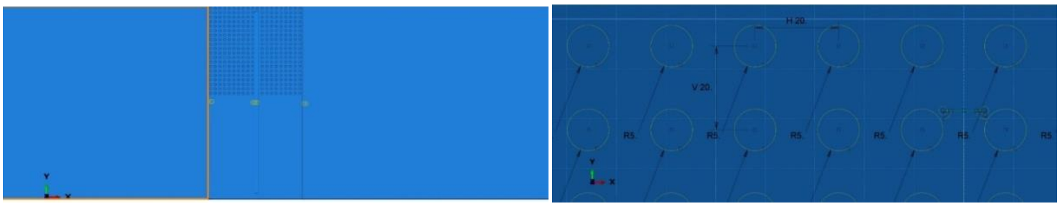


Figure 23. Schematic representation of the damage states (DS) for the girder fatigue analysis



(a) (b)

Figure 24. Configuration of the sensing nodes for the girder

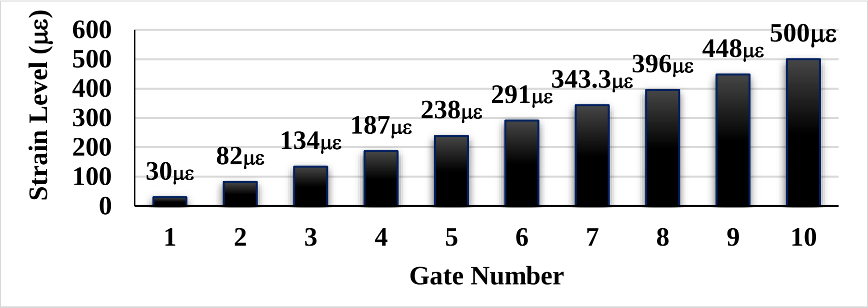


Figure 25. The preselected strain levels considered for the girder fatigue analysis

The selected sensing nodes are schematically shown in Figure 26. As it is seen, a higher density is considered around crack. In order to evaluate the information provided by the sensing nodes, the μ and σ values for all of the 400 sensing nodes for different damage scenarios are calculated. Out of the available data, the PDF graphs are plotted for some of the selected sensing nodes distributed throughout the girder. The PDF plots for some of the selected nodes are shown in Figures 27 to 34. As seen in these figures, for the sensing nodes located at the corners and far from the damage location (e.g. Sensing nodes 1, 20, 41, 60), there is no notable sense of damage as the PDFs are fairly identical. By getting closer to the damage zone, the strain patterns remarkably change and therefore the shape of the PDFs transforms from D1 (0-1 million cycles) to D2 (5-6 million cycles). It is seen that the mean of the distribution, i.e. μ, decreases and the standard deviation, i.e. σ, increases. The important observation is that the PDFs shift to left and their width increases due to damage progression. However, this is not completely clear for all of the sensing nodes. Only those that are at a specified distance from the crack (e.g. Sensing nodes 8, 14, 48, 53) generate sound results. For the sensing nodes adjacent to the crack (e.g. Sensing nodes 10, 12), there are no rational trends due to the singularities at those locations. The interesting point that can be concluded from the results is that the PDFs give a good insight into the location of damage. If there is no sense of damage, they do not change. But if the sensors are very close to the damage zone, the outcomes are becoming chaotic. On the other hand, if the sensors are not too far from the damage nor adjacent to the crack zone, the graphs shift backward and expand due to damage progression.

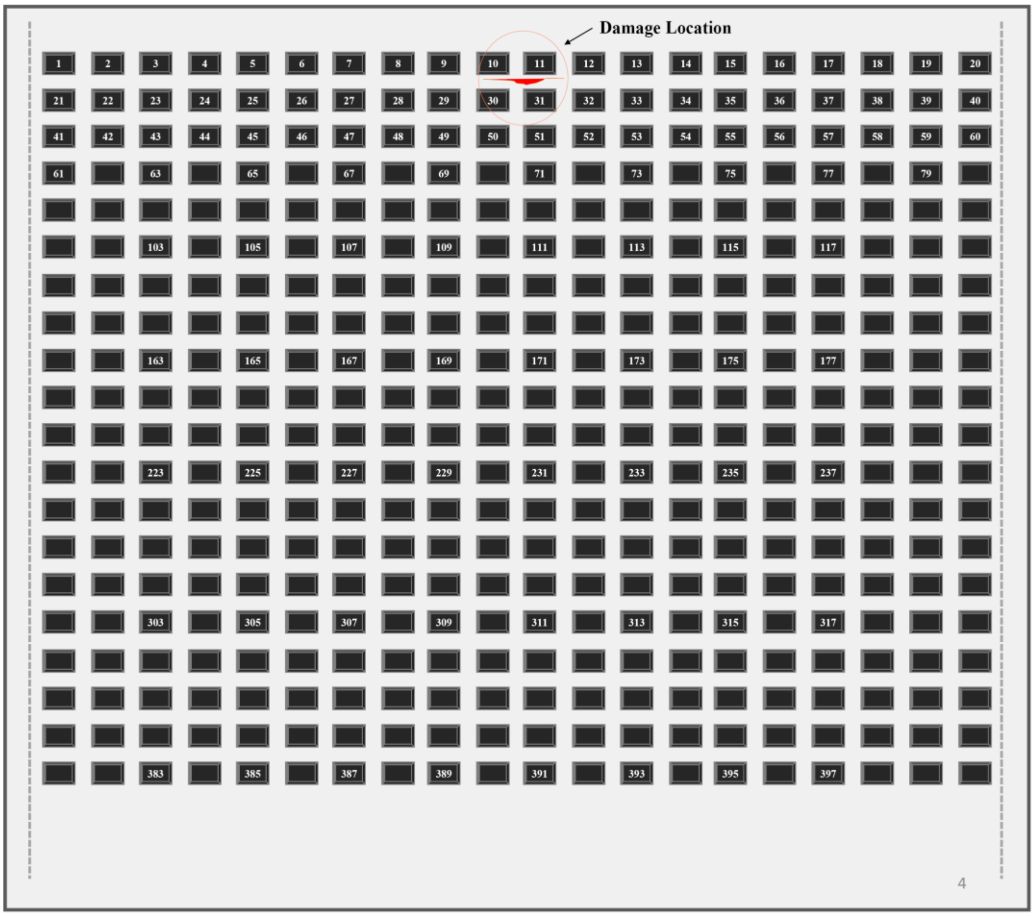


Figure 26. A schematic representation of the sensing nodes configuration for the girder

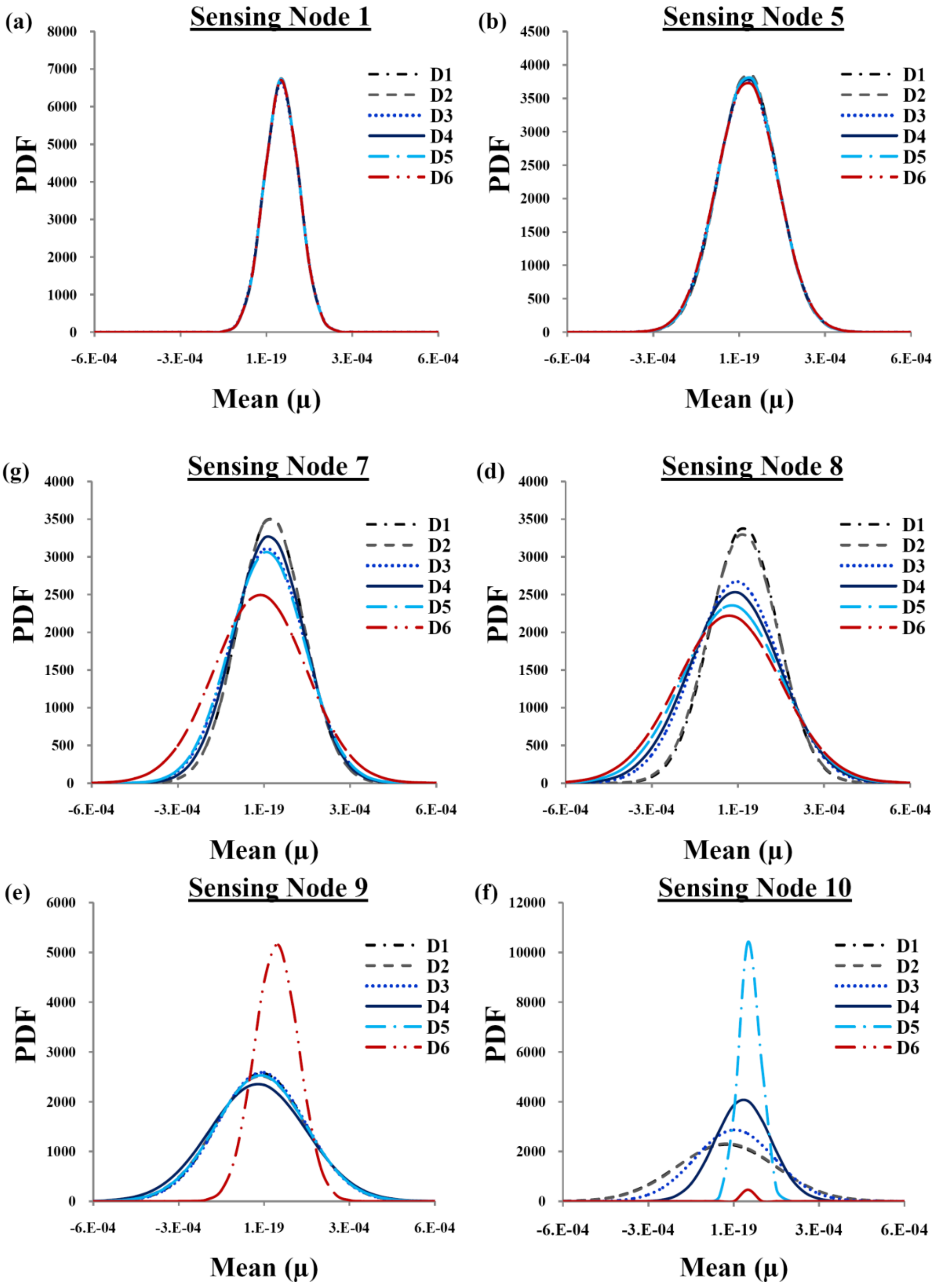
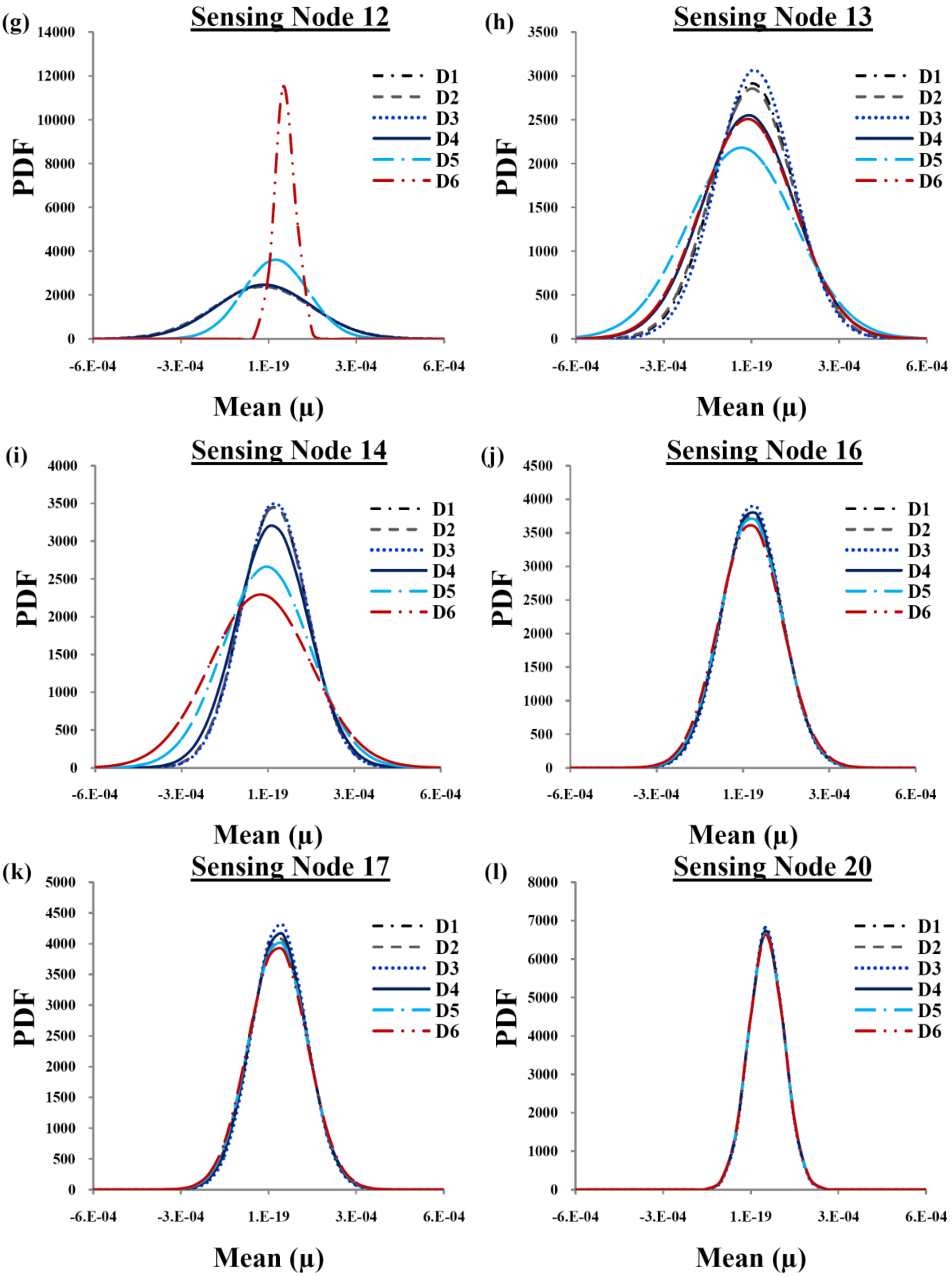


Figure 27. The PDF plots for the selected sensors (1 to 20) obtained from the girder fatigue analysis

Figure 27. (cont'd)



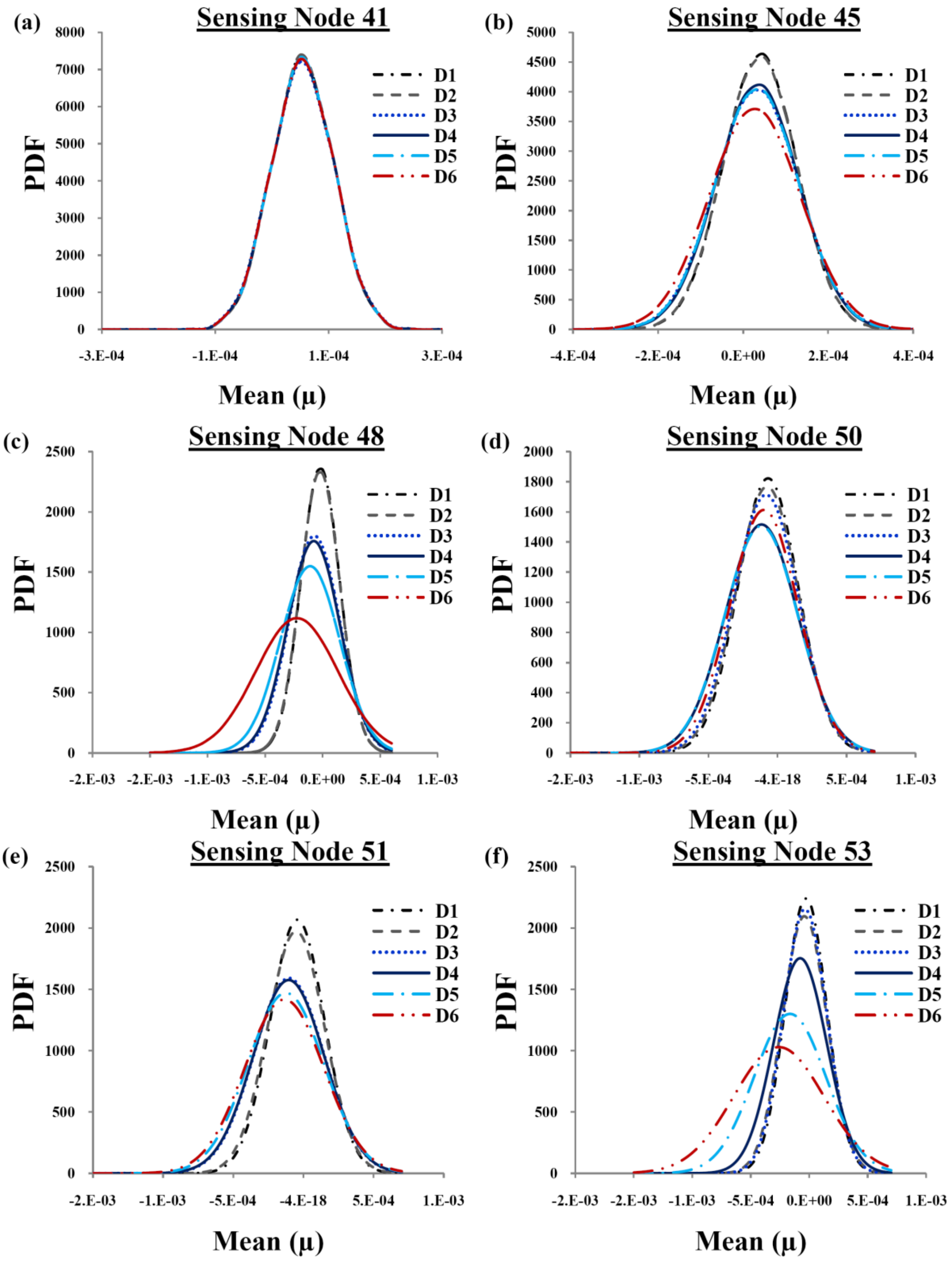
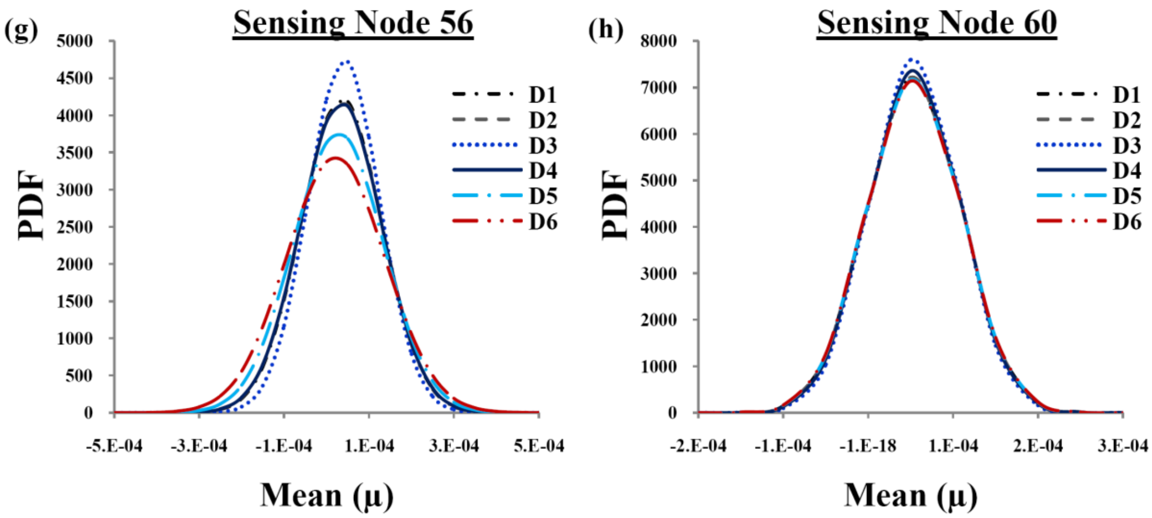


Figure 28. The PDF plots for the selected sensors (41 to 60) obtained from the girder fatigue

Figure 28. (cont'd)



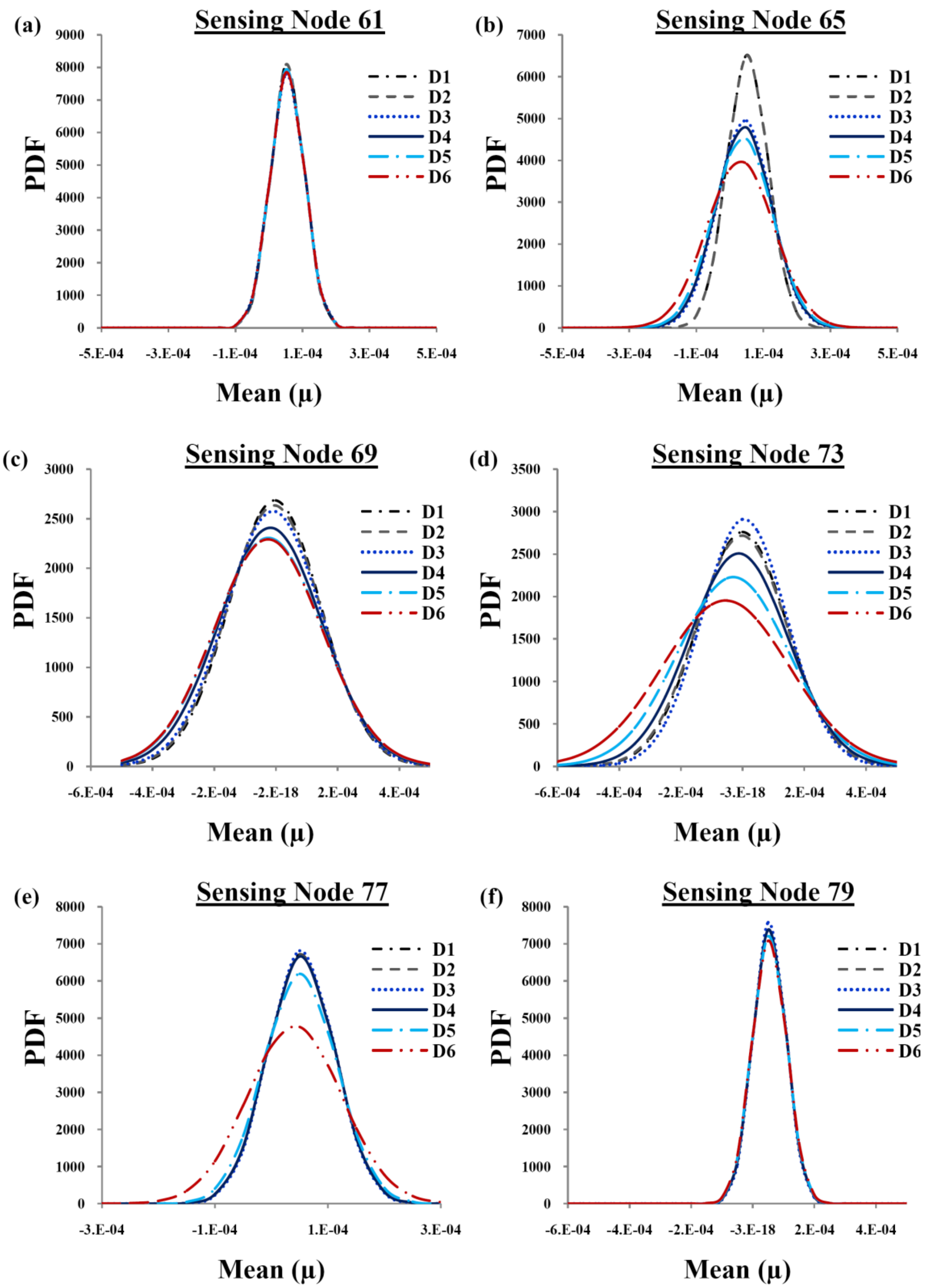


Figure 29. The PDF plots for the selected sensors (61 to 80) obtained from the girder fatigue analysis

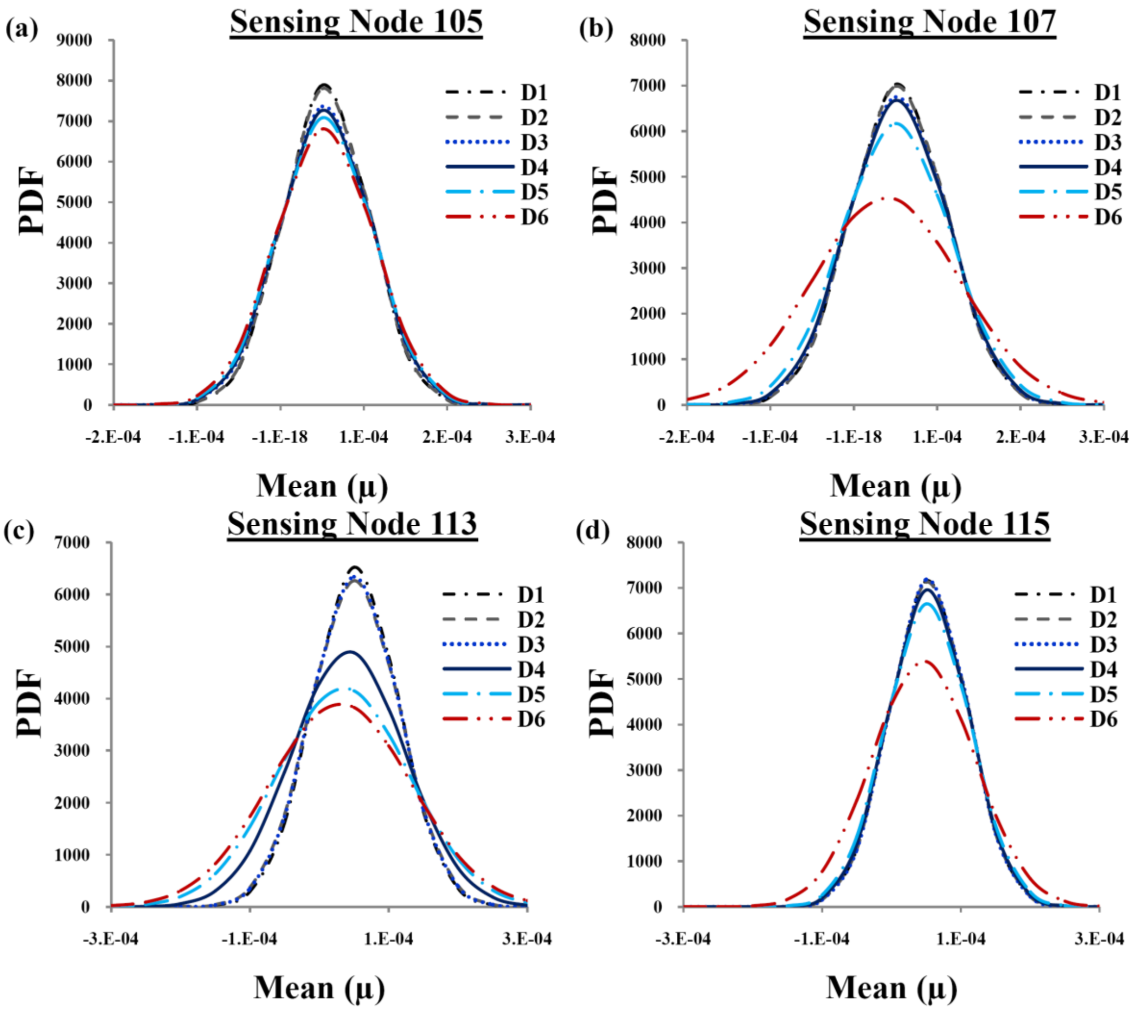


Figure 30. The PDF plots for the selected sensors (101 to 120) obtained from the girder fatigue analysis

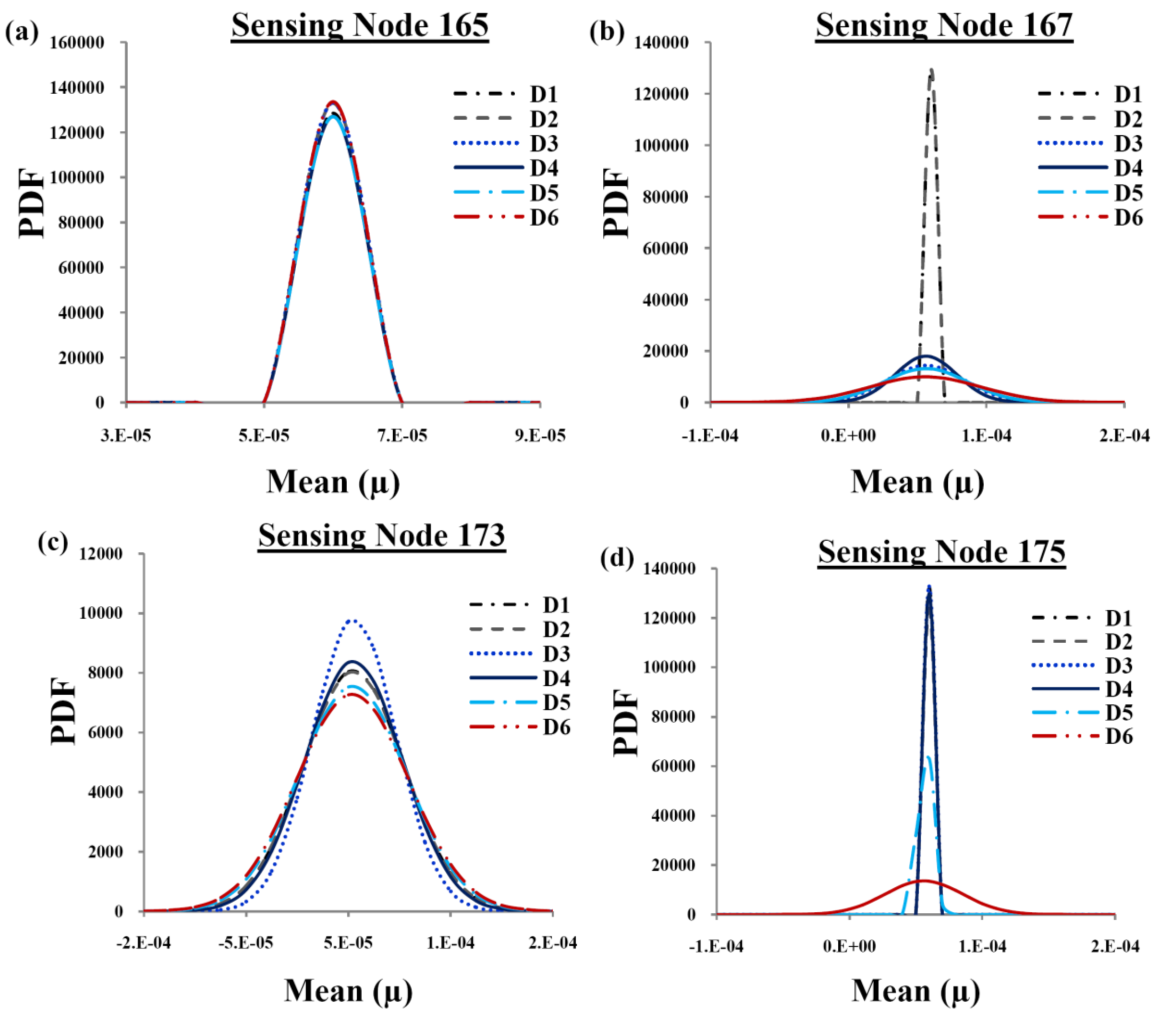


Figure 31. The PDF plots for the selected sensors (161 to 180) obtained from the girder fatigue analysis

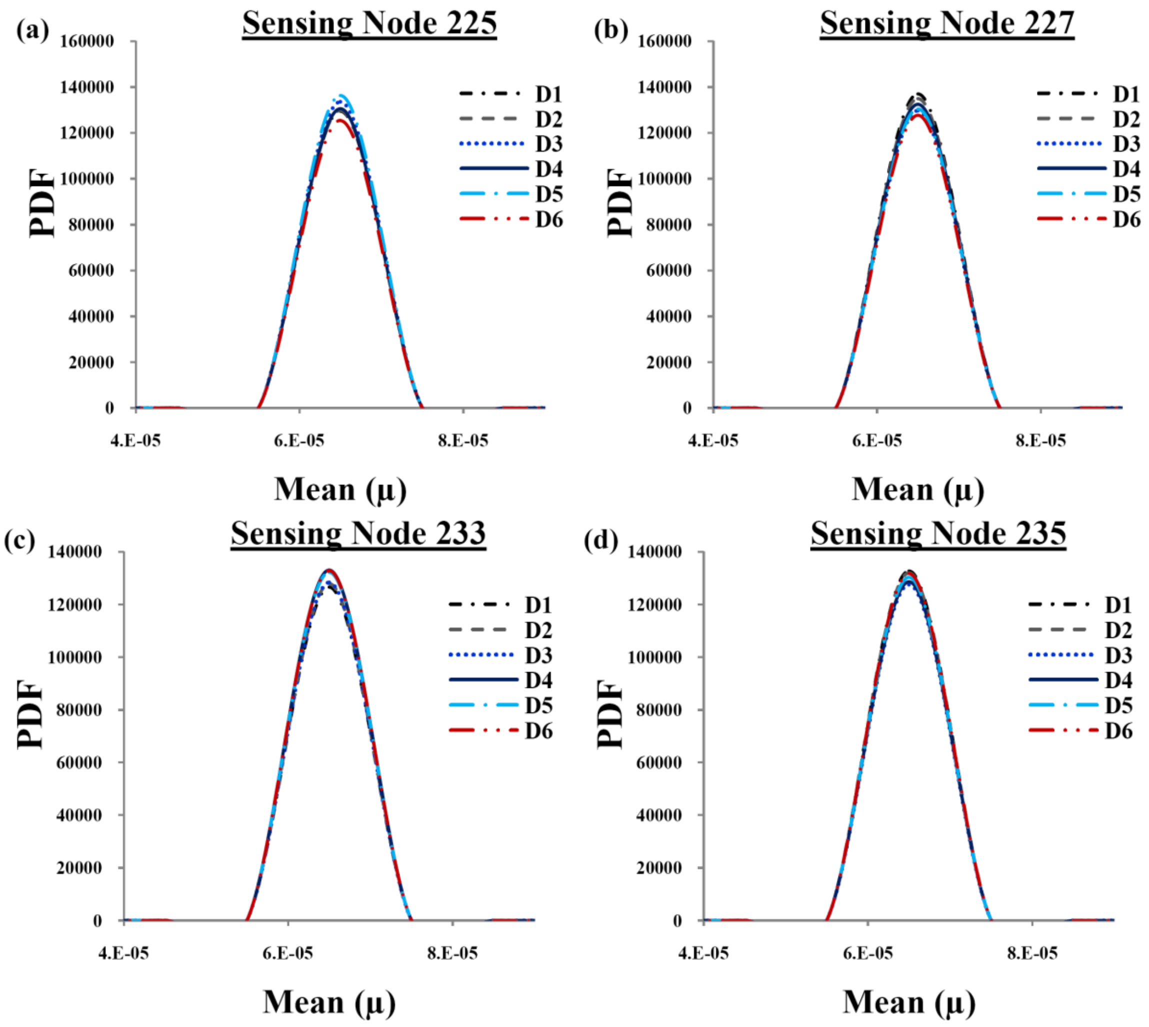


Figure 32. The PDF plots for the selected sensors (221 to 240) obtained from the girder fatigue analysis

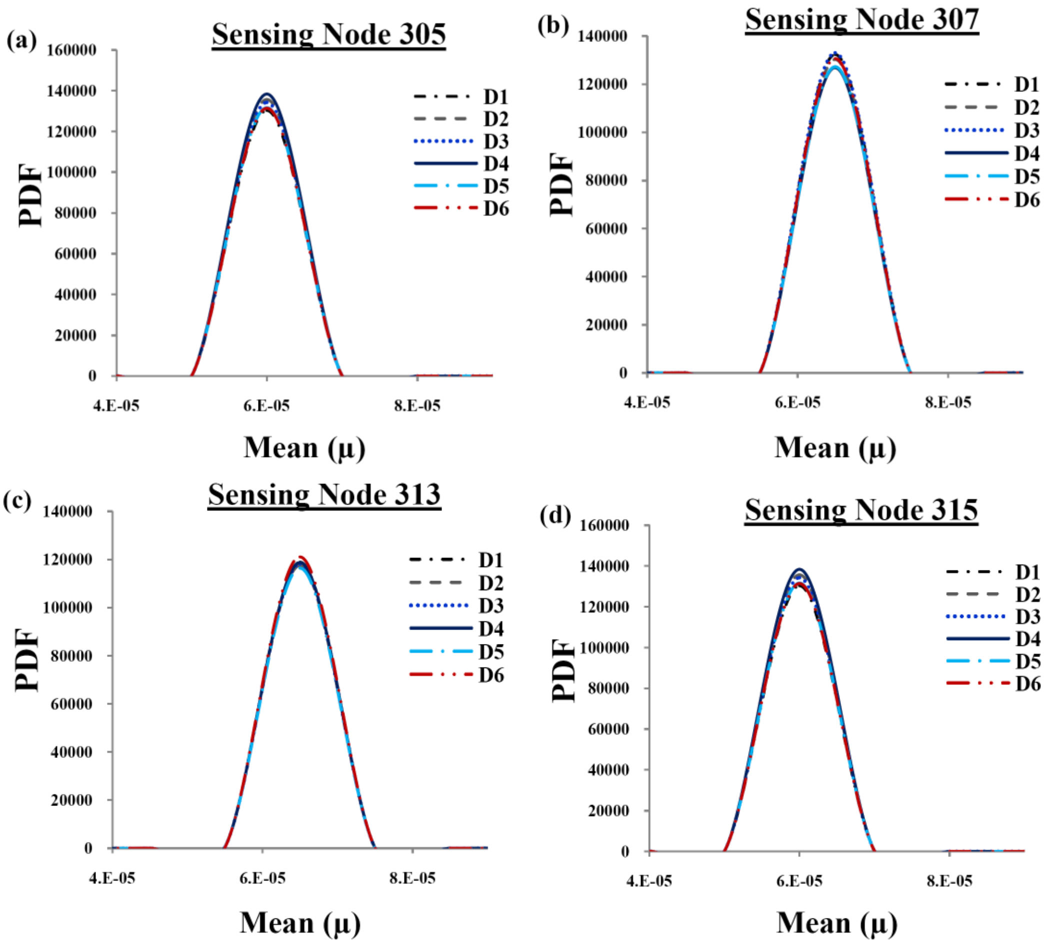


Figure 33. The PDF plots for the selected sensors (301 to 320) obtained from the girder fatigue analysis

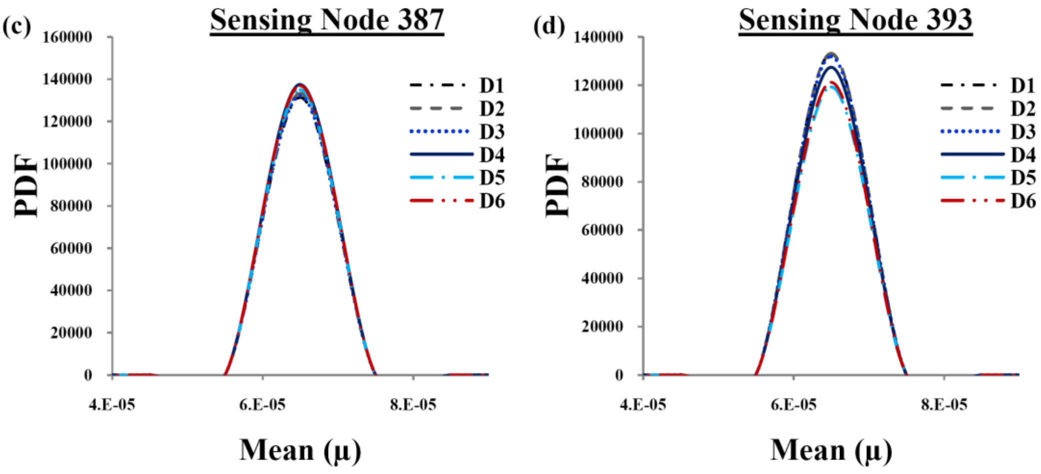
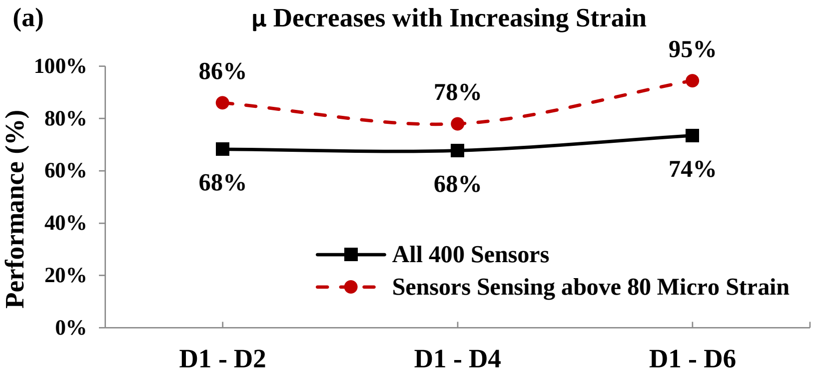
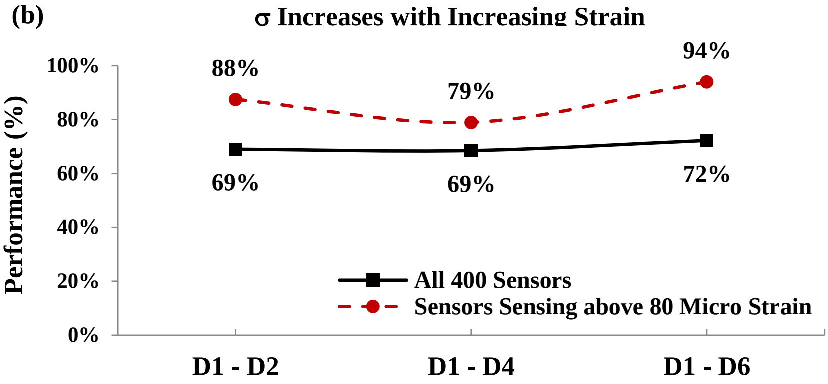
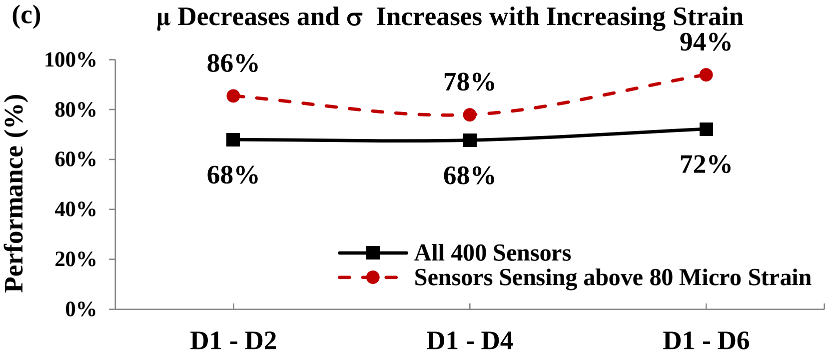


Figure 34. The PDF plots for the selected sensors (381 to 400) obtained from the girder fatigue analysis

Evidently, there are limitations for detecting damage with the individual sensor approach. Only sensing nodes at specified locations provide reasonable trends. Also, the exact relationship between the PDF parameters and decrement or increment of strains or damage progression is not clear. Thus, a statistical analysis is performed to find the changes of μ and σ with variations of strain patterns due to damage progression. The strain patterns changes with damage progression. It is observed from Figures 27 to 34 that the PDFs generally shift to left and their width increases as a consequence of damage progression. The damage progression usually results in higher strains throughout the system. However, this might not be the dominant case for the sensors that are not along the direction of crack propagation. Hence, the focus of this step is placed on calculating the percentage of the sensors of which μ decreases and σ increases with increases of strain and damage progression. Figure 35 presents the result of the statistical analysis for the girder with fatigue cracking. The comparison is carried out between the D1 and the two other typical damage states (e.g. D4 and D6) for brevity. In these figures, the vertical axes demonstrate the percentage of the sensors that their μ decreases and σ increases with increases of strain and damage progression. The black lines shown in these figures represent the results for all of the sensors and the red lines denote the results for the sensors that measure above a particular threshold. Figures 35 (a), (b) and (c) indicate that for about 70% of the sensors, μ decreases and σ increases with increasing strain. In general, the relationship between μ and strain is more reasonable than that of σ and strain. On the other hand, it is not always the case that a sensing node would experience higher strain due to damage progression. For instance, if the crack propagates to a direction opposite of the sensing location, the stress concentration will decrease at the node resulting in reduction of strains. This issue complicates the interpretation of the data. As seen in Figure 35d, the percentage of the sensors of which μ values decrease with damage progression is fairly low (maximum of 61%). The main reason for considering the thresholds is to deal with this issue. Referring to Figure 25, it is seen that second strain level for the memory gates is 82 με. Based on the FE simulation results, a fairly high percentage of the sensors (near half of the sensors) experienced strains below this threshold. In other words, Sensing nodes 200 to 400 mostly sensed strains lower than 80 με. This is while only one 30 με gate is considered to record the data for these sensors. Thus, a notable amount of data is missing for the analysis. By ignoring the sensors with strains lower than 82 με, remarkably better results are obtained. As shown in Figure 35 by red lines, there is a very good correlation between the decrease of μ, increases of σ, increases of strain, and damage progression. Specifically, it can be claimed that if sufficient levels are defined for the memory gates for low strains, the damage progression can be detected by assessing the decrements of μ and increments of σ with good accuracy.







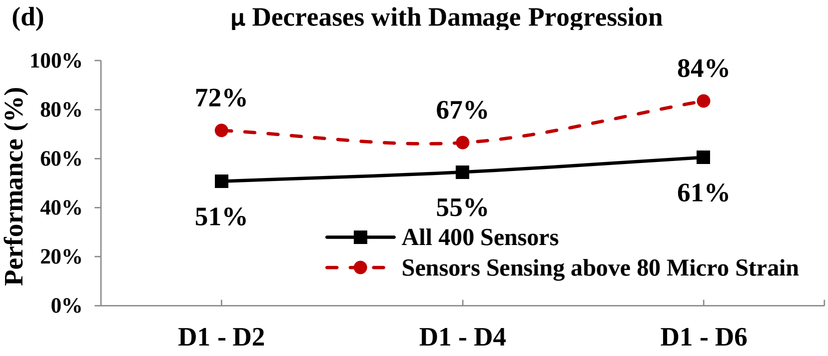


Figure 35. Changes of μ and σ with variations of strains due to damage progression for the girder with fatigue cracking

2.2.3 Failure Analysis of Pavement Systems

Pavement health monitoring plays a key role in pavement management systems. Early repair and maintenance scheduling increase the safe operation and in-service performance of pavement. This can be achieved through an accurate and consistent monitoring of pavement condition. In general, the existing approaches for pavement health monitoring can be divided into external evaluation technologies and in situ pavement sensors (Xue et al., 2014). The external evaluation methods have been commonly used for the evaluation of surface distresses. Typical examples in this context are using image analysis techniques to analyze the pavement distress (Mohajeri and Manning, 1991; Koutsopoulos and Downey, 1993), or stereo-imagery for measuring pavement deformation (Mills et al., 2001). Besides, there are numerous nondestructive evaluation (NDE) methods for the assessment of assess the behavior of pavements and other infrastructures (Goel and Das, 2008; Plati et al., 2014). The in-situ pavement sensing methods have been the focus of many studies for the last decades as alternatives to the traditional monitoring (Potter et al., 1969; Badr and Karlaftis, 2013; Badr and Karlaftis, 2012). There are several full-scale test studies to measure the in situ pavement responses under traffic load (Al-Qadi et al., 2004; Rollings and Pittman, 1992). However, this section presents a new system for the continuous long-term health monitoring of pavement structures based on the SWS data. A 3D FE analysis is performed to obtain the pavement responses under moving tire loading. The main goal is to detect the fatigue cracking due to excessive tensile strain at the bottom of the asphalt concrete. A quasi-static approach is used for the simulation of movement of the load at the desired speed. In order to analyze the response of sensors embedded within the asphalt layer, a series of tests are conducted on an asphalt concrete beam under a three point bending configuration. A new miniaturized spherical packaging system is designed and tested to protect the sensors embedded inside the specimen. The possibility of localizing the damage and quantifying its severity is investigated and discussed.

2.2.3.1 Numerical Study

A 3D FE model is developed to analyze the dynamic response of the pavement under a moving truck tire loading. ABAQUS is used for the modeling and post-processing of the results. A series of acquisition (sensing) nodes are considered at the bottom of the asphalt layer as the potential sensors. The studied pavement is composed of 3 layers: asphalt, base and subgrade layers. The FE model is shown in Figure 36.

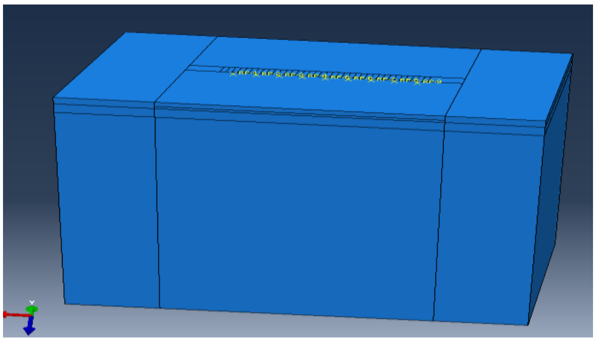


Figure 36. The 3D FE Model for the pavement structure

Geometry: The model has a dimension of 6200 mm (244") along the direction of traffic and 4170 mm (164") along the transverse direction (the width of one lane). A model of such size is used to minimize any edge effect errors, especially on the longitudinal tensile strain.

Material properties and thickness of layers: Table 7 shows E, ν, and thickness of layers of the pavement system.

Table 7. Material properties and layer thickness

Layer	E	ν	Thickness
Asphalt layer	2757.9 MPa (400 ksi)	0.35	152.4 mm (6")
Base	344.7 MPa (50 ksi)	0.35	127 mm (5")
Subgrade	34.55 MPa (5 ksi)	0.48	4826 mm (190")

Mesh: A schematic illustration of the FE mesh is illustrated in Figure 37. The model is idealized with linear hexahedral element of type (C3D8R). On the basis of a sensitivity analysis, the mesh dimensions are considered about 12.7 mm (0.5") in the loading area and 25.4 mm far from the contact zone. The total number of elements is 615076 elements of type C3D8R. Due to the high number of degrees of freedom of the model, a supercomputer is used to run the ABAQUS simulations for both the intact and damaged models. The available servers with high performance computing at Division of Engineering Computing Services (DECS) at MSU are used for this aim. Using a server with 384 GB RAM and 72 logical RAM, the simulation time for the intact configuration took around 42 hours.

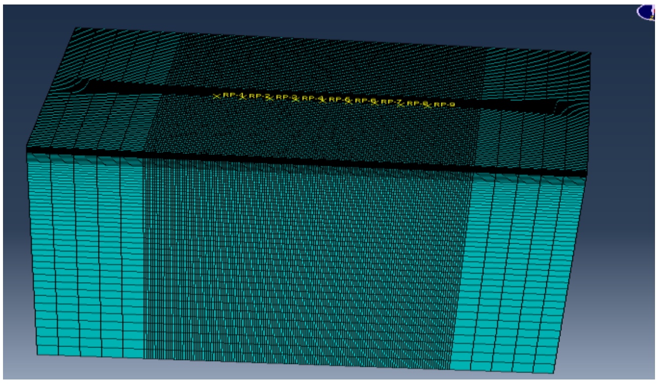


Figure 37. The FE mesh for the pavement model

Boundary conditions: The vertical or horizontal movements at the bottom of the layer are restrained. Also, there is no horizontal displacements perpendicular to the perimeters of the pavement while the rotation is free. Furthermore, to simulate a new pavement condition, the interface between different layers had no relative movement.

Loading: To simulate the movement of the load at the desired speed, a quasi-static analysis is adopted (Al-Qadi and Wang, 2010). The location of the load and its amplitude are gradually shifted over the loading path at each step until a single wheel pass is completed (Figure 38). Based on the concept of a continuously moving load, an element is increasingly loaded when the load approaches and then unloaded as the load leaves it (Al-Qadi and Wang, 2010; Yoo and Al-Qadi, 2007). The traffic speed and the length of the element can be used to calculate the step time. In this study, it is supposed that the vehicle runs at a constant speed of 6 mph (about 10 km/h). The tire pressure is taken 0.69 MPa (100 psi). The load is moved a 76.2-mm (3") in each increment over 6 elements. The total number of increments (locations of the load) required to achieve one full passage of the tire over the entire model is 32. More details about the procedure followed in this study to simulate the moving load can be found in Al-Qadi and Wang (2010).

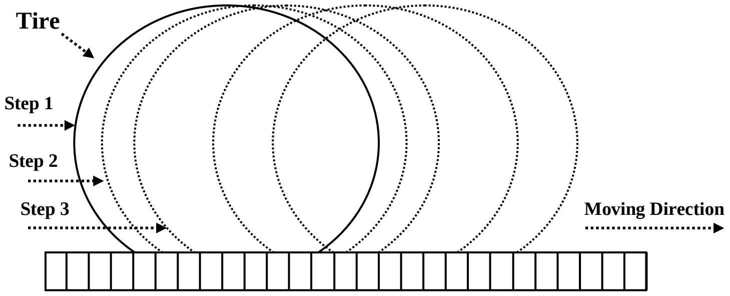


Figure 38. A schematic representation of the moving load modeling

Location of sensors and damage zone: Figure 39 shows the location of the sensing nodes. As it is seen, the sensing nodes are located at a distance of 50.8 mm (2") from the bottom of the asphalt layer. Figure 40 illustrates the layout of the sensing nodes inside the pavement. The distance between two consecutive sensing nodes is 304.8 mm (12"). Also, distance of Sensing nodes 1 and 9 from vertical boundaries is equal to 1880 mm (74"). In this analysis, 55 sensing nodes are used to measure the longitudinal and transversal strains. Based on the FE simulations, the strain values for the sensing nodes far from the loading path are very low. Thus, it is not feasible for the sensors to detect the change. Consequently, out of the available sensing nodes, only those along the loading path which experienced strains higher than 20 με are kept for the analysis. These are Sensing nodes 1 to 9 located under the loading path. Since the tire loading imprint area is started from the location of Sensing node 1 and terminated by Sensing node 9, only the results for Sensing nodes 3 and 7, located at an adequate distance from loading points, are included in the analysis. Later, a new FE model is developed with a damage zone at the bottom of the asphalt layer. The length of the crack zone is 0.5" from the bottom of the asphalt layer (Figure 39). The location of the sensing nodes is the same as the intact model. Sensing node 5 is above the damaged area.

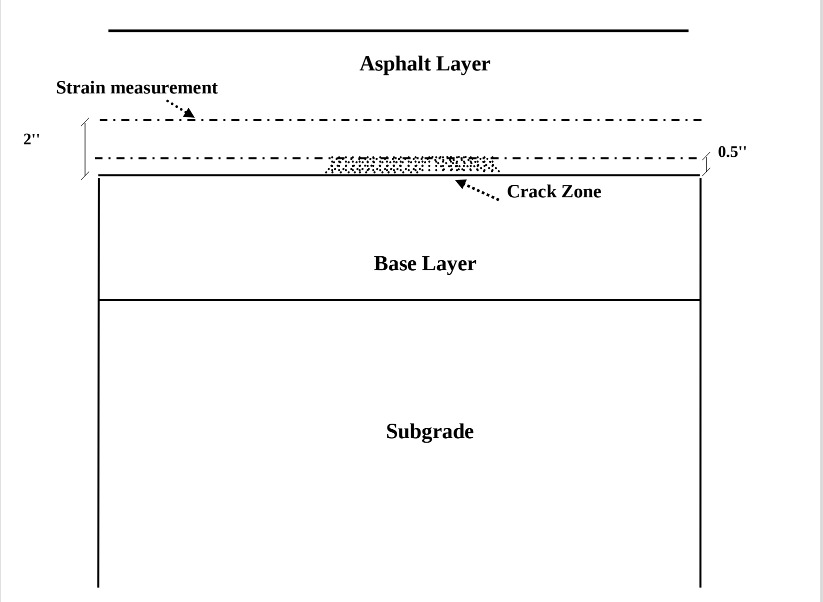


Figure 39. Crack zone and measurement location

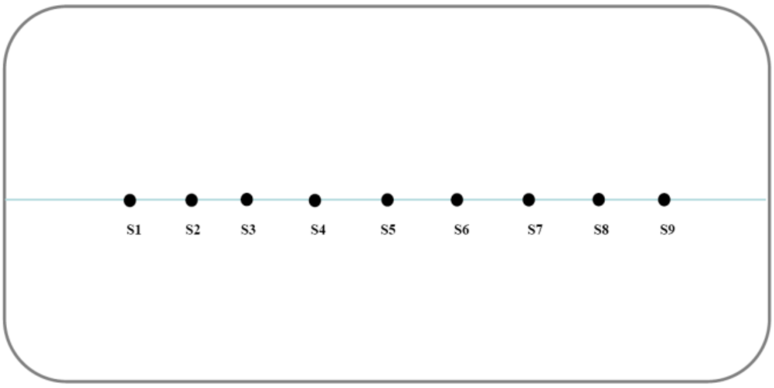
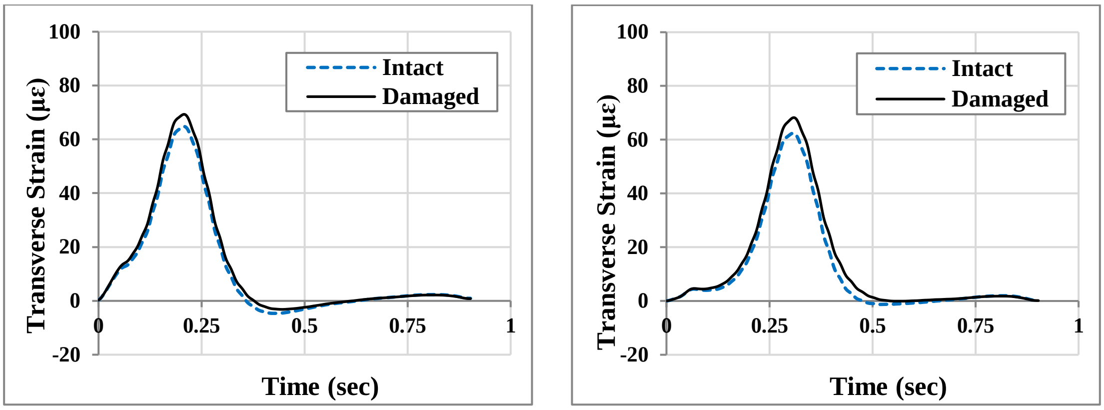


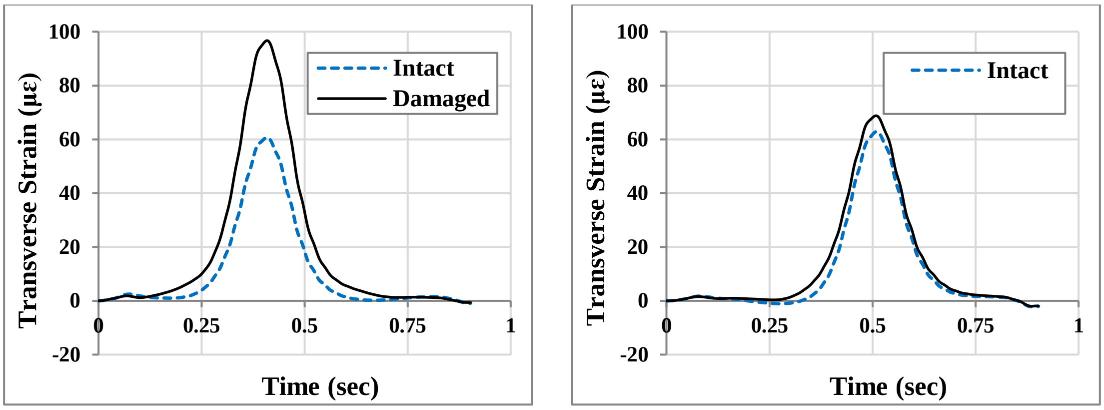
Figure 40. Layout of data acquisition (sensing) nodes

FE simulation results: Figure 41 show the transverse strains for the intact and damaged pavements. As it is seen, the damaged pavement experiences higher strains in all sensing nodes. This is more evident for Sensing node 5 which is located above the damage zone.



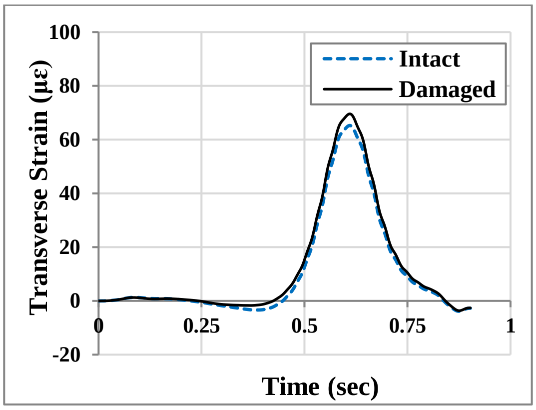
(a) Sensing node 3

(b) Sensing node 4



(c) Sensing node 5

(d) Sensing node 6



(e) Sensing node 7

Figure 41. Comparison of the transverse tensile strains for the intact and damaged pavement

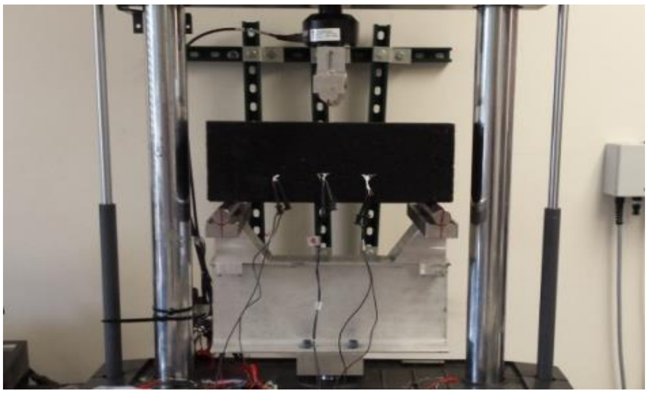
2.2.3.2 Experimental Study

The performance of the sensors is tested through an experimental study on an asphalt concrete specimen. The sample is tested under a three point bending configuration (Figure 42). The test fixture is similar to that of Single Edge Notched Beam SE(B) Test which is one of the most documented methods to determine fracture properties (ASTM E1820-09, 2009; Marasteanu et al., 2012). However, the size of the sensor packaging (about 1") limits the use of recommend by ASTM E1820-09 (2009) for the beam geometry in the SE(B) method. In this study, the fixture is developed for an asphalt concrete sample with a span length of 381 mm (15"), thickness of 165.1 mm (6.5") and a width of 152.4 mm (6") (Figure 42). The rollers have a diameter of 30 mm (1.2") and are setup to be free in rotation. The slab is built using the HMA, 4E1 mixture type provided by Michigan Department of Transportation (MDOT). The sample is compacted using PReSBOX Asphalt Shear-box Compactor. The temperature is around 150 °C (302 °F). The weight of the hot mix asphalt (HMA) is 25 kg and the length of the slab is equal to 450 mm (17.75"). The center load point is designed to provide a uniform loading. The three point bend tests are carried out using an MTS machine. All tests are conducted under constant axial displacement rate control. The tests are done at 2 and 5 Hz loading frequency for 0.1, 0.15 and 0.2 mm amplitudes. Before starting the test, a preload equal to 0.5 kN is applied to the sample to ensure it is seated on the fixture. Thereafter, the cyclic displacements are applied.

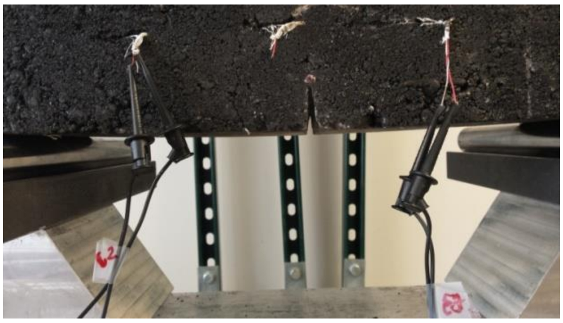
The main goal is to evaluate the response of the embedded piezoelectric transducers due to damage progression. Damage is introduced by making a notch at the bottom of the asphalt layer using a circular saw. The damage states are defined by increasing the notch size (a) as follows:

- Intact: Intact plate (a= 0 mm)
- Damage 1: a = 22.2 mm (7/8")
- Damage 2: a = 31.75 mm (1 1/4")

For this stage, different damage states are defined to evaluate the sensitivity of the sensors to the damage severity. After introducing the second damage phase, the displacement is increased to 0.3 mm to evaluate the behavior of the sample for higher amplitudes. After a number of cyclic loadings, a crack propagation phenomenon is observed. The crack length is 12.7 mm (0.5"). This new damage phase is considered as Damage 3. Accordingly, the total of length of crack for Damage 3 is 44.45 mm (1 3/4"). Figure 43 illustrates the captured crack growth in the test.



(a) Three point bending configuration



(b) A typical notched specimen representing Damage 2

Figure 42. Test setup and sensor locations

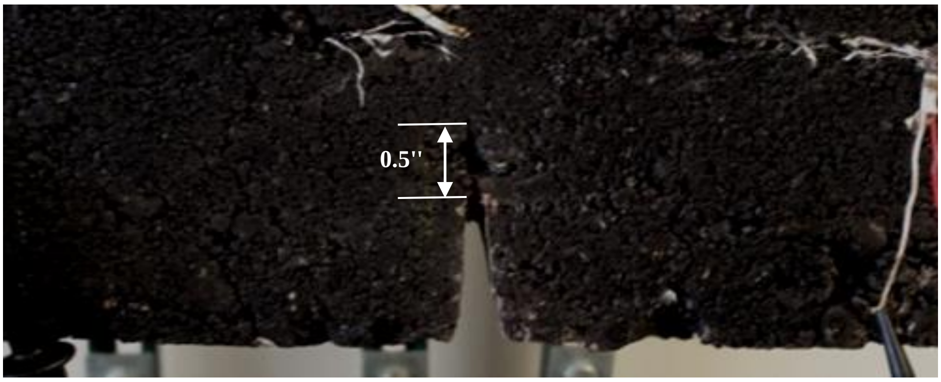


Figure 43. The crack propagation phase during the test (Damage state 3)

Design and implementation of a small scale packaging system: Current installation procedures demand considerable care during construction. This is to insure that the commonly-used H-gages are properly bonded to the pavement surface layer, and are properly aligned in both horizontal and vertical directions. For the purpose of this study, the packaging system is miniaturized so that it can be tossed in the pavement material during construction or can be used within a mesh network distributed over the base layer (Figure 44). The size of the designed spherical packaging system is of the same order of a coarse aggregate particle. All the electronics already have a small size and the antenna can be miniaturized to fit the desired size. The spherical packaging systems are designed using SolidWorks software and built by a 3D printer. The top part has an opening to put epoxy into the mould. The two parts combine together as a sphere with six anchor legs in three directions.

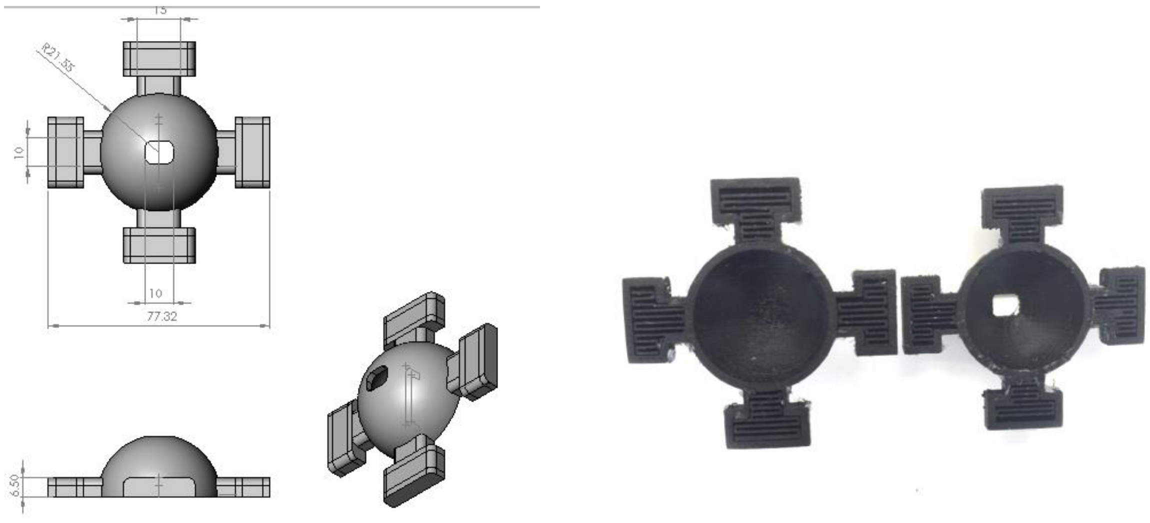


Figure 44. The designed sensing system in a spherical casing

The RFID, antenna and electronics are fragile and sensitive parts. They also have very low melting temperature. As a result, they cannot survive alone in asphalt pavement because asphalt pavement is under high pressure and temperature during constructions. Epoxy materials have relatively high melting temperature and excellent flexibility. Thus, epoxy materials can be used as the protectors for the sensor electronics. On the other hand, when the epoxy is strained, an induced axial loading is applied to the piezo. Therefore, choosing an appropriate epoxy resin to manufacture the spherical casing is an important issue. There should be stiffness compatibility between the epoxy and the host material (HMA). Stiffness compatibility can be achieved using epoxy with a lower stiffness than the host material. Also, nonmetallic materials should be used so that it allows for wireless communication. Among different epoxies tested (e.g. Araldite® GY-6010 and EPIC S7514), Conathane® TU-981 epoxy is found to perform better. The robustness of the Conathane® TU-981 epoxy to withstand harsh environmental conditions has been already verified (Lajnef et al., 2013). Thus, this epoxy is used for encasing the proposed spherical packaging system. Conathane® TU-981 is a two-component liquid casting system that produces a 65-Shore D elastomer of exceptional toughness and extraordinary processing flexibility.

The elastic modulus of the Conathane® TU-981 epoxy is obtained through an experimental study. The MTS machine is to this aim. The dimension of the Epoxy TU-981 sample is 40 × 35 × 16 mm (length × width × height). The tests are done using cyclic loading on two directions of epoxy specimens. The measured elastic modulus of Epoxy TU-981 is 258.8 MPa. Besides, the survivability of the epoxy in high pressure and temperature during asphalt pavement compaction is verified using the Asphalt Shear-box Compactor. For brevity, the results for this phase are not presented.

Figure 45 presents the manufacturing process of the packaging system with embedded piezoelectric transducers. The epoxy resin is cured for 24 hours at room temperature. A series of 3 PZT ceramic transducers with spherical packaging are embedded within the asphalt layer in slab compactor. The PZT ceramic transducers are manufactured by Steiner & Martins Inc., USA. The PZT properties are shown in Table 8. The spheres are located at a distance of approximately 2" from the bottom of the layer. The first PZT is placed at mid-span and the remaining ones are located at a distance of 76.2 mm (3") from the middle PZT. The layout of the PZTs is shown in Figure 46. After compaction, a similar module to the plate study is utilized to measure the PZT outputted voltage.

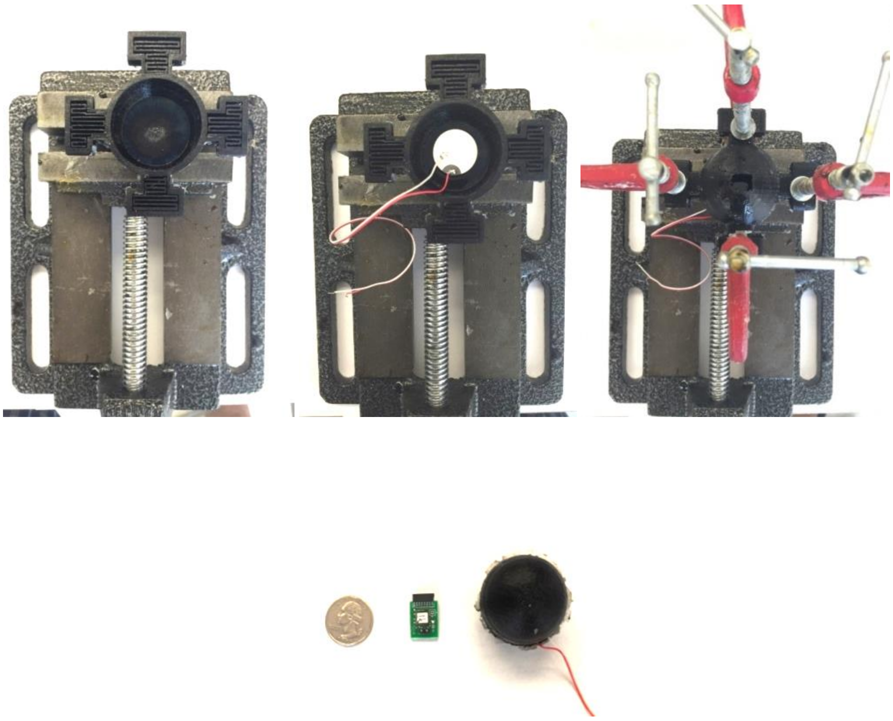


Figure 45. Manufacturing process of spherical packaging

Table 8. The properties of PZTs

Parameter		Value
Diameter		20 mm
Thickness		3 mm
Piezo material		SM111 (PZT-4)
Resonant frequency		690 KHz ± 3%
Resonant impedance (Z_m)		≤3.6 Ω
Static capacitance (C_s)		1265pF±10%
Piezoelectric constant	d_{33}	320×10-12m/v
	d_{31}	-140×10-12m/v
Density		7.9g/cm3
Electromechanical coupling coefficient	K_p	0.58
	K_t	0.45
	K_{31}	0.34

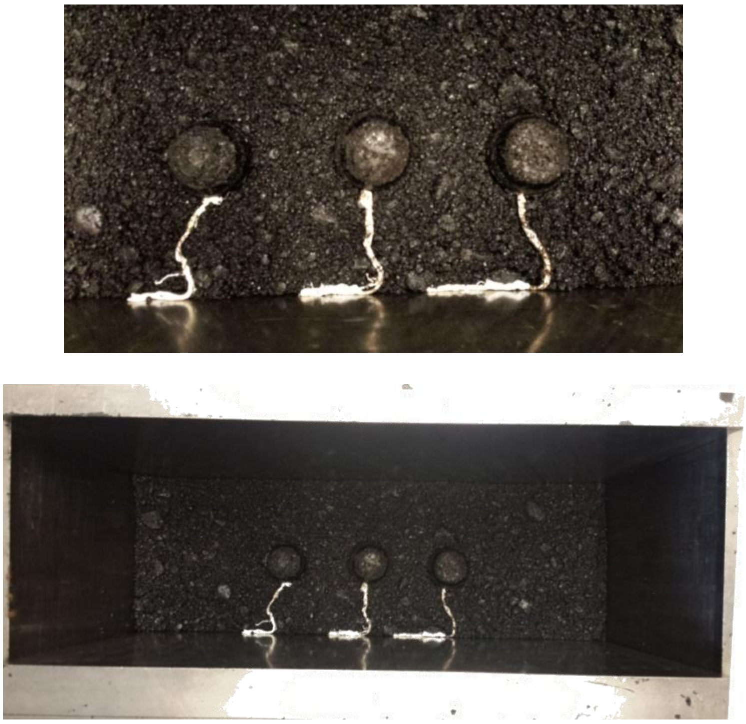
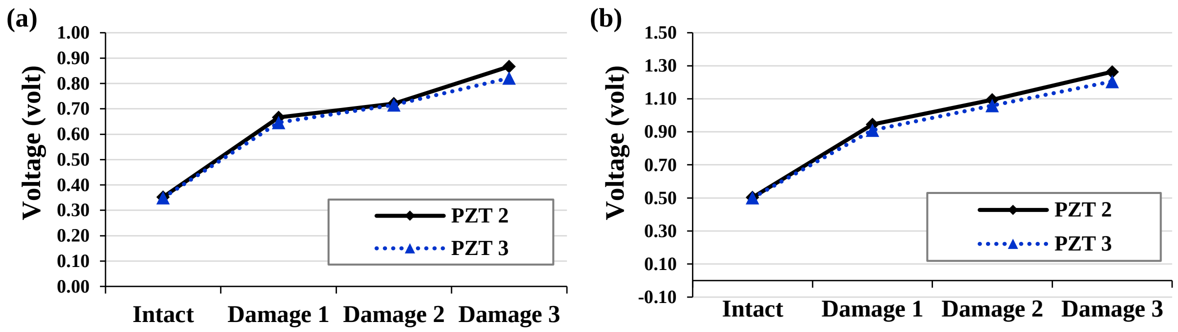
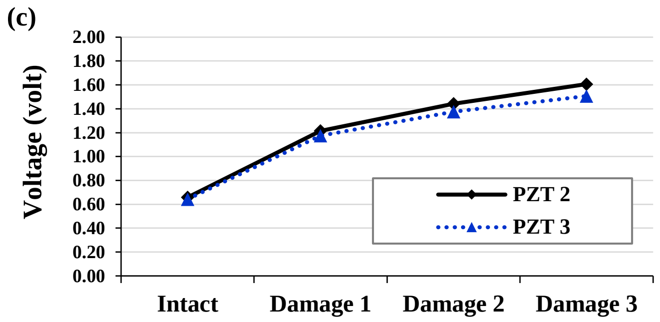


Figure 46. Layout of the spherical packaging in the compactor

Comparison of the maximum voltage delivered by PZTs for different damage states: The preliminary tests showed that PZT 1 located at the middle of the slab is not delivering voltage. A probable reason is failure of the PZT or the connection wires during the compaction process. Accordingly, the results for the other PZTs are recorded and presented in Figures 47 and 48. As can be observed in these figures, the PZTs are experiencing higher strains and therefore outputting higher voltage by increasing the notch size.

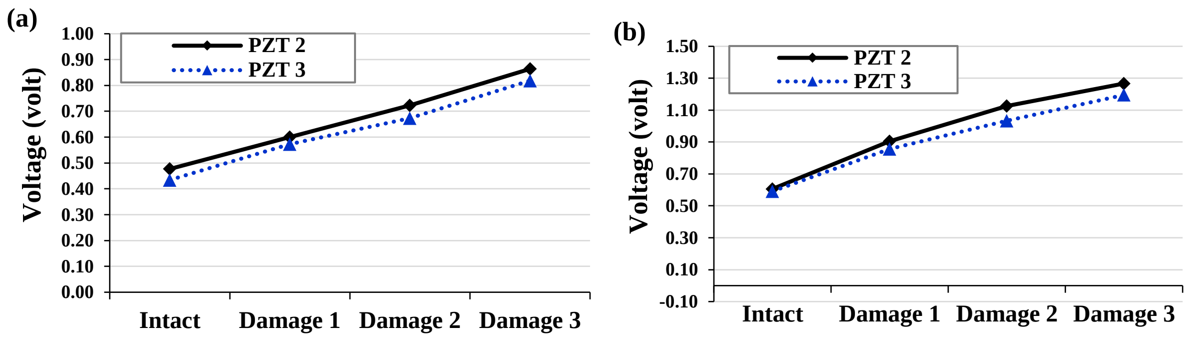


(a) Displacement: 0.1 mm (b) Displacement: 0.15 mm



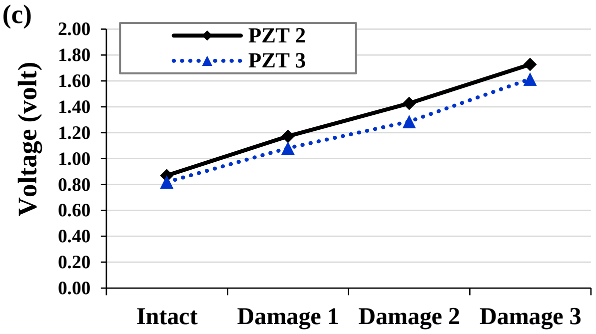
(c) Displacement: 0.2 mm

Figure 47. A comparison of maximum delivered voltage by PZT discs for 2 Hz loading frequency



(a) Displacement: 0.1 mm

(b) Displacement: 0.15 mm



(c) Displacement: 0.2 mm

Figure 48. A comparison of maximum delivered voltage by PZT discs for 5 Hz loading frequency

2.2.3.3 Damage Detection Based on the FE Results

As discussed before, a series of sensing nodes is considered near the bottom of the asphalt layer to detect the damage progression. Fatigue cracking is one of these distresses and it is caused by excessive tensile strain at the bottom of the asphalt concrete (Hafiang, 2001; Ghuzlan and Carpenter, 2006). Considering the importance of tensile strain in the prediction of fatigue cracking, this type of strain is measured for the analysis. Referring to Section 2.2.3.1, the results for sensors with a reasonable distance from loading points are included, i.e., Sensing nodes 3 and 7. 10 strain levels are defined to cover the lower and upper limits of the strain values extracted from the FE simulations. The minimum level of strains to be captured by piezoelectric transducers is about 20.00 με. On the other hand, the maximum of the strain value extracted from the FE simulations is about 97 με. Considering a reasonable value of 10 strain levels for the gates, the difference between the strain levels is 8.56 με. Table 9 presents the preselected strain levels for the pavement layer. The μ and σ values are used to plot the PDFs corresponding to each sensor (Figure 49). As seen in Figures 49 (a)-(e), μ decreases and σ increases by transitioning from intact to damaged mode. Accordingly, the PDFs shift to left, their width increases and their height decreases due to damage progression. It is evident that for Sensing node 5 which located above the crack, there is a distinct change in the PDF shape. It is worth mentioning that even for sensors far from the damage, the PDFs still shift to left and expand due to damage progression.

Table 9. The preselected strain levels considered for the pavement FE analysis

Gate number	Strain Level (με)
1	20.00
2	28.56
3	37.11
4	45.67
5	54.22
6	62.78
7	71.33
8	79.89
9	88.44
10	97.00

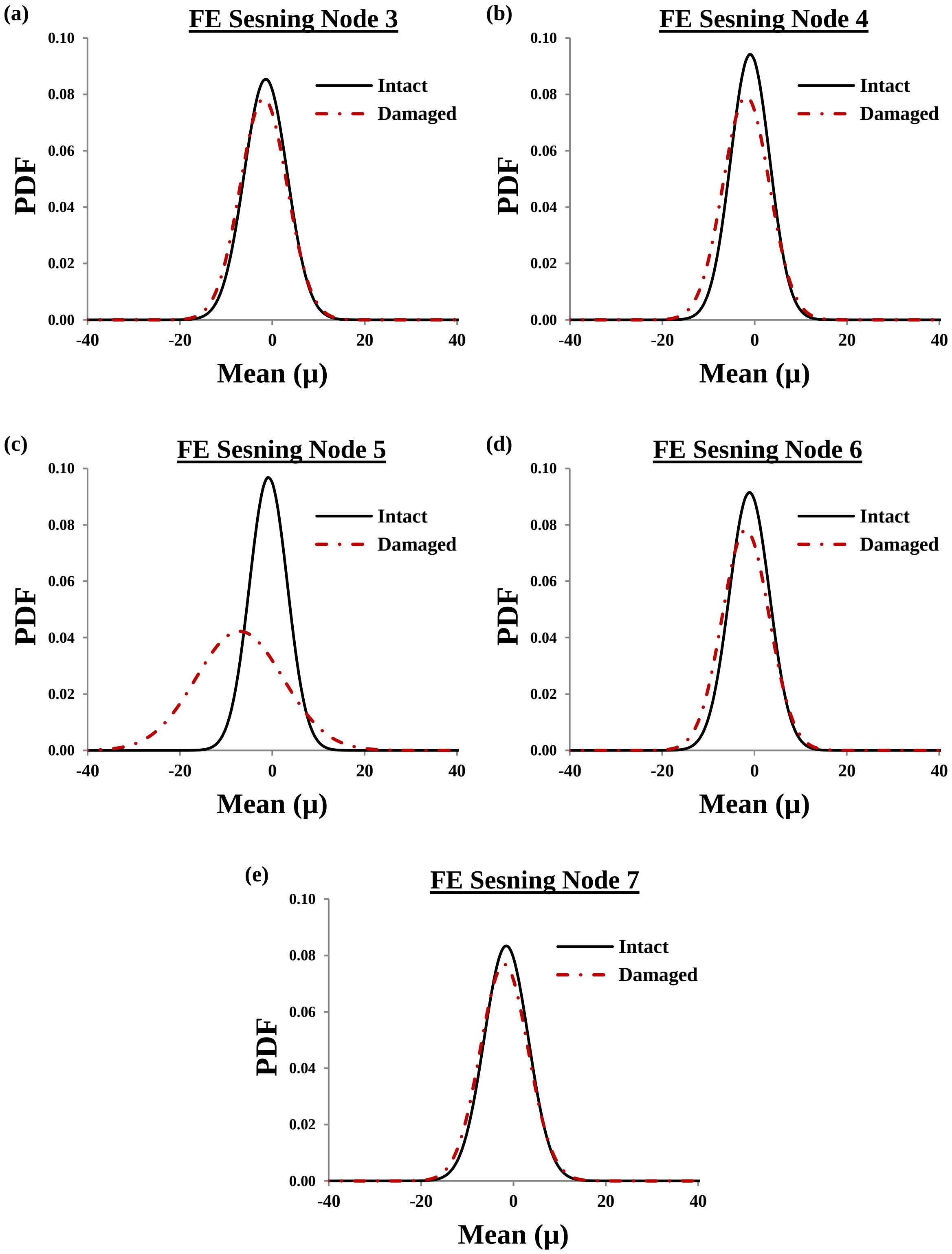


Figure 49. Changes of PDFs due to damage based on the pavement FE analysis

2.2.3.4 Damage Detection Based on the Experimental Results

The damage detection results are provided for amplitude of 0.2 mm and loading frequencies of 2 and 5 Hz. In this study, 7 voltage levels are taken for the gates. The minimum and maximum voltage thresholds are set to 0.2 and 2 volts, respectively. Accordingly, the difference between the strain levels is 0.3 volt. Table 10 presents the preselected voltage levels for the pavement layer. As discussed before, the voltage supplied by piezoelectric transducers is proportional to experienced strain (Sirohi and Chopra, 2000). However, the recorded voltage histories of PZTs 2 and 3 are inputted into a script in MATLAB to find the μ and σ of the strain distribution in the intact and damaged models. The μ and σ values are then used to plot the PDFs corresponding to each sensor (Figures 50 and 51).

For brevity, the cumulative time versus gate numbers and the fitted CDF curves are not presented. Also, since there are merely 2 sensors are involved, performing the group effect analysis is not done. As can be seen in Figures 50 and 51, the values of μ and σ, respectively, decrease and increase due to damage progression. This is in accordance with what previously observed for the FE model. Interestingly, the PDFs changes notably by transiting from Intact to Damage 1 compared to other stages. This is because the first introduced notch is deeper than those considered for Damage states 2 and 3. Additionally, the crack propagation process is accurately detected by the sensors.

Table 10. The preselected strain levels considered for the pavement experimental study

Gate number	Voltage Level (volts)
1	0.20
2	0.50
3	0.80
4	1.10
5	1.40
6	1.70
7	2.00

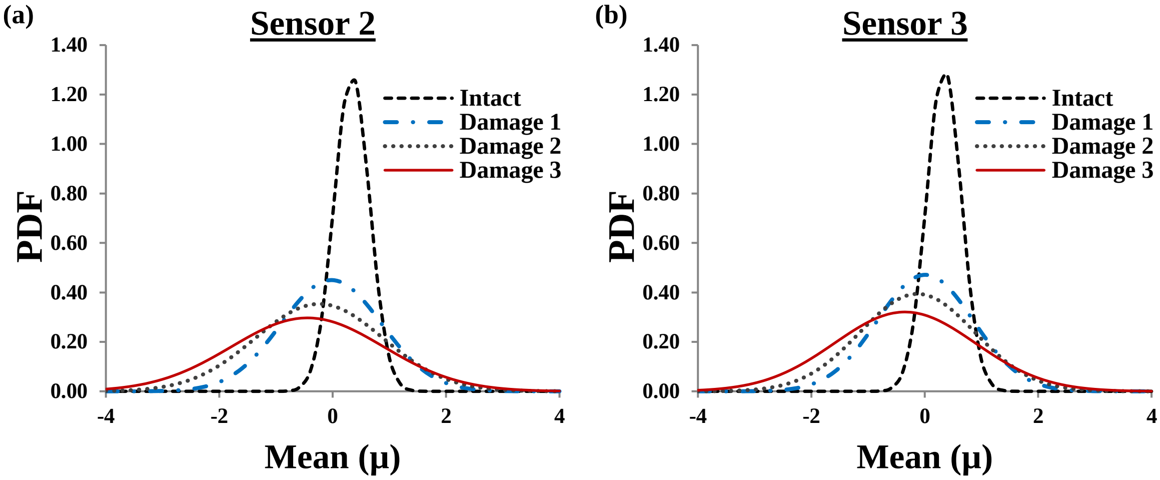


Figure 50. Changes of PDFs due to damage progression in the asphalt specimen for 2 Hz loading frequency

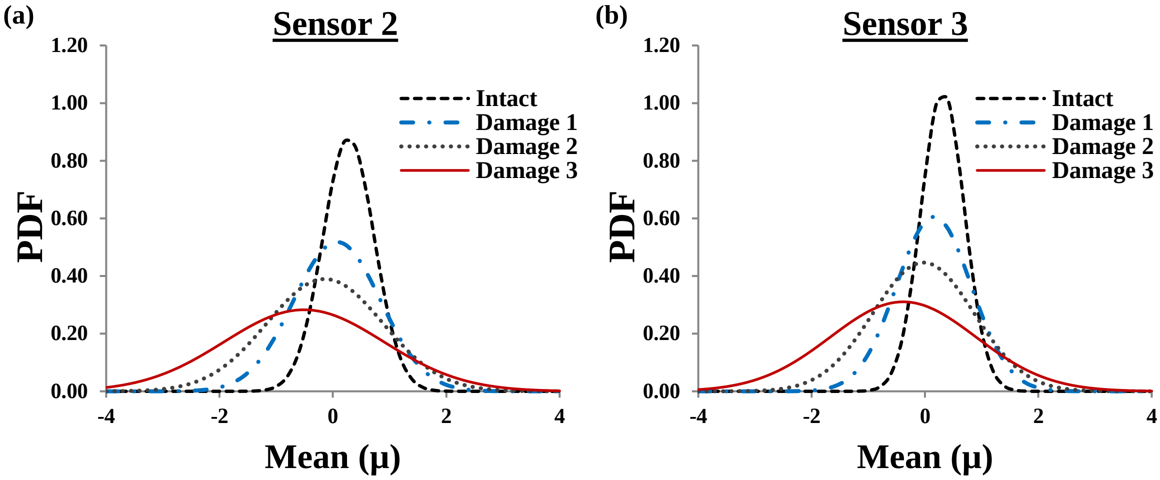


Figure 51. Changes of PDFs due to damage progression in the asphalt specimen for 5 Hz loading frequency

2.2.3.5 Discussion

A major problem with utilization of the existing wireless sensors is providing a reliable power source for them. To cope with this concern, a novel approach has been developed in this study for the continuous battery-less monitoring of structures. The results for the numerical and experimental study clearly indicate that the outputs of the developed Self-powered sensor are damage indicators. The main observation from both of the FE and laboratory studies is that the PDFs obtained from the cumulative time histograms change reasonably with damage progression. The expansion of PDFs and their shift to left side is shown as a sign of damage occurrence. In general, changes of σ seem to be a better sign of damage progression. On the other hand, the experimental study proved that the designed miniaturized sensor packaging withstands temperature and pressure during compaction. Such small size system can be easily used and integrated in large-scale sensor networks by State highway agencies. However, in addition to its damage detection capability, the proposed approach gives an insight into the damage localization and quantification. To clarify this issue, the percentage of variation of damage indicator parameters for the intact and damaged states in the FE simulations is presented in Table 11 and Figure 52. As it is seen, the highest variation belongs to Sensor 5 which is located above the crack zone. The variation has a proportionally increasing trend as the load approaches the damage zone (either from Sensors 3 and 4, or from Sensors 7 and 6 up to Sensor 5). This is valid for both of the μ and σ trends.

Table 11. Variation of damage indicator parameters of μ and σ based on the pavement FE analysis

	1st Feature: μ			2nd Feature: σ		
	Intact	Damaged	Variation Percentage	Intact	Damaged	Variation Percentage
Sensor 3	-1.43	-1.87	-31%	4.67	5.09	9%
Sensor 4	-0.95	-1.82	-92%	4.23	5.04	19%
Sensor 5*	-0.84	-7.10	-750%	4.12	9.44	129%
Sensor 6	-1.08	-1.89	-75%	4.36	5.10	17%
Sensor 7	-1.54	-1.98	-29%	4.78	5.19	9%

*Sensor 5 is located above the crack zone.

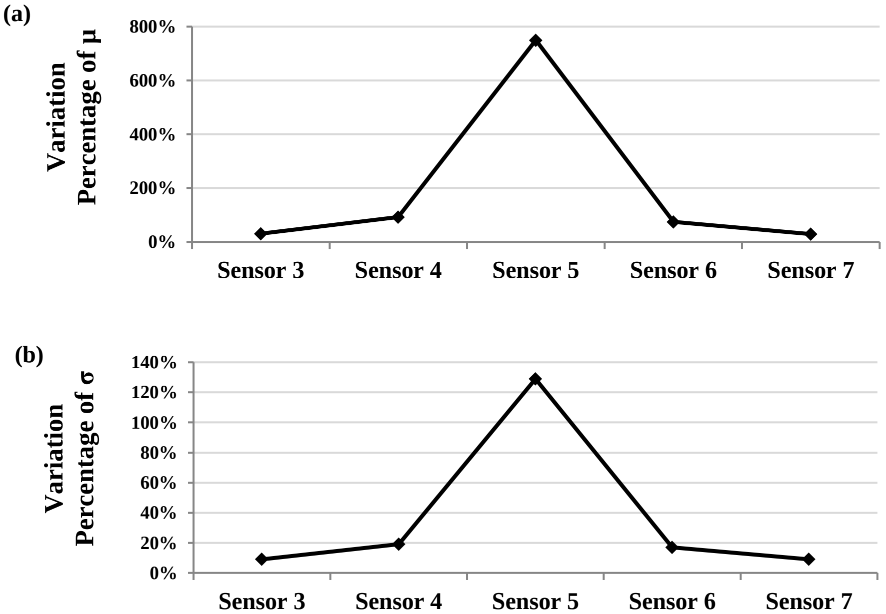


Figure 52. Percentage of variation of damage indicator parameters for the intact and damaged states based on the pavement FE analysis

2.3 Conclusions

In order to evaluate the functionality of the proposed damage detection approach, specific infrastructure problems are studied. The FE simulation results for the pavement system, steel plate and steel bridge girders indicate that damage can be identified using point strain sensors. The same results are observed from the laboratory testing. In this case, the sensors could detect different damage states including crack propagation phase. It is found that damage progression can be considered as a function of PDF parameters (μ and σ) obtained from the strain distribution. μ and σ are derived by curve fitting of the sensor output distribution collected from the entire memory cells of the sensor. The important observation from the individual sensor analysis is that the PDFs shift to left (μ decreases) and their width increases (σ increases) due to the damage progression. Based on the results, variations of σ can be regarded as a better indicator of damage.

CHAPTER III

DATA FUSION IN A NETWORK OF SELF-POWERED STRAIN SENSORS

3.1 Data Fusion Systems for Damage Detection

Sensor fusion is a process of integration and extraction of desired information from two or more sensors. In other words, it is a process of combining multiple sensors to provide more useful information than the sum of individual sensors. Fused sensor data from various sensors offers several advantages compared to the data from a single sensor (Hall and Llinas, 2001). In general, data fusion structure can be divided into three types (Hall and Llinas, 2001; Gang et al., 2010): signal-level, feature-level, and decision-level. In the signal-level fusion, all sensor raw data from a measured object are combined directly and a feature vector is then extracted from the fused data. In the feature-level fusion, features are extracted from each sensor according to the type of raw data. Then, this sensor information is combined at the phase of the feature level. All feature vectors are combined in turn to a bigger single feature vector, which is then used in a special classification model for decision making. In the decision-level fusion structure, the processes of feature extraction and pattern recognition are employed for single-source data obtained from each sensor. Then the generated decision vectors are fused using decision-level fusion techniques. Based on another common framework, data fusion architectures can be independent, centralized, or decentralized (Staszewski et al., 2004; Su et al., 2009). Independent fusion architecture is the simplest from in which signal features are extracted and recognition is carried out independently by individual sensors. In the centralized fusion architecture, a generic set of features with provided by individual sensors are extracted in parallel for subsequent recognition. Finally, decentralized fusion architecture executes feature extraction and selection for each sensor independently. The features can be in common or irrelevant from sensor to sensor. All the extracted features are then fused for recognition (Su et al., 2009). In all of these schemes, data mining techniques play a key role in providing more reliable and accurate information. In this research, statistical and AI approaches are used to facilitate the process of the damage detection data fusion systems.

3.2 Statistical Approach

A major challenge in application of wireless sensors results from the fact that damage in structures is an intrinsically local phenomenon. As discussed in Chapter II, it is possible to detect the damage progression with individual (point) sensors. The point sensors provide discrete and localized measurements. Although they deliver valuable quantitative information, measurements at a single location might not be sufficient for accurate damage detection (Burton et al., 2014). Rationally, the closer the sensors to the damaged zone are, the higher the damage detection accuracy is. For the cases where the potential sensors are at a far distance from the imposed damage, the sensor may not have a sense of damage or the results may not have a reasonable trend. In order to tackle this issue, a potentially more effective data fusion strategy is defined to improve the damage detection performance through spatial measurements over an area of the considered structures. On this basis, it is decided to use the information provided by a group of sensors, termed as "group effect". In this case, even if one sensor does not sense the damage, the group effect will help detect the damage. Figure 53 shows the structure of the proposed statistical-based data fusion system for detecting damage progression. The main goal is to find the relationship between the PDF parameters of a group of sensors and damage progression as follows:

$$\textit{Damage progression} = \textit{function of}\ (\mu_{group}, \sigma_{group}) \tag{14}$$

Accordingly, the average, standard deviation (STD), range, minimum, maximum, skewness, and kurtosis of the PDF parameters (μ and σ) for the group of sensors are obtained and normalized for both of the experimental and numerical studies. The equations for skewness and Kurtosis are defined as:

$$Skewness = \frac{n}{(n-1)(n-2)}\sum\left(\frac{x_i-\bar{x}}{s}\right)^3 \tag{15}$$

$$Kurtosis = \frac{n(n+1)\sum(x_i-\bar{x})^4}{(n-1)(n-2)(n-3)s^4} - \frac{3(n-1)^2}{(n-2)(n-3)} \tag{16}$$

where $\bar{x}$ is the mean and s is the standard deviation of the distribution. n is the number of samples.

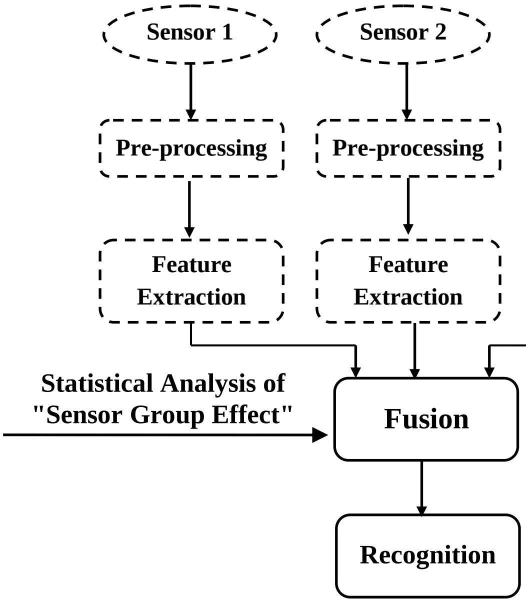


Figure 53. The statistical data fusion framework for detecting damage progression

3.2.1 Crack Growth Detection in Steel Plate

In order to analyze the sensor group effect, average, STD, range, minimum, maximum, skewness, and kurtosis of the PDF parameters (μ and σ) are obtained and normalized. Based on preliminary analyses, the average, range, minimum, maximum, skewness, and kurtosis of the PDF parameters did not have a sound relationship with the damage progression. Thus, only the variation of the STD of μ and σ of group of sensors is taken into account. Figure 54 presents the results for different configurations. The red circles represent the sensing nodes that were included in the analysis. As it is seen, the STD of μ and σ increases with damage progression for all layouts including the sensors located along the crack (Figures 54 (a)-(f), (k)-(n), (q)-(t)). This is an expected case because Sensors 4-7 in the group experience higher variations due to damage and therefore lead the behavior no matter if there exists a dormant sensor or not. The results pertaining to the STD of μ and σ are similar to each other. The best agreement between the FE and experimental results belongs to the layout with Sensors 3, 5, 6, 8 (Figure 54(m) and (n)) followed by the one including all sensors (Figure 54(a) and (b)). Comparing Figures 54(q) and (r) with Figures 54(s) and (t) reveals the higher impact of Sensors 5 and 6 than 4 and 7 in leading the increasing trend. The experimental results are more chaotic than FE for the layouts that do not include Sensors 4-7 (Figure 54(g)-(j)). For these cases, even the sensors located above the notch (Sensors 3 and 8) cannot correct the trends. However, the sensor group effect becomes more important mainly because the location of damage might not be known in advance. In this case, the group effect can be checked to assess the damage growth. As discussed for the individual sensors in Chapter II, it is also possible to localize the damage using the group effect. This can be done by making different groups and checking which sensors lead to a notable and consistent increasing trend if included in the analysis. In the present study, Sensors 5 and 6 clearly have the highest impact on the trends and locate the damage zone.

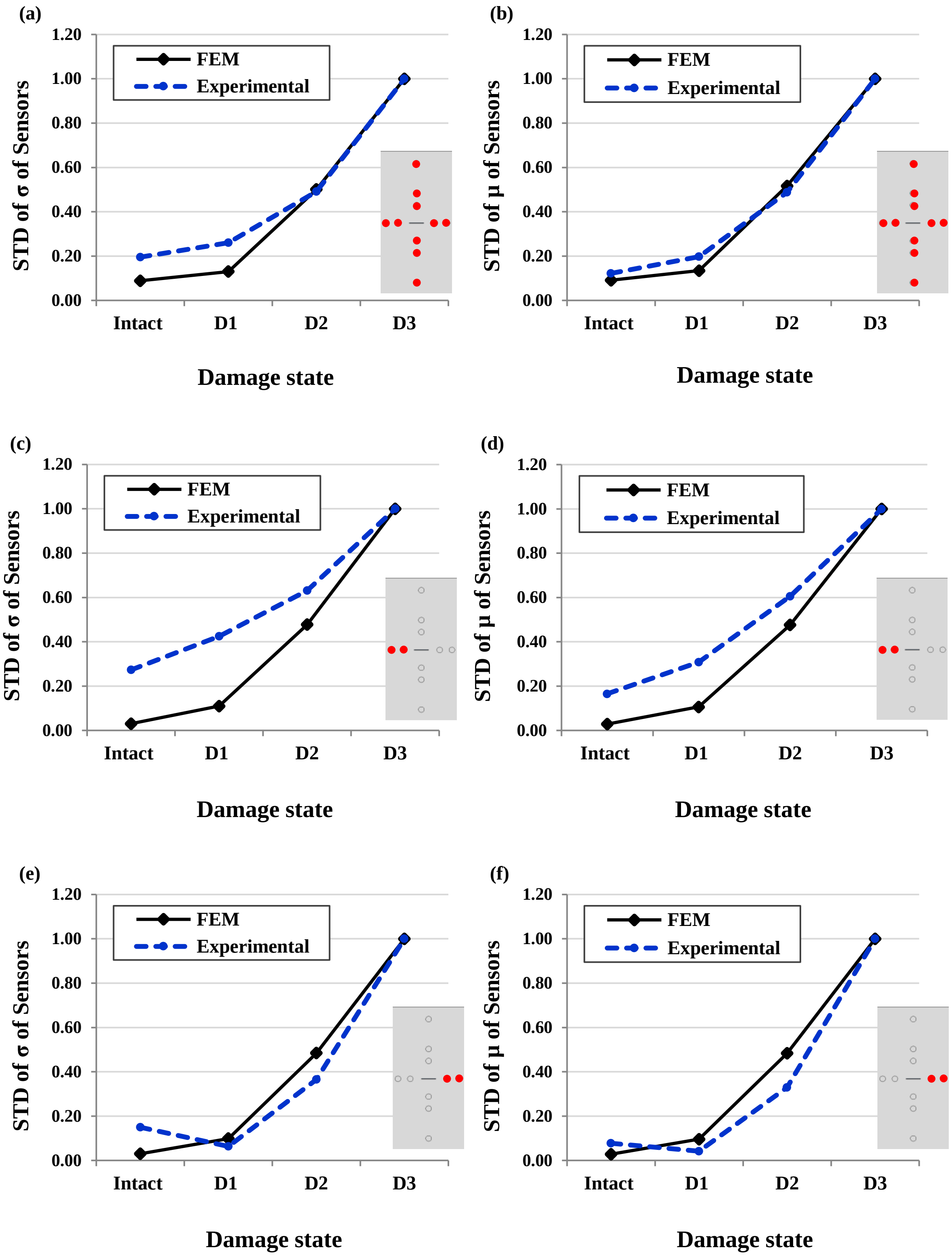


Figure 54. The STD of σ and μ of group of sensors mounted on the plate

Figure 54. (cont'd)

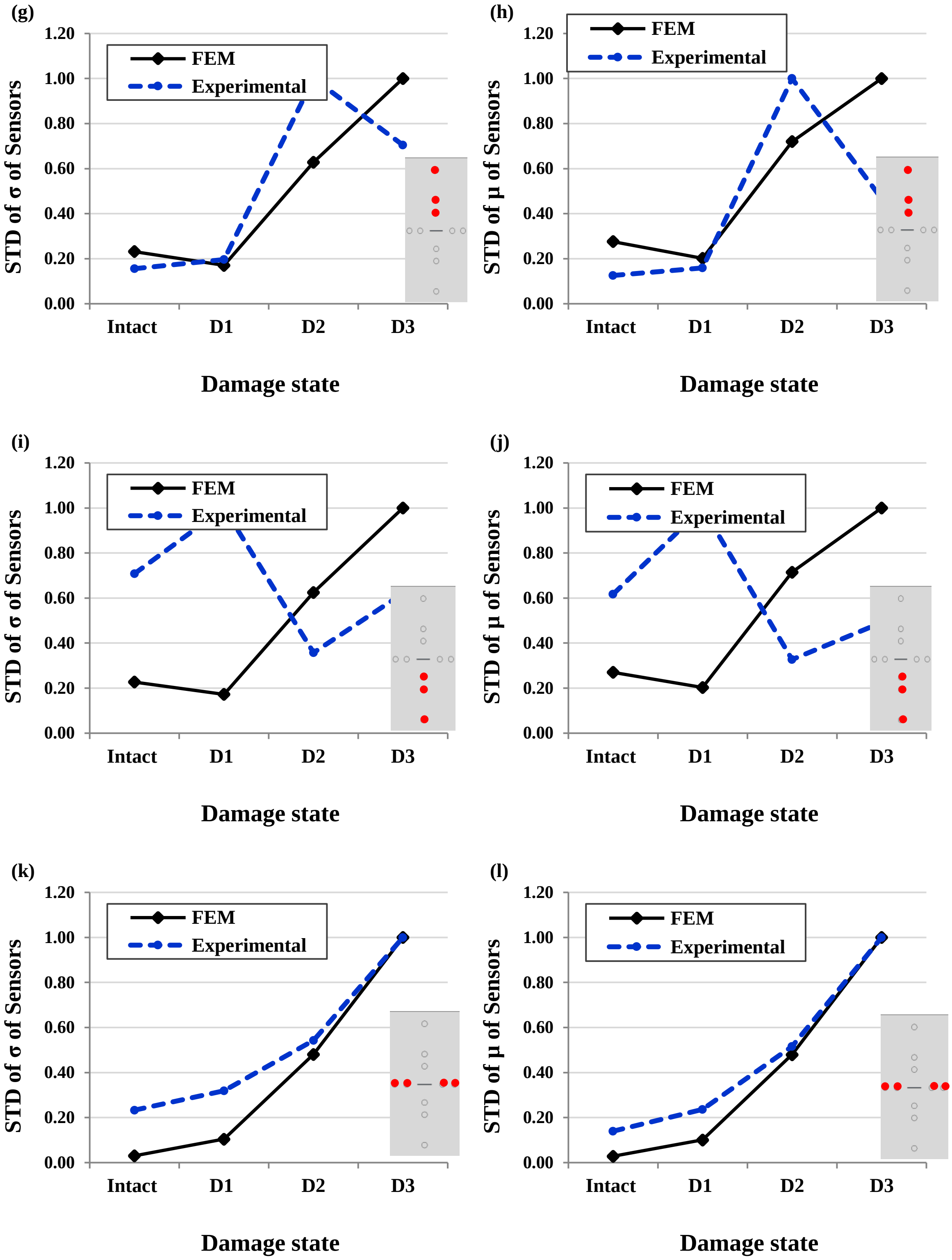


Figure 54. (cont'd)

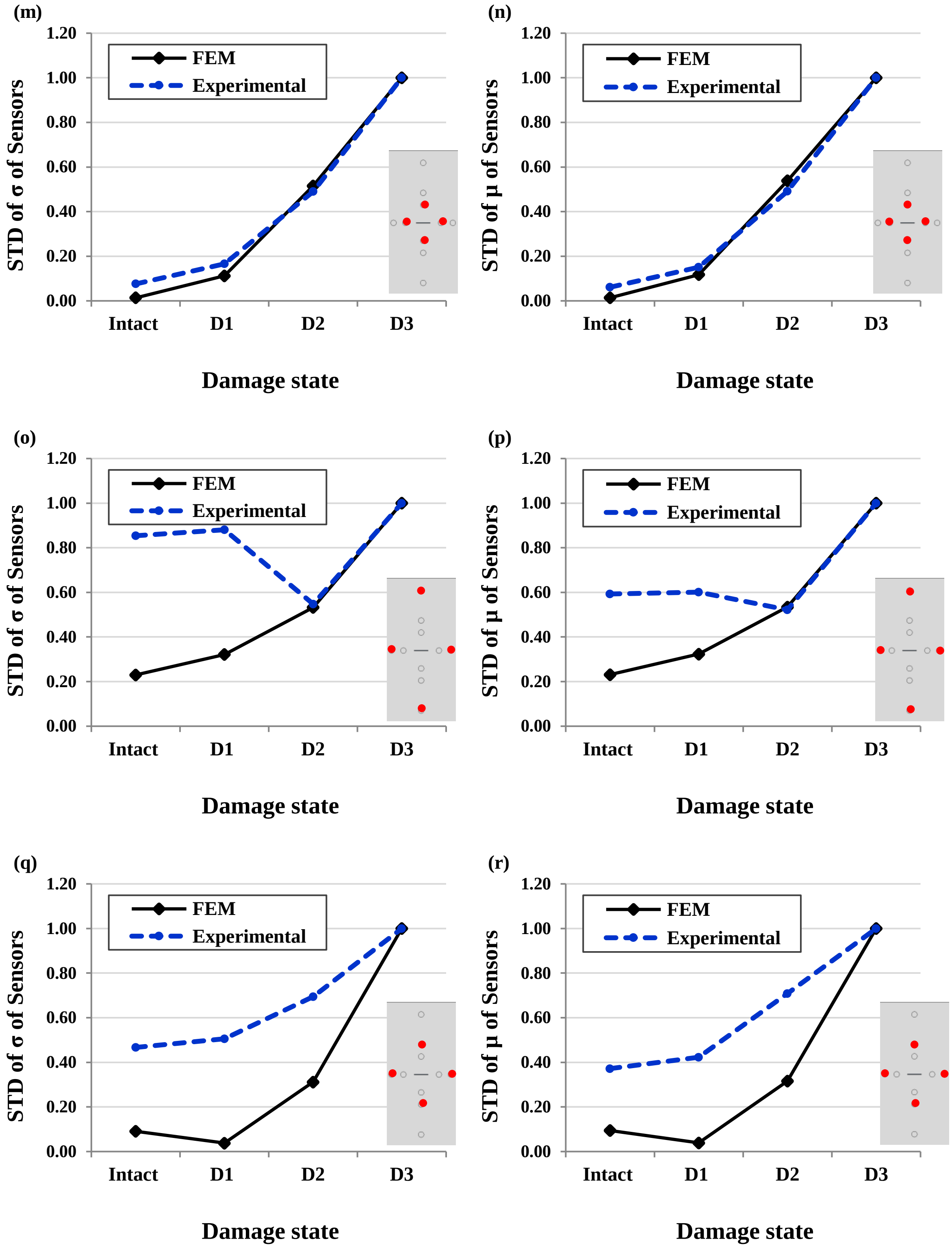
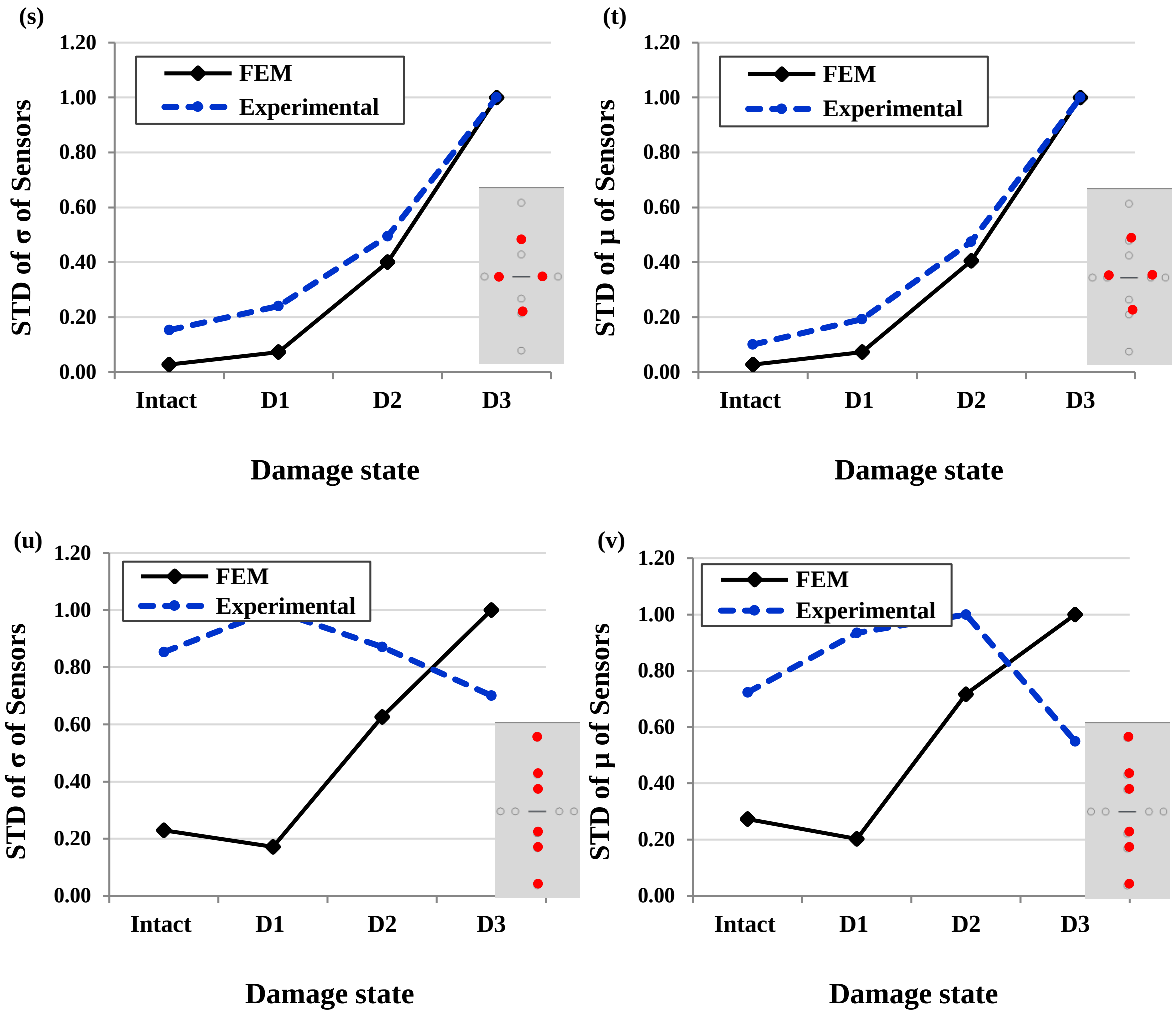
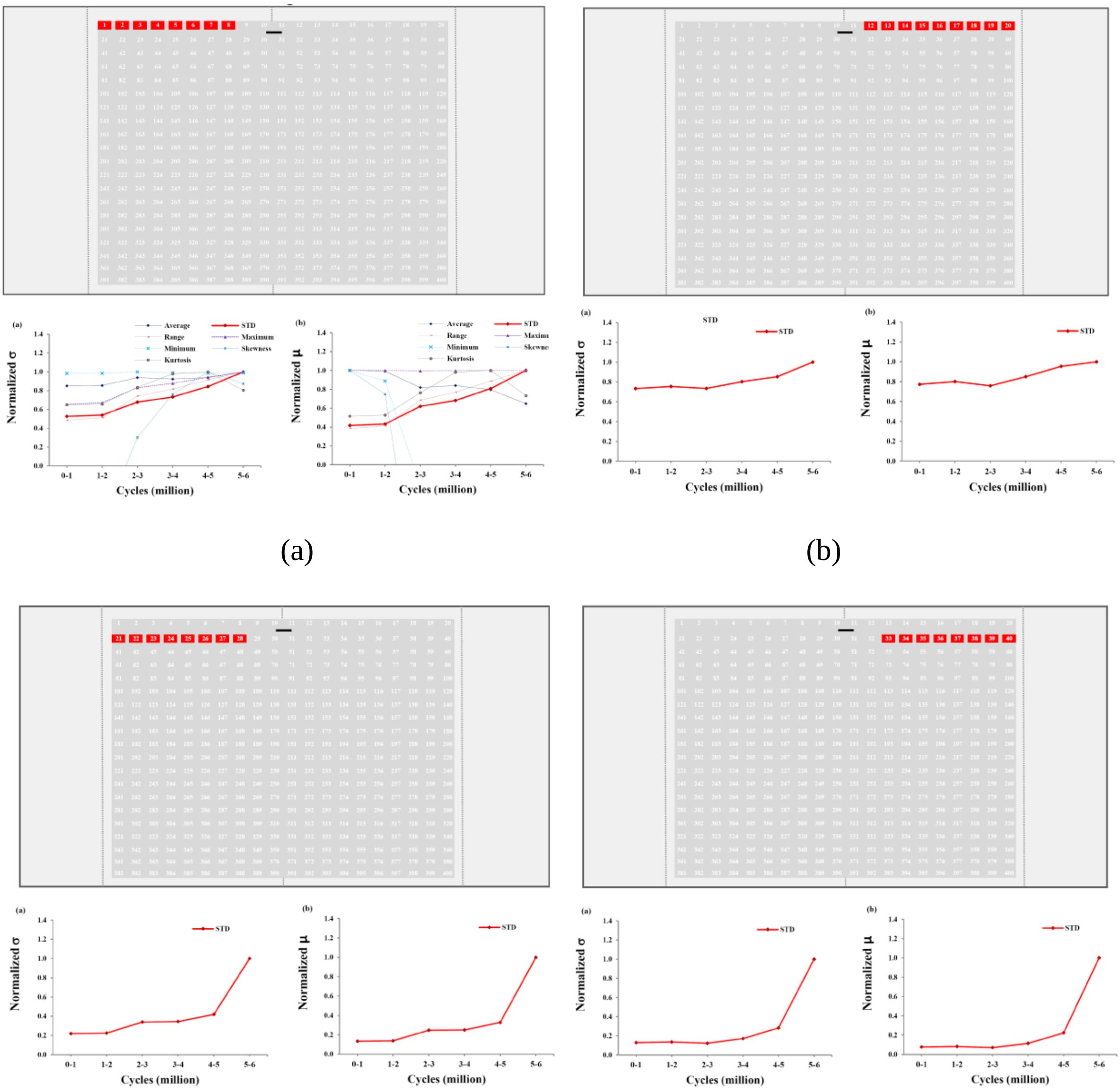


Figure 54. (cont'd)



3.2.2 Distortion-Induced Fatigue Cracking in Bridge Girders

More in depth analyses are carried out for the case of distortion-induced fatigue cracking in bridge girders because of higher number of sensing nodes. Extensive simulations are conducted based on over 70 combinations of the sensing nodes. For each combination, the average, STD, range, minimum, maximum, skewness, and kurtosis of the PDF parameters (μ and σ) are obtained and normalized. Figure 55 presents the variation of the descriptive statistics of μ and σ for some typical configurations.

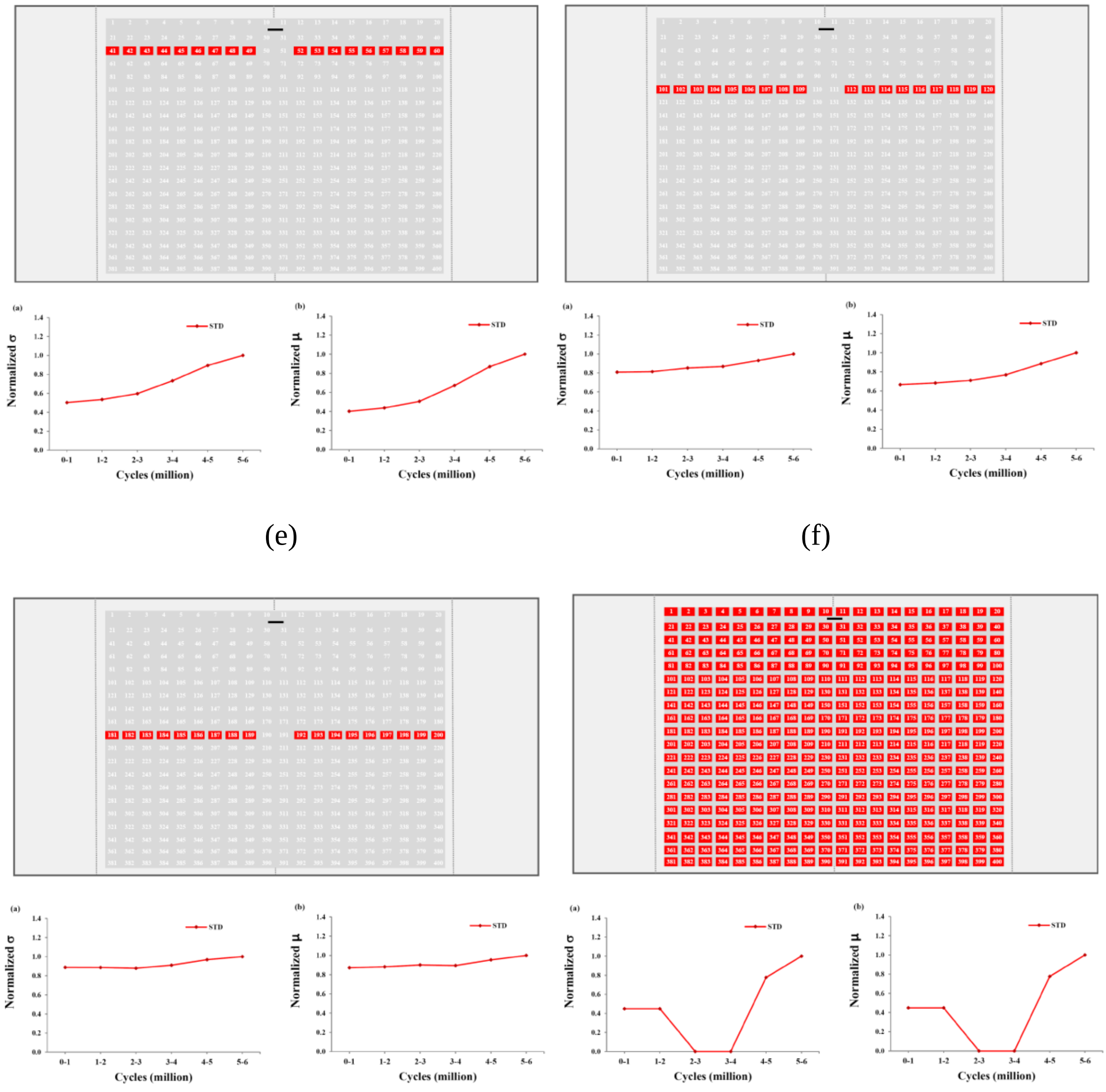


(a) (b)

(c) (d)

Figure 55. Variation of the descriptive statistics of μ and σ of different groups of sensing nodes obtained from the girder FE model

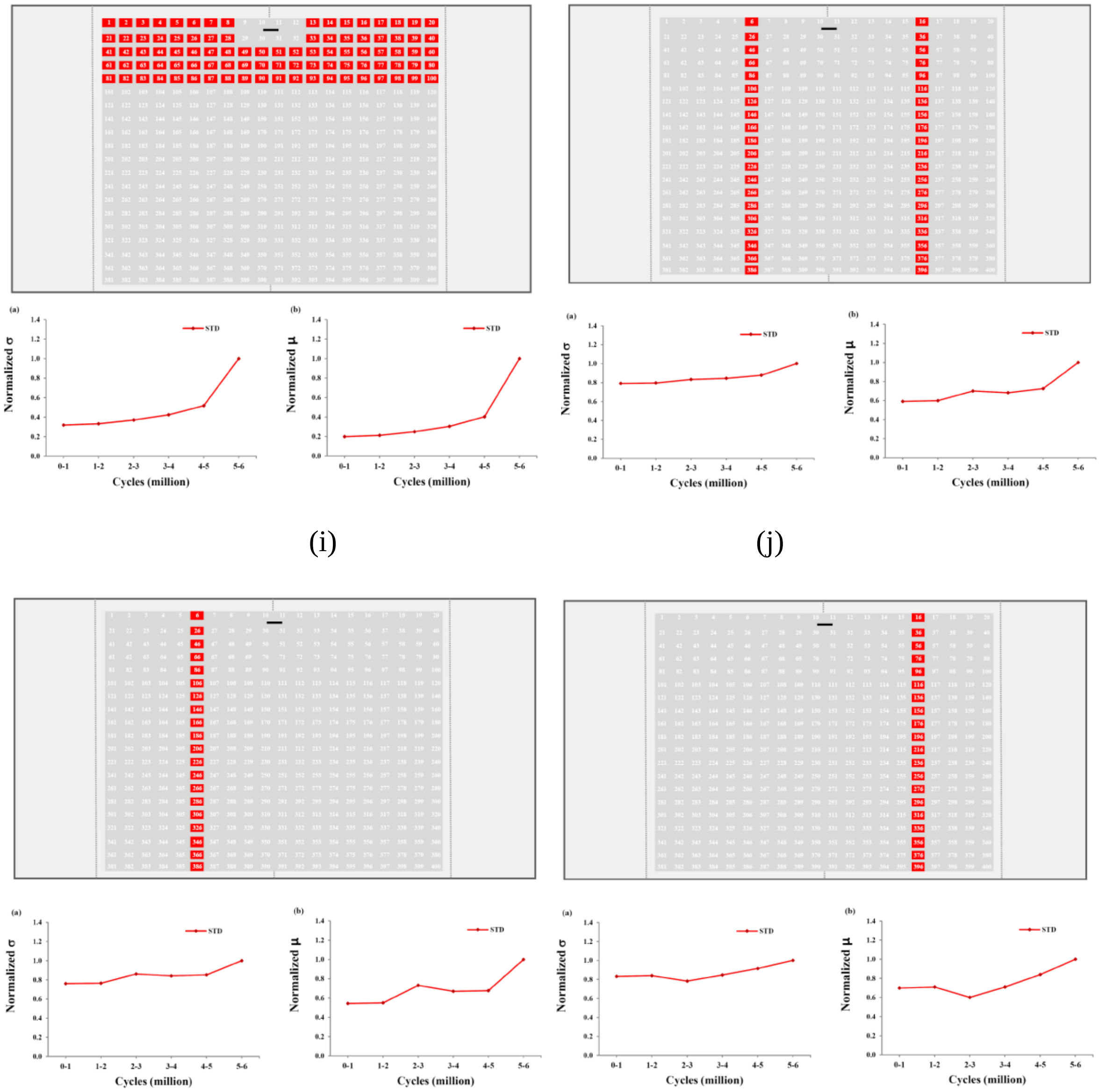
Figure 55. (cont'd)



(e) (f)

(g) (h)

Figure 55. (cont'd)



(i)

(j)

(k)

(l)

Similar to the plate study, the average, range, minimum, maximum, skewness, and kurtosis of the PDF parameters did not have a sound relationship with the damage progression (e.g. Figures 55(a)). Thus, only the variation of the STD of μ and σ of group of sensors is taken into account. The important findings from these figures are as follows:

i. The most acceptable scenario is that the STD of μ and σ of group of sensors increases with damage progression. This is clear for the configurations that do not include the sensors very close to the damage location nor close to the web connection with diaphragm. The results pertaining to the STD of μ of group of sensors are slightly better indicators of damage progression compared to those for σ.

ii. One of the best configurations to detect the damage progression is shown in Figure 55(a). The spacing between the sensing nodes is 20 mm and each sensor has a diameter equal to 10 mm. Accordingly, for the mentioned configuration, the location of the sensors to detect damage should be 50 mm (about 2") far from the critical zone.

iii. Including the sensors adjacent to the damage zone resulted in obtaining chaotic trends. This is due to the singularities around the crack.

iv. The location of web connection with diaphragm is also a critical region. This zone includes the sensors along number 10 and 11. Excluding the sensors located in this zone resulted in obtaining smooth increasing trends.

v. As the distance with the damage zone increases, the incremental rate vanishes or becomes disordered for the sensor groups placed horizontally. This is an expected case as the information sensed by the sensors diminishes.

vi. It can be observed from Figure 55(h) that even including the information from all of the sensors does not provide good indications of the damage progression. In fact, an increasing trend can be observed as soon as the effect of the sensors close to the crack is ignored.

vii. Assuming that the location of the damage is not known, the best scenario seems to be a vertical distribution of the sensors on sides of the web connection (e.g. Figures 55(j)).

It is possible to localize the damage by checking: (1) which sensors provide chaotic response when included in the analysis and (2) in which locations the incremental rates of the STD of μ and σ of group of sensors notably decreases.

3.2.3 Failure of Pavement System

Figure 56 presents the variation of the STD of μ and σ of all of the FE Sensing nodes 3 to 7. As it is seen, the STD of μ and σ of group of sensors increases with damage progression.

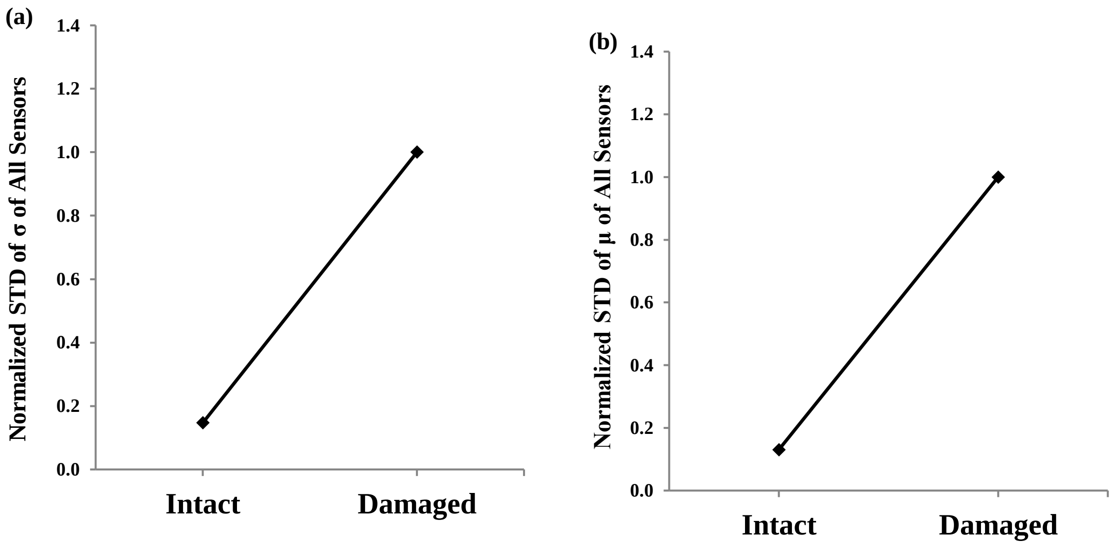


Figure 56. The STD of σ and μ of group of sensors obtained from the pavement FE model

3.3 Artificial Intelligence Approach

A notable limitation of the statistical approach described in Section 3.1 is that the damage progression can be accurately detected by sensors that are located at a specific location with regard to the damage zone. The trends obtained by the sensors far from the damage zone or near the boundaries are not clear. It is desirable to develop a robust method that is capable of fusing the data from the entire sensor network regardless of the distance of the sensors to the damage zone or individual changes of μ and σ. Obviously, damage detection process can be treated as a pattern recognition and classification problem. The solution is to use a classifier which can classify structures either as damaged or healthy. To this aim, an AI-based data fusion system is proposed for damage detection in civil infrastructure. The AI techniques are considered as alternatives to existing traditional methods for tackling real world problems. They determine the model structure by automatically learning from data. In the last two decades, the AI methods have been widely used for tackling problems in civil engineering domain (Flood and Christophilos, 1996; Shahin and Jaksa, 2005). AI has different well-known branches such as artificial neural network (ANN), fuzzy inference system (FIS), adaptive neuro-fuzzy system (ANFIS), and support vector machines (SVM), etc. These techniques have been successfully employed to solve a variety of problems in engineering field including damage detection and structural identification (Szewezyk and Hajela, 1994; Wu et al., 1992; Masri et al., 1993; Elkordy et al., 1993; Zhao et al., 1998). The proposed AI-based data fusion framework consists of three main stages:

(1) Structural simulation,

(2) Information fusion in which features that are expected to characterize different properties of structures are extracted from a network of sensors, and

(3) Fusion of the clustered features. The fusion is performed using the AI methods.

The AI data fusion framework is shown in Figure 57. A flow chart of the proposed approach is also visualized in Figure 58. In the FE simulation phase, different damage scenarios are defined for the given structures. Subsequently, the cumulative time of occurrences at predetermined strain levels are determined for the data acquisition points (sensors). Using the fitted μ and σ parameters, different damage indicator features would be obtained for a specified number of sensors. The defined features simultaneously fuse the information provided by array of scattered sensors. This is mostly of importance for resolving the concerns for installation procedure of sensors in the field, particularly for pavements. Since the method is capable of analyzing the data generated by the randomly distributed sensors, the pebble size SWS can be placed in the mix at the site or tossed into the paving materials during construction. The damage indicator vectors are then used for the calibration of the classifier. Subsequently, a validation phase is performed to check the damage detection performance of the classifier. If convergence condition (desired accuracy) is satisfied, then the process is stopped and the optimal number of sensors is reported.

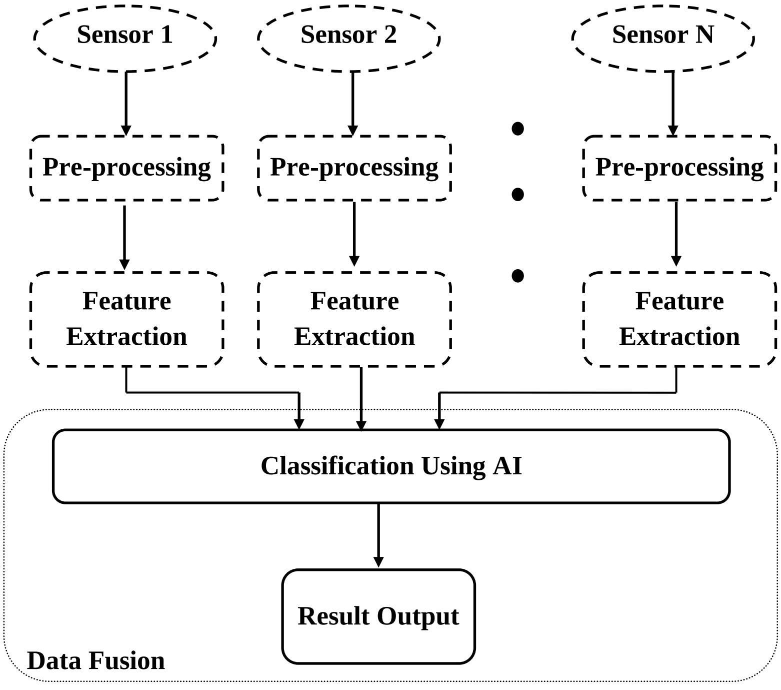


Figure 57. The proposed AI-based data fusion framework

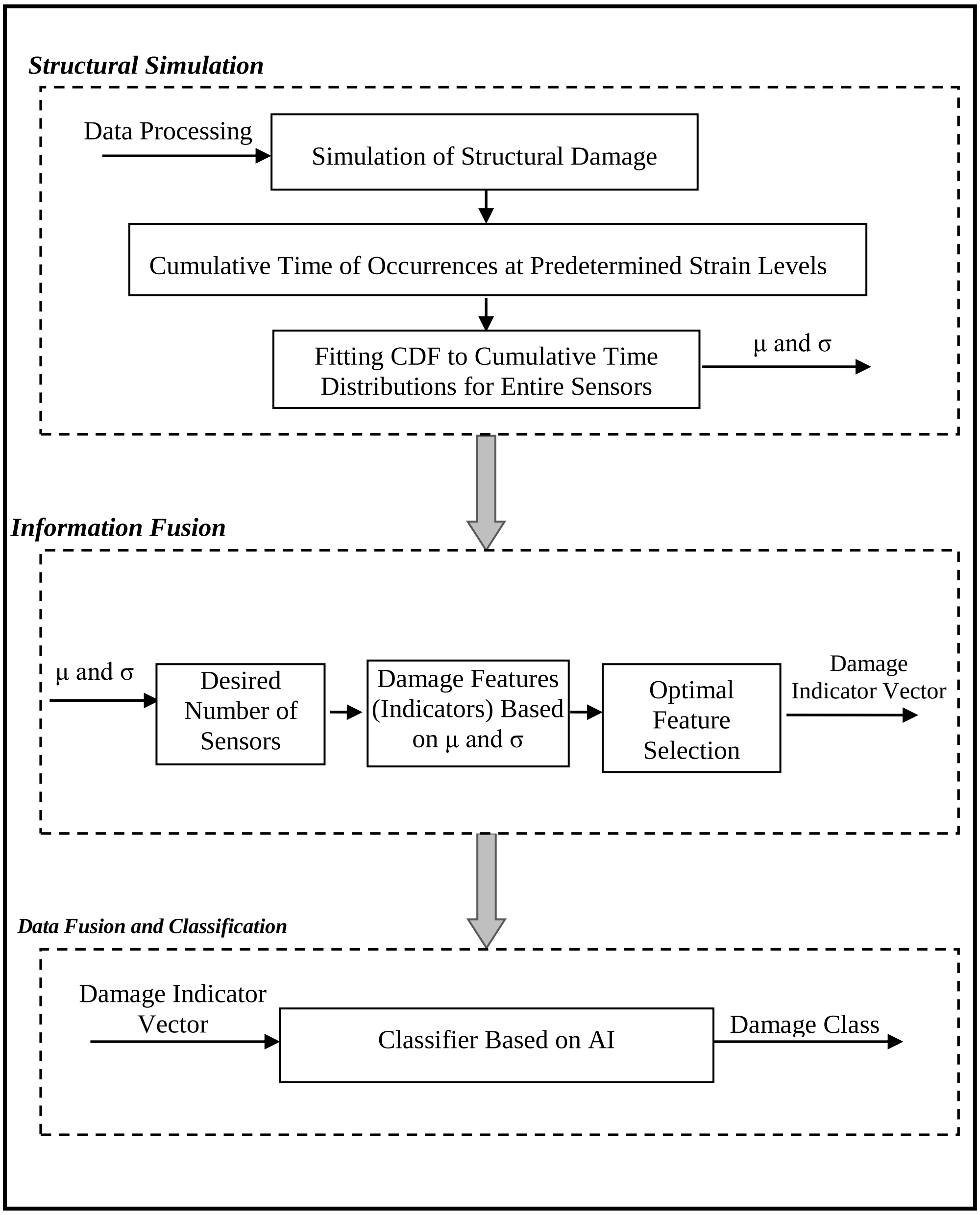


Figure 58. Flow chart of the AI-based data fusion method for damage detection

Extraction of features that efficiently characterize the system is a critical step in multi-sensor data fusion. An appropriate information fusion can reduce imprecision, uncertainties and increase the robustness and reliability of identification. For the first phase of the AI analyses, the damage state is considered to be a function of the following parameters:

$$Damage\ state = f(X, Y, \mu, \sigma) \tag{17}$$

where,

X: Horizontal distance of the sensor from the beam center

Y: Vertical distance of the sensor from the beam center

μ: Mean of the strain distribution

σ: Standard deviation of the strain distribution

However, X, Y, μ, and σ merely include the information for single sensors. In order to include the sensor group effect, a set of new features are introduced to the formulation of the damage state as follows:

$$Damage\ state = f\left(\mu_D, \sigma_D, \mu_S, \sigma_S, Z_{\mu 1}, Z_{\sigma 1}, Z_{\mu 2}, Z_{\sigma 2}\right) \tag{18}$$

where,

μ_D: $\frac{\mu}{\mu_{ave}}$ and μ_{ave} is the average of μ of all sensors for a specific damage scenario that μ belongs to it.

σ_D: $\frac{\sigma}{\sigma_{ave}}$ and σ_{ave} is the average of σ of all sensorsfor a specific damage scenario that σ belongs to it.

μ_S: $\mu - \mu_{ave}$ and μ_{ave} is the average of μ of all sensorsfor a specific damage scenario

σ_S: $\sigma - \sigma_{ave}$ and σ_{ave} is the average of σ of all sensorsfor a specific damage scenario

$Z_{\mu 1}$: $\frac{\mu - \mu_{ave}}{\mu_{STD}}$ and μ_{STD} is the standard deviation of μ of all sensors for a specific damage scenario

$Z_{\sigma 1}$: $\frac{\sigma - \sigma_{ave}}{\sigma_{STD}}$ and σ_{STD} is the standard deviation of σ of all sensorsfor a specific damage scenario

$Z_{\mu 2}$: $\frac{\mu - \mu_{ave}}{\sigma_{ave}}$

$Z_{\sigma 2}$: $\frac{\sigma - \mu_{ave}}{\sigma_{ave}}$

In fact, $Z_{\mu 1}$ and $Z_{\sigma 1}$ are z-score functions. $Z_{\mu 2}$ and $Z_{\sigma 2}$ are functions defined in this study inspired by the form of the conventional z-score function. However, $Z_{\mu 1}, Z_{\sigma 1}, Z_{\mu 2}$, and $Z_{\sigma 2}$ are termed as Z-functions in this study. As can be observed, the considered input variables efficiently take into account the sensor group effect which would improve the detection performance, specifically for low number of sensors. For example, assume the case where 10 sensors with random locations are used for the damage detection and one is located at the corner of the beam. For sure, μ and σ of the sensor at the corner will not have sufficient information for detecting the damage state. This is while by using parameters such as $\mu_D, \sigma_D, \mu_S, \sigma_S$, and Z-functions the information from the sensors distributed along the beam would enhance the classification accuracy.

On the computing domain, numerous probabilistic, least squares, and AI methods have been used for multi-sensor information fusion in I/SHM (Quadri and Sidek, 2013). This is while, the significant data fusion capabilities of the evolutionary computation techniques in diagnosis of the health condition of structures have not yet been investigated in the literature. This research proposes a hybrid method that combines a robust branch of evolutionary computational, called genetic programming (GP), with logistic regression (LR) algorithm. Moreover, a probabilistic neural network (PNN) approach based on Bayesian decision is also employed to improve the damage detection accuracy.

3.3.1 Evolutionary Computation

Despite acceptable performance of ANNs, they are considered as black-box models. That is, they are not capable of generating practical prediction equations. Another limitation of ANNs is that their structure should be defined in advance (Alavi and Gandomi, 2011). Inspired by the natural evolution and the Darwinian concept of “survival of the fittest”, evolutionary computational (EC) methods are well-known branches of AI. Some of the subsets of EC are evolutionary strategies (ESs) (Schwefel, 1975) and evolutionary programming (EP) (Fogel et al., 1996). These techniques are collectively known as evolutionary algorithms (EAs). In general, an EA consists of an initial population of random individuals improved by a set of genetic operators (e.g., reproduction, mutation and recombination). The individuals are encoded solutions in form of binary strings of numbers evaluated by some fitness functions (Coello et al., 2007). Improvement of the population is a process to reach the fittest solution with the maximum convergence. Typically in an EA, a population of individual is randomly created and then the members are ranked according to a fitness function. The members with the highest fitness ranking are given a higher chance to become parents for the next generation (offspring). The approach used to generate offspring from the parents is referred to as the reproduction heuristic. Then selected members are randomly transformed into new members via mutation, recombination or crossover. These steps are repeated until the convergence conditions are satisfied and the fittest member is selected (Koza, 1992; Fogel et al., 1996; Coello et al., 2007). The differences between EAs are in the way that they represent the individual structures, types of selection mechanism, forms of genetic operators, and measures of performance.

Genetic Algorithm (GA) has been shown to be a robust EA for dealing with a wide variety of complex civil engineering problems (Hung et al., 2012; Unal et al., 2014). GP is a specialization of GA where the encoded solutions (individuals) are computer programs rather than binary strings (Banzhaf et al., 1998). Figure 59 shows a comparison of the encoded solutions (individuals) by GA and GP. In GP, inputs and corresponding output data samples are known and the main goal is to generate predictive models relating them (see Figure 60) (Weise, 2009). GP has several advantages over the other AI techniques such as ANNs. A notable feature of GP and its variants is that they can produce highly nonlinear prediction equations without a need to pre-define the form of the existing relationship (Alavi and

Gandomi, 2011). However, application of GP and its variants to structural damage detection is conspicuous by its near absence (Harvey and Todd, 2013).

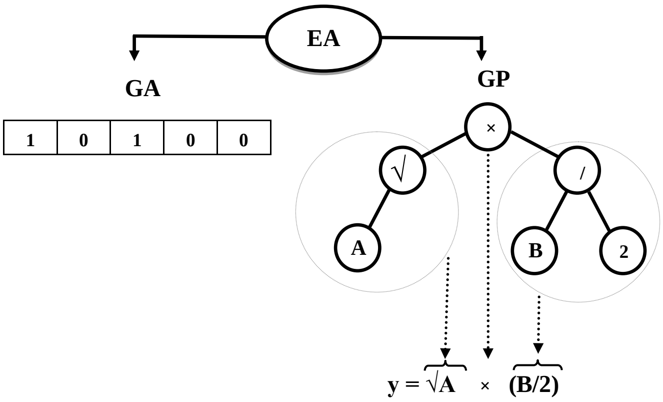


Figure 59. A comparative illustration of encoded solutions by GA and GP

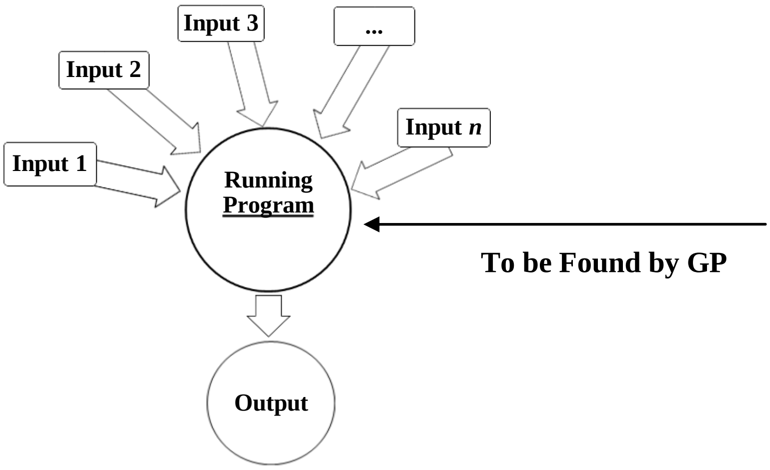


Figure 60. Conceptual scheme of input-process-output (IPO) in GP

The GP solutions are represented in different ways such tree-shaped, graph-shaped and linear encodings (Alavi and Gandomi, 2011). Tree-shaped is the mostly widely used representation of the GP programs. However, in addition to classical tree-based GP, there are other types of GP where programs are represented in different ways. These are linear and graph-based GP (Banzhaf et al., 1998). The emphasis of the present study is placed on a linear-based GP technique called gene expression programming (GEP) (Ferreira, 2001). The programs evolved by linear variants of GP are represented as linear strings that are decoded and expressed like nonlinear entities (Oltean and Groşan, 2003). A linear GP system can run several orders of magnitude faster than comparable tree-based interpreting systems. The enhanced speed of the linear variants of GP permits conducting many runs in realistic timeframes. This leads to deriving consistent and high-precision models with little customization (Poli et al., 2007; Gandomi et al., 2011; Alavi et al., 2013). Comprehensive descriptions of the GP and GEP algorithms can be found in (Alavi and Gandomi, 2011).

However, one of the limitations of the GEP algorithm is that it is basically designed for binary classification. Thus, this study proposes a hybrid approach called GPLR that combines the GEP and LR algorithms for classifying multi-stage damage states. In GPLR, GEP is utilized for the development of pattern recognition models and LR is used to assign probabilities to model scores to find the most probable damage state. The steps of the GPLR approach are described below.

The GEP algorithm is trained for each of the classes in the corresponding sensor configurations. Consequently, separate models are derived for each class. Each of the optimal models derived for a particular sensor configuration operates independently and predicts whether the given information belongs to a specific class or not. In order to combine the models to evaluate the final outcome for different damage classes, a probabilistic approach based on LR is utilized. The idea is to substitute the highly nonlinear models evolved by GEP with an array of predictor variables in the LR original formulation and therefore assign the probabilities to obtained models. Once the probability for each damage state is known, it is possible to make categorical predictions about different states. The basic LR predicts a logit transformation of the probability of presence of the dichotomous outcome variable as a linear relationship with a set of independent variables (Youn and Gu, 2010). In other words, assume that for a damage structure:

$$Y = \begin{cases} 0, & Intact \\ 1, & Damage\ State\ 1 \end{cases} \tag{19}$$

Thus, Y is the binary outcome. If $P(Y = 1\,|X)$ is the probability of Y to be 1 and X is an array of predictor variables, the logistic regression of Y on X is stated as follows:

$$logit\big(P(Y = 1\,|X)\big) = \ log\left(\frac{P(Y=1\,|X)}{1-P(Y=1\,|X)}\right) = ln\left(\frac{P(Y=1\,|X)}{P(Y=0\,|X)}\right) = \alpha\ +\ \beta X \tag{20}$$

$\frac{P(Y=1\,|X)}{1-P(Y=1\,|X)}$ is the odds of being damaged defined as the ratio of the probability of being damaged over the probability of being intact. α and β are, respectively, the slope and intercept of the regression line. X can be a combination of predictor variables (e.g., x_1, .., x_n). In this case, the term βX in the original logistic regression equation will be replaced by $\beta_1 x_1 + \ldots + \beta_n x_n$. Often the cut-off probability (rounding threshold) of 0.50 is used to determine the classes, e.g., intact (class 0) or damaged (class 1). Generally, for a binary classification, P values equal to or greater than 0.50 are classified into class 1 while gusset plates with P of failure less than 0.50 are classified into class 0. If the logistic equation is solved for P, the following formula will be obtained for the evaluation of the probabilities:

$$P = \frac{1}{1+e^{-(\alpha\,+\,\beta X)}} \tag{21}$$

As it is seen, the above method is mainly applicable to binary cases. Thus, in order to adopt the logistic concept for the existing problems with different damage classes, the GEP-based model for each class is replaced with X in the original logistic equation (Eq. (21)). As it is, in this case, X would be a nonlinear combination of the predictor variables. As can be observed from Eq. (20), log odds ($ln\left(\frac{P}{1-P}\right)$) is the core of LR and therefore, it can be used to derive the α and β coefficients for each model. To this aim, a number of quantiles or bins are first defined for the model outputs. Then, the distribution of both positive (1s) and negative (0s) categories is determined for each bin. The probability of positive cases (P) for each bin is equal to the ratio of the number of positive cases to the total cases. Similarly, the probability of negative cases (*1-P*) for each bin can be determined. This procedure is done for all of the bins. Using these values, the odds ratio and log odds for all the bins are obtained. Finally, by plotting the log odds values on the y-axis against the model output in the x-axis and performing a weighted linear regression, the slope (α) and intercept (β) of the regression line for each of the models can be easily derived. By substituting these parameters into Eq. (21), the probability of each damage model can be evaluated. Apparently, the damage state with higher probability is the dominant state. The application of the proposed approach is not only limited to the investigated cases but also the entire classification systems based on the GP method.

3.3.2 Probabilistic Neural Network

A major drawback of the conventional ANNs pertains to the time-consuming iterative procedure required during training of the network to obtain the optimal learning parameters (Yan and Miyamoto, 2003). To overcome such limitation, PNN has been proposed by Specht (1990). PNN is essentially based on the well-known Bayesian classification by combining the Bayes strategy for decision making with a non-parametric estimator for the probability density functions (PDFs). It can be used for direct estimation of posterior probability densities and pattern classification (Goh, 2002). There are some studies in the literature focusing on the application of PNN to the structural damage identification (e.g., Lee and Yun, 2007; Ni et al., 2001). Some of the advantages of PNN over the conventional ANN are (Adeli and Panakkat, 2009): (i) The PNN execution is generally much faster than the conventional neural network because it does not require a separate training phase, (ii) Training of the PNN algorithm with new training data is fairly easy, and (iii) This method provides good classification performance in domains with noisy data. Despite significant capabilities of

PNN, its application to the interpretation of the data generated by WSN or SWS is totally new and original.

The PNN algorithm adopts a Bayes decision rule, i.e. it considers a test vector x with m dimensions that belongs to one of the classes $C_1, C_2, \ldots, C_k$. From the multi-category classifier decision, x belongs to C_k if (Yan and Miyamoto, 2003):

$$P_k L_k F_k(x) > P_j L_j F_j(x) \qquad \text{for all k}\neq\text{j} \tag{22}$$

where $F_k(x)$ and $F_j(x)$ are the PDFs for classes C_k and C_j respectively, L_k is the loss function associated with misclassifying the vector as belonging to class C_j while it belongs to class C_k, L_j is the loss function associated with misclassifying the vector as belonging to class C_k while it belongs to class C_j, P_k and P_j are the prior probabilities of occurrence of the classes C_k and C_j, respectively. In many situations such as damage assessment problem, L and P are usually assumed to be equal for all classes. Hence, the key to using the decision rule given by Eq. (22) is to estimate the PDFs from the training patterns. PNN operates based on the concept of a nonparametric estimation technique known as Parzen windows classifier and its application to Bayesian statistics to estimate the PDFs for each classification class (Goh, 2002; Yan and Miyamoto, 2003; Adeli and Panakkat, 2009). If the jth training pattern for class C_k is x_j, then the Parzen estimate of the PDF (F_k) for class C_k is (Yan and Miyamoto, 2003):

$$F_k(x) = \frac{1}{(2\pi)^{m/2}\eta^m}\frac{1}{n}\sum_{j=1}^{n} exp\left(-\frac{\left(x-x_j\right)^T(x-x_j)}{2\eta^2}\right) \tag{23}$$

or

$$F_k(x) = \frac{1}{(2\pi)^{m/2}\eta^m}\frac{1}{n}\sum_{j=1}^{n} exp - \left(\frac{\left\|x-x_j\right\|^2}{2\eta^2}\right) \tag{24}$$

where n is the number of training patterns of class C_k, m is the input space dimension, and η is an adjustable smoothing parameter. In fact, the parameters x_j and η represent the center and spread (or volume) of the Gaussian bell curve, respectively. The parameter η must be determined experimentally (Goh, 2002; Yan and Miyamoto, 2003; Adeli and Panakkat, 2009). Eq. (23) implies that any smooth density function can be expressed simply as the sum of small multivariate Gaussian distributions.

Figure 61 shows a typical representation of the PNN architecture. As it is seen, the PNN architecture consists of four layers: (1) input layer, (2) pattern layer, (3) summation layer, and (4) a single-node output layer. The input layer includes the m input variables (x_1, $x_2,\ldots, x_m$). All of the variables x are distributed by the input layer to all the neurons in the pattern layer. These layers are fully connected so that one neuron is allocated for each pattern in the training set. Thus, the number of nodes in the pattern layer is equal to the number of available training input vectors (n). The number of nodes in the summation layer is equal to the defined classes. A dot product operation is applied by each pattern neuron j to the input pattern vector x with a weight vector w_j such that $A_j= xw_j$. A transfer function in the form of $exp\left(\frac{A_j-1}{\eta^2}\right)$ is applied to A_j *and* then it is outputted to the summation neuron (Goh, 2002; Yan and Miyamoto, 2003). As both x and w_j are normalized to unit length, this is equivalent to performing the dot product operation:

$$exp\left(-\frac{(w_j-x)^T(w_j-x)}{2\eta^2}\right) \tag{25}$$

which can be written as:

$$exp\left(\frac{2x^Tw_j-x^Tx-w_j^Tw_j}{2\eta^2}\right) \tag{26}$$

Since x and w_j are normalized to unit length, the dot products x^Tx and $w_j{}^Tw_j$ are equal to 1. Thus, the above equation can be expressed as:

$$exp\left(\frac{x^Tw_j-1}{\eta^2}\right) \tag{27}$$

It can be observed that the transfer function $exp\left(\frac{z_j-1}{\eta^2}\right)$ and the exponential term in Eq. (23) have the same form. It is necessary to compute this exponential term for each of the neurons in the pattern layer (Goh, 2002; Yan and Miyamoto, 2003).

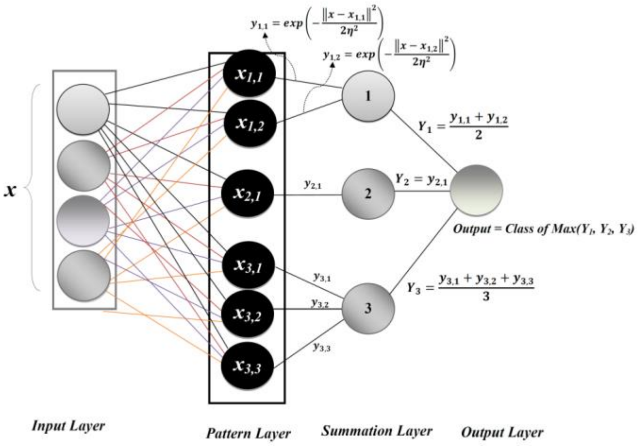


Figure 61. A typical architecture of PNN with 4 input variables, 6 training instances, and 3 defined classes

The summation of units in the summation layer simply sums all the inputs from the pattern units that correspond to a given class. This results in the determination of the PDF of each category given by Eq. (23). In fact, there is one summation- layer neuron for each category. For instance, the output of the summation layer neuron corresponds to the class C_k is (Yan and Miyamoto, 2003):

$$C_k(x) = \sum_{j=1}^{n_k} exp\left(\frac{(x.w_{kj}-1)}{\eta^2}\right) \tag{28}$$

Comparing Eq. (22) with Eq. (28), the constant term of Eq. (22) can be ignored. This results in the determination of the PDF of each category given by Eq. (22). It is worth mentioning that the transfer function is not limited to being an exponential or Gaussian. A number of different transfer functions are presented by Specht (1990). The smoothing parameter, η, has the same value throughout the network. Training of PNN can be regarded as finding the best η value for a set of vectors $\boldsymbol{x}$ which maximizes the classification accuracy of another independent set of vectors (testing data). The fourth layer of the network is the output-layer. This layer gives a binary output value corresponding to the highest PDF. The highest PDF indicates the best classification or category choice for that pattern. In general, a PNN for M classes can be defined as follows:

$$Y_j(x) = \frac{1}{n_j}\sum_{i=1}^{n_j} exp\left(-\frac{\|x-x_{j,i}\|^2}{2\eta^2}\right), j = 1, \dots, M \tag{29}$$

where n_j denotes the number of data points in class j. PNN assigns x into class m if $Y_m(x) > Y_j(x)$, $j \in [1, \dots, M]$. $\|x-x_{j,i}\|^2$ is calculated as the sum of squares.

3.3.3 Performance Evaluation of the AI Data Fusion Systems

In order to verify the efficiency of the proposed data fusion damage identification system, its performance is evaluated on a simply supported beam under three-point bending and a complicated case of gusset plate of bridge. The performance analysis is based on extensive numerical studies. First, the results of the PNN analysis are presented for both of the beam and gusset plate cases. The performance of the developed GPLR method is assessed on the gusset plate.

3.3.3.1 The PNN Method: Case Studies of Simply Supported Beam and Bridge Gusset Plate

The performance verification with PNN is done in two stages. At the first stage, the method is applied to simply supported intact and damaged concrete beams under three-point bending. The second stage is focused on the verification of the method for a much more complicated case which is a bridge gusset plate. A series of FE simulations are conducted for both of the cases and the derived models are considered as representatives of the real structures. Static strain measurements at specified or randomly selected locations are used to calibrate the supervised learning algorithm. Different techniques can be utilized to solve linear elastic fracture mechanics problems (Elvin et al., 2003; Elvin and Leung, 1999). However, the FE method is considered reliable because of its applicability to most elasticity problems, as well as its ease of implementation (Elvin et al., 2003; Rosenstrauch et al., 2013). Herein, the structures under consideration are assumed to be linearly elastic and are analyzed as 3D FE models using ABAQUS Version 6.12-3. In the FE simulations of beam, axial strains at the sensor locations are of interest. That is to say, only in-plane strains are considered to produce electric charge. For the gusset plate, maximum principle strains are used due to the complexity of the geometry. As a fairly similar loading pattern to real traffic load distribution, the input loading type is in the form of half-sine loading (Figure 29). There are data acquisition points remote to the damage sites that will not be any influenced by the damage. Thus, in real conditions, the sensors will not record any information. To consider this issue in this study, the data for such as acquisition points are not included in the analyses. It is worth mentioning that main focus of this research is to propose a pioneering damage detection concept and then verify its performance for a simple case and then for a complicated structure, such as the gusset plate. Thus, the issue of controlling different mesh sizes, changing the material properties, loading, etc is not within the scope of this work.

3.3.3.1.1 Failure of Simply Supported Beam

Figure 62 shows the geometry and loading of the investigated beams. The beam is modeled using three dimensional eight node linear brick elements (C3D8R). For Case I, the sensor locations are assumed to be randomly distributed but not very close to the notch (crack). Thus, sophisticated modeling (meshing) of the notch is not required. The FE model and mesh are shown in Figure 63. The model of intact beam consists of 158 C3D8R elements corresponding to 380 nodes. Taking into account the size of the real smart pebble sensors and also to facilitate the implantation of the algorithm, a fairly course meshing is considered. Thus, each of the elements of the FE model can be a possible sensor location. Alternatively, in case of applying a finer mesh, the averages of the strains at elements equivalent to the probable sensor volume can be used. It is assumed that the damaged zone extends through the entire width of the beam. The damage is simulated by removing rectangular sections of different sizes at the centre of the beam. In this model, uniform pressure load is applied on the top of the beam. The amplitude of the load is optimally taken in a way that there would be a sense of stress throughout the beam. However, the material and geometry properties used in the model are as given below.

- E = 29000MPa
- Poisson Ratio = 0.2
- Density = 2400 kg/m³
- Dimensions (L, D, W): 400×100×10mm
- Load (Pressure): 10 MPa
- Load Frequency = 2Hz

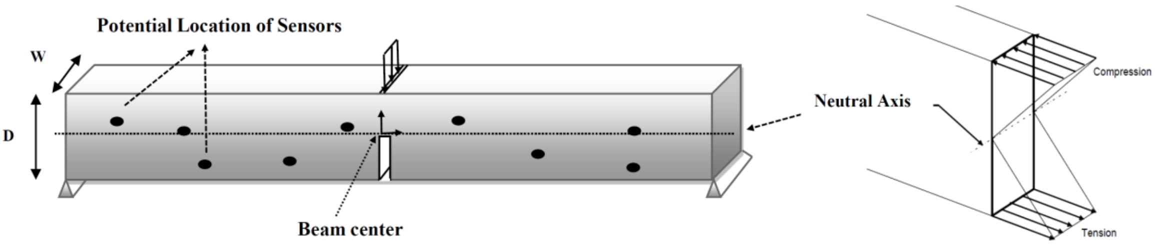


Figure 62. Geometry and loading of a damaged simply supported concrete beam

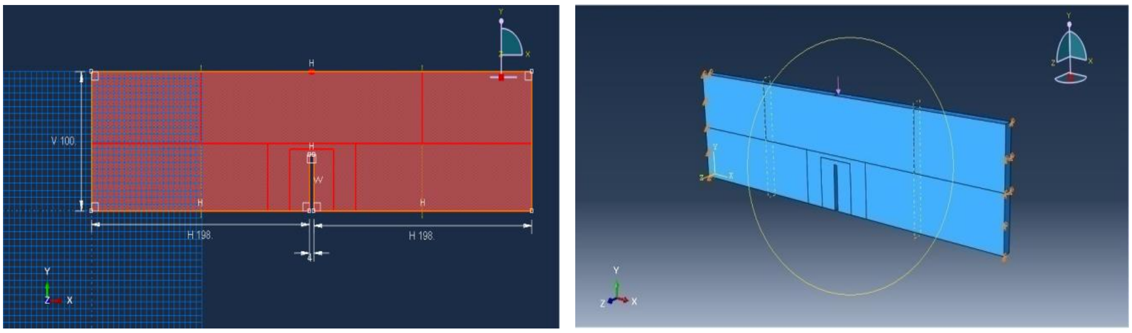


Figure 63. The FE model of a simply supported beam

Data Processing: The FE model of the beam is used as a representation of the real structure. Static strain data of the surface nodes is extracted from the simulations and used to derive the feature vectors for the damage state classification process. To validate the efficiency of the proposed approach, different damage scenarios are considered. The severity of damage is defined by changing the notch to the beam depth (a/D) as follows:

1: Intact beam (a/D = 0.00)

2: a/D = 0.05

3: a/D = 0.1

4: a/D = 0.2

5: a/D = 0.3

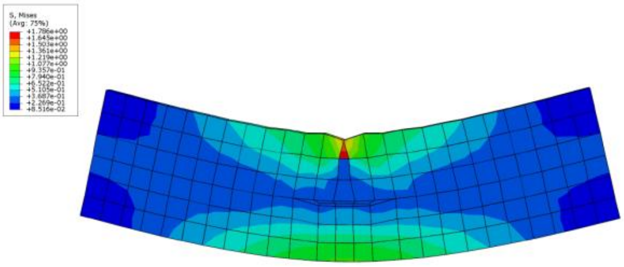
6: a/D = 0.4

7: a/D = 0.45

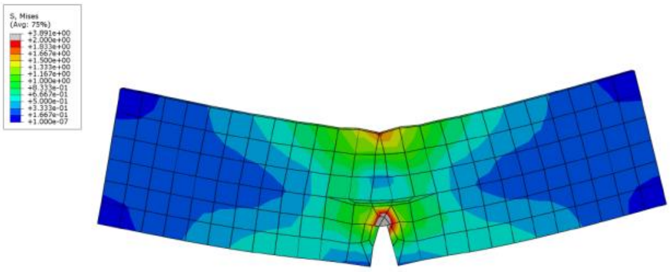
The FE simulation results for some of the damage scenarios (i.e., Classes 1, 5, 7) are shown in Figure 64. The presented results belong to the maximum loading step. As discussed before, SWS has a series of memory cells (gates). Each of these gates cumulatively stores the duration of strain events at a preselected levels. In general, the number of gates is dependent on the nature of the problem and the material. In this study, a typical number of 10 is considered for the strain levels to efficiently cover the lower and upper limits of the strain values extracted from the FE simulations of seven damage scenarios. The minimum and maximum of the strain values are about 0.122 and 288.965 με, respectively. Thus, for the analysis, the lower and upper bounds of the strain levels are, respectively, set to 0.100 and 300.000 με. Ten gates are considered and therefore the difference between the strain levels is 33.322 με. The preselected strain levels are shown in Table 12. The obtained data are subsequently used to calibrate the PNN classifier.

Table 12. The preselected strain levels considered for the beam failure analysis

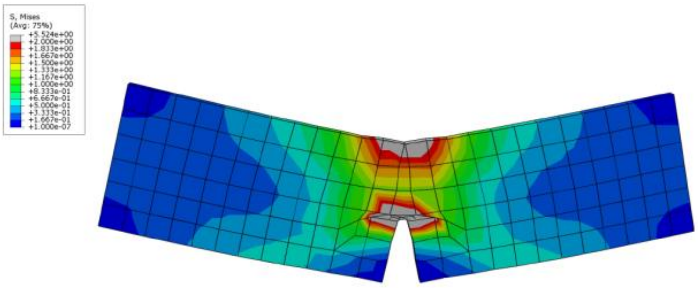
Gate number	Strain Level (με)
1	0.100
2	33.422
3	66.744
4	100.067
5	133.389
6	166.711
7	200.033
8	233.356
9	266.678
10	300.000



(a) Intact beam(Class 1: a/D= 0)



(b) Damaged beam (Class 5: a/D= 0.3)



(c) Damaged beam (Class 7: a/D= 0.45)

Figure 64. The FE simulation results for different damage scenarios for Case I (Von Mises stress distribution)

3.3.3.1.2 Failure of Gusset Plate of Bridge

In 2007, the deck truss portion of the I-35W Highway Bridge in Minneapolis, Minnesota collapsed. According to The National Transportation Safety Board (NTSB, 2008), the collapse occurred due to a bending instability in the U10W gusset plates. Due to the importance of this catastrophic event, the failure analysis of the I-35W Highway Bridge has been the focus of many studies (e.g. Liao and Okazaki, 2009; Holt and Hartmann, 2008; Hao, 2010; Liao et al., 2011). In this study, a gusset plate with a structure similar but not exactly identical to that of the I-35W Highway Bridge is considered for further verification of the proposed approach. The 3D FE model for the gusset plate is shown in Figure 65. The dimension of the joint is given in Figure 66. Besides, Figure 67 describes the location and the magnitude of the loading. The load magnitude is taken equal to 10% of the critical loading at the time of bridge collapse (Liao and Okazaki, 2009). The plate is modeled using three dimensional linear tetrahedral elements (C3D4). The model of the gusset plate consists of about 50000 C3D4 elements corresponding to about 25000 nodes. However, the material and geometry properties used in the model are as given below.

- E = 200GPa (Steel ASTM-A36)
- Poisson Ratio = 0.3
- Density = 7800 kg/m^3
- Load Frequency = 0.5Hz

The thickness of the gusset plate is 12.7 mm (0.5"). The diameter for the data accusations nodes (potential sensors) is equal to 10 mm. The average of the max principle strains at the nodes within the sensor specified area is taken as the representative strain value for each sensor. Several sensors are considered for the analysis. The distance between the sensors is taken 80mm. Considering the dimension of the plate, 28 and 16 sensors are defined in the horizontal and vertical directions, respectively. Thus, the maximum number of data acquisitions nodes is 28 × 16 = 448. The location of sensors in the gusset plate can be seen in Figure 65.

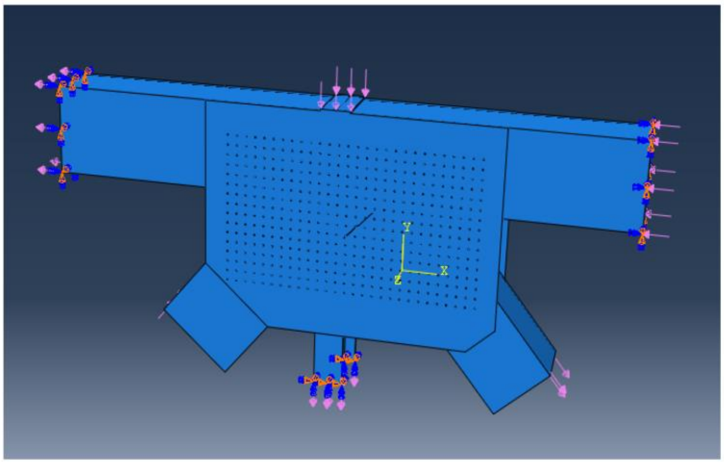


Figure 65. The geometry of the gusset plate

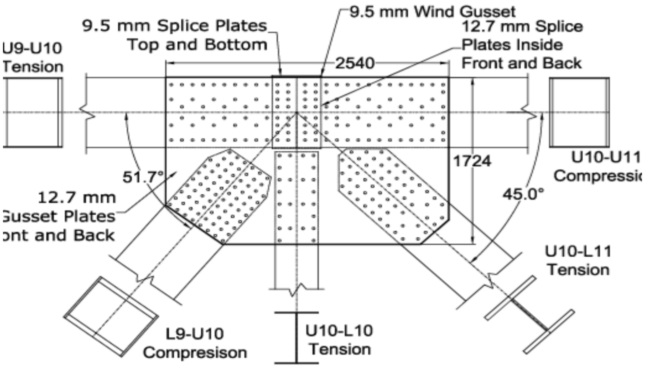


Figure 66. Dimensions of the gusset plate joint (Liao and Okazaki, 2009)

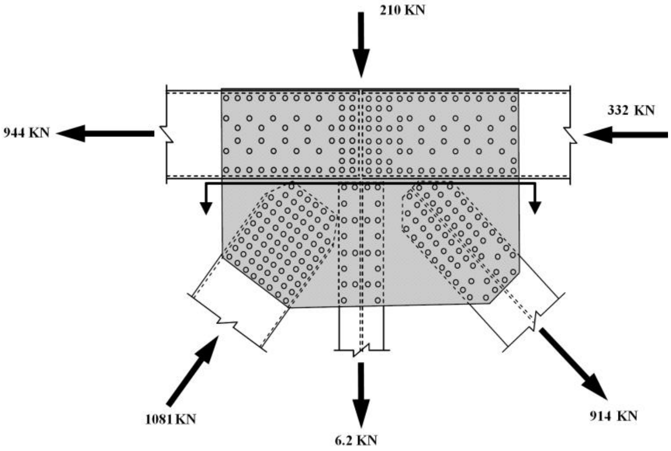
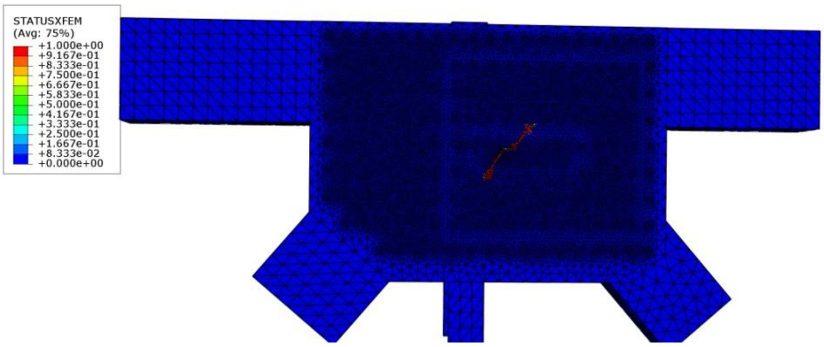
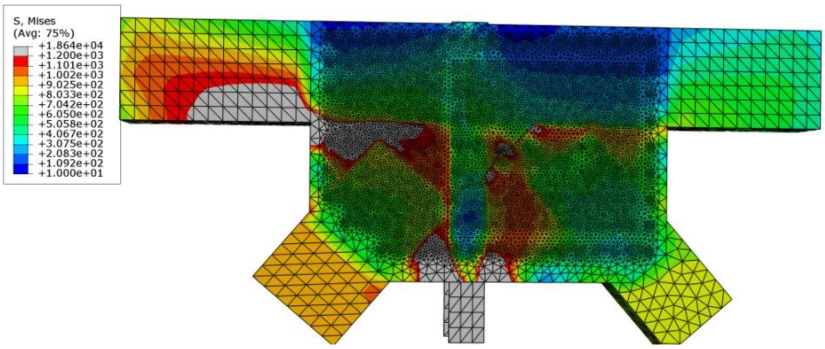


Figure 67. The location and the magnitude of the loading

Data Processing: Similar to Case I, static strain data of the surface nodes extracted from the FE simulations are used for the damage detection. In order to have a more realistic definition of the damage progress, an extended finite element method (XFEM) analysis is first performed. XFEM is an efficient extension to classical FEM to model the propagation of various discontinuities such as cracks (Moes et al., 1999). Herein, an XFEM crack with a small length (10 mm) is created at the middle of the plate and then the load is increased to capture the crack propagation status. Figure 68 illustrates the results of the XFEM analysis for critical loading along with the crack propagation direction.



(a) Crack propagation direction



(b) Von Mises stress distribution for critical loading

Figure 68. The results of the XFEM analysis

Taking into account the final length of the crack at the plate failure (530 mm) and its direction, fifteen damage states (classes) are defined. The damage is simulated by creating notches of different sizes through a new series of FE analyses. Figure 69 presents a schematic definition of different damage states for the gusset plate. The first damaged case belongs to the initial notch with 10 mm length. The other states are created by adding 20 mm to each side of the initial notch on the direction already detected by XFEM. For instance, the third damaged class pertains to the notch with an initial length of 10 mm plus 20 mm on each side, 10 + 20 + 20 = 60 mm. Subsequently, the damage classes can be defined by changing the notch size (a) as follows:

1: Intact beam (a= 0 mm)
2: a = 10 mm (Initial notch)
3: a = 50 mm
4: a = 90 mm
5: a = 130 mm
6: a = 170 mm
7: a = 210 mm
8: a = 250 mm
9: a = 290 mm
10: a = 330 mm
11: a = 370 mm
12: a = 410 mm
13: a = 450 mm
14: a = 490 mm
15: a = 530 mm

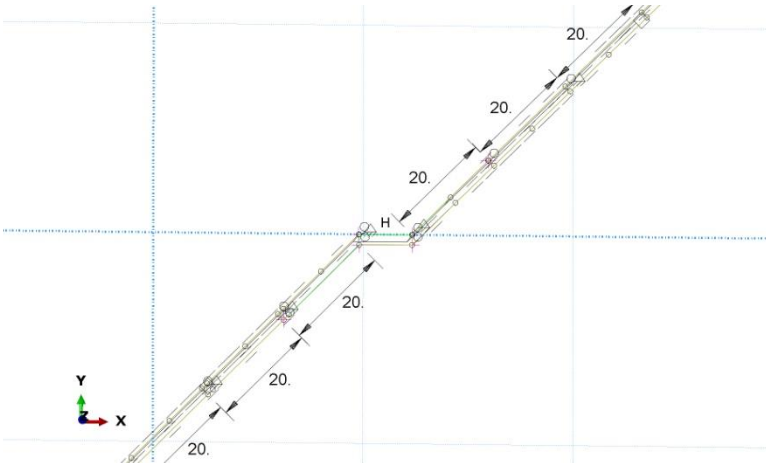
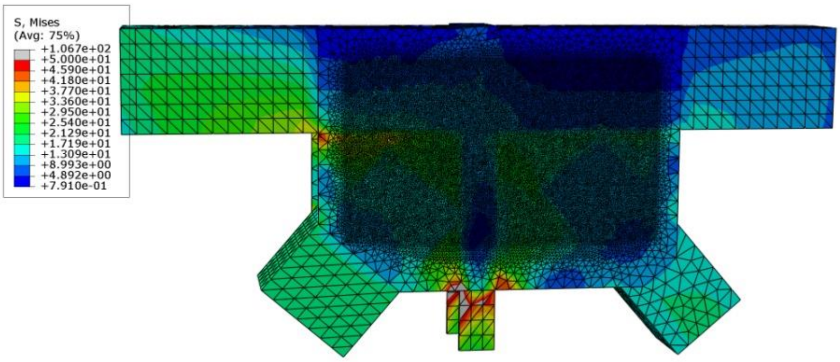
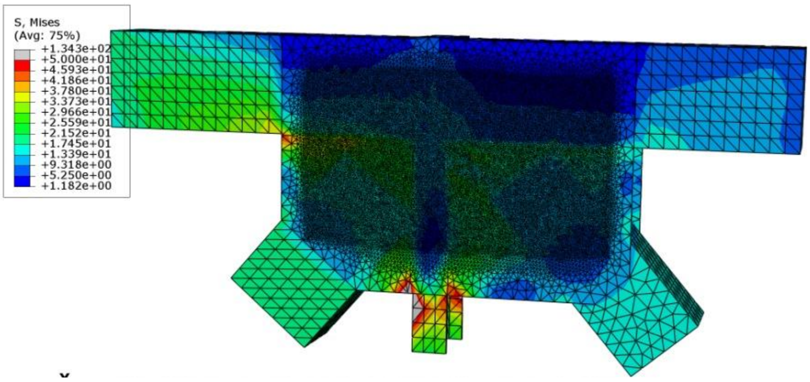


Figure 69. Definition of different damage states for the gusset plate

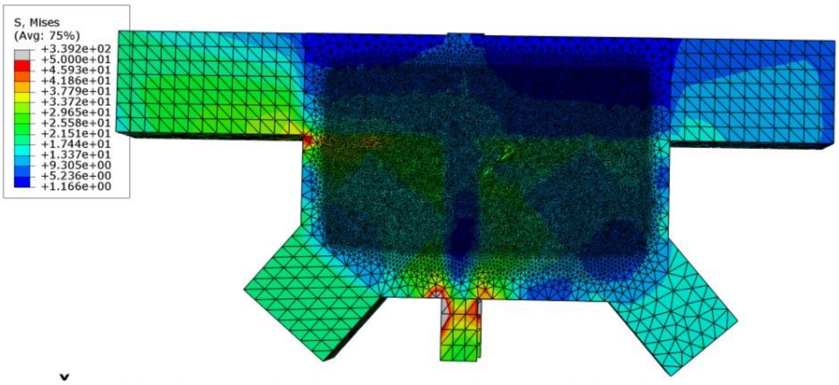
The FE simulation results for some of the damage scenarios (i.e., Classes 1, 7, 9) are shown in Figure 70. Since Case II, is a realistic case, the number of gates (strain levels) is selected according to the piezoelectric properties considered for the design of the smart pebble sensors. The minimum level of strains to be captured by piezoelectric transducers is about 30.00 με. On the other hand, the maximum of the strain value extracted from the FE simulations of fifteen damage scenarios is about 150 με. Considering a reasonable number of ten strain levels for the gates, the difference between the strain levels is 13.33 με. Table 13 presents the preselected strain levels for the gusset plate. For this complicated case, the new MATLAB script is written that takes the strain-time data from ABAQUS and finds the strain information for the considered 448 or any other number of sensors with different sizes (for this case: 10 mm).



(a) Intact plate (Class 1: a = 0 mm)



(b) Damaged plate (Class 7: a = 200 mm)



(c) Damaged plate (Class 9: a = 280 mm)

Figure 70. The FE simulation results for typical damage scenarios for Case II (Von Mises stress distribution)

Table 13. The preselected strain levels considered for the gusset plate failure analysis

Gate number	Strain Level (με)
1	30.00
2	43.33
3	56.67
4	70.00
5	83.33
6	96.67
7	110.00
8	123.33
9	136.67
10	150.00

3.3.3.1.3 Implementation and Simulation Results

As discussed before, each of the nodes and elements of the FE models can be a possible sensor location. In order to perform the damage analysis, different numbers of data acquisition nodes are considered as potential sensors. The first stage is a basic analysis focused on detecting the damage introduced to the simply supported beam. Thereafter, the findings from this important stage are used to detect the damage states defined for the realistic gusset plate case. For the beam case, each of the elements represents a sensor location. For the gusset pate with complex meshing, the diameter for the potential sensors is equal to 10 mm. Although the main goal is to provide good damage detection accuracy, the nature of the simulations provides a chance of evaluating the optimal number of sensors required for a precise damage detection. The performance of the models developed for each sensor configuration is measured using the detection rate (DR) defined as follows:

$$DR = \frac{Number\ of\ Damage\ States\ Correctly\ Classified}{Total\ Number\ of\ Data\ Sets\ in\ Configuration} \tag{30}$$

In the following sections, the details of the PNN algorithm used for the damage state classification are described. Subsequently, comprehensive explanations about the implementation of the method and the simulation results are provided.

Damage Detection for Simply Supported Beam: For the simply supported beam, several analyses are performed with 150, 100, 50, 25, 20, 15, 10 and 5 “randomly selected” elements (sensors). In order to have an insight into the effect of pre-determining of the location of the sensors, two separate scenarios are considered for sensor numbers equal to 10 and 5. For these cases, the sensors are assumed to be located in the mid-span of the beam and above the notch. As seen in Figures 59 (a) to (c), this critical location is notably influenced by the applied load for both the intact and damaged cases. Each of the damage classes is represented by a dummy variable. The variables of 1 to 7 stand for intact beam, a/D = 0.05, a/D = 0.1, a/D = 0.2, a/D = 0.3, a/D = 0.4, and a/D = 0.45, respectively. For the first phase of the analyses, the damage state is considered to be a function of X, Y, μ and σ. For the analysis, the generated data sets for each of the sensor configurations are randomly classified into three subsets: (1) calibration, (2) validation, and (3) test subsets. The calibration set is used to fit the models and the validation set is used to estimate the classification error for model selection (Alavi et al., 2011). For each of the sensor configurations, a number of repetitions with newly generated random locations are considered to guarantee that the models with the best performance on both the calibration and validation phases are derived. Finally, the testing set is employed for the evaluation of the generalization ability of the final chosen models. The calibration, validation and testing data are usually taken as 50-70%, 15-25% and 15-25% of all data, respectively (Alavi et al., 2011). In the present study, 70% of the data vectors are used for the calibration process and about 15% of the data are taken as the validation data. The remaining data sets are used for the testing of the obtained models. Table 14 shows the descriptive statistics of the μ and σ values for the entire elements.

Table 14. Descriptive statistics of the μ and σ values for the beam

Parameter	μ	σ
Mean	1.68E-05	6.00E-06
Median	1.68E-05	5.44E-06
Standard Deviation	7.22E-06	1.19E-05
Sample Variance	5.22E-11	1.42E-10
Kurtosis	4.66E+02	1.75E+02
Skewness	-2.05E+01	1.15E+01
Range	1.86E-04	2.31E-04
Minimum	-1.70E-04	9.73E-07
Maximum	1.97E-05	2.32E-04

Each calibration (training) sample is set as one neuron in the pattern layer of PNN. Thus, the number of neurons in the pattern layer of the PNN models made with 150, 100, 50, 25, 20, 15, 10 and 5 randomly selected elements is 731, 490, 245, 123, 74, 49, and 25, respectively. Neurons in the summation layer correspond to 7 damage patterns. An extensive trial study is performed to select the most relevant input parameters for the PNN model and the smoothing parameter (η). Several runs are conducted to obtain a parameterization of PNN with enough generalization. When presenting new input vectors, each neuron in the summation layer outputs the PDF estimations for each pattern at the test sample point. The pattern class with the largest PDF indicates the damage class of the current test sample. A Gaussian function is used as the window function for the PNN algorithm. The best classification results for different number of sensors are presented in Table 15. Based on the conducted runs, the models are not any sensitive to X and Y. That is to say, μ and σ can be considered as sufficient representatives of the damage progress. However, as can be observed in Table 15, the models provide good estimations of the damage on the calibration data but very poor performance on the validation and testing data. Furthermore, slightly better results are provided by locating the 10 and 5 sensor numbers in the mid-span of the beam compared to the randomly distributed cases.

Table 15. The damage detection performance for the beam using X, Y, μ and σ as the predictor variables

Number of potential sensors	Damage Detection Performance (%) Calibration	Validation	Testing	Optimal smoothing parameter
150	85%	24%	24%	1E-25
100	85%	30%	24%	1E-20
50	88%	21%	33%	1E-20
25	94%	19%	12%	1E-25
15	97%	25%	20%	1E-20
10	98%	0%	20%	1E-20
5	92%	20%	0%	1E-25
10 (Set)^1	100%	9%	20%	1E-15
5 (Set)^2	96%	20%	20%	1E-30

1 and 2: Sensors located in the mid-span of the beam, above the notch

In order to improve the detection performance, the parameters that incorporate the "group effect" are also used for the simulations (i.e., $\mu_D, \sigma_D, \mu_S, \sigma_S$, and Z-functions). After extensive preliminary runs, it is found that the parameters defined in Eqs. (17) and (18) do not provide good estimations when used together. A possible reason is that each set of the input parameters require a specific smoothing parameter. Also, it is revealed that X, Y, μ, and σ even decrease the detection performance when used with other parameters. However, the input parameters for the final analyses are categorized into two different groups: (1) $\mu_D, \sigma_D, \mu_S, \sigma_S$, and (2) Z-functions. Table 16 and 17 present the best classification results for different number of sensors for two different input categories. For a better visualization of the results, the performance of the models on the validation and testing data is separately shown in Figure 71. As can be observed from Tables 15 to 17 and Figure 71, introducing the new set of the input parameters (i.e., $\mu_D, \sigma_D, \mu_S, \sigma_S$, and Z-functions) to the modeling process results in a significant improvement of the damage detection performance. The models using $\mu_D, \sigma_D, \mu_S, \sigma_S$, and Z-functions as the predictor variables have a very good performance both on the calibration data and on the validation and testing data. Expectedly, the precision of the models decreases with decreasing the number of sensors. Moreover, it can be seen that, nearly in all cases, the models built with Z-functions have a better performance than those made usingμ_D, σ_D, μ_S and σ_S as the input parameters. Also, the models developed with the second set of the predictor variables for the cases of 10 and 5 sensor numbers in the mid-span of the beam provide a significantly better performance than those established using only X, Y, μ and σ. Considering the above explanations, it can be concluded that $\mu_D, \sigma_D, \mu_S, \sigma_S,$ and Z-functions contain sufficient information for the detection of the damage state. Besides, Figure 72 presents the detailed classification performance of the derived models on testing data with confusion matrixes. A confusion matrix (Kohavi and Foster, 1998) contains information about actual and predicted classifications. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. As an example, Figure 72(a) presents the confusion matrix for the training data for 150 sensors. The entire testing data for this case is 156 records which is the summation of the all of the upper numbers in green and red boxes of the matrix. Each column shows the number of data sets in a specific class. According to this figure, there are 23, 22, 25, 18, 23, 23, and 19 records available in 1 to 7 classes, respectively. The green boxes show the number of the correctly classified classes and their percentage to the total number of data sets. For instance, for class 1, 23 cases of the total of 23 records are correctly classified (100% accuracy) and the ratio of the 23 cases to the total of the 156 records is 14.7%. For class 7, out of total of 20 cases, 19 cases are correctly classified (95% accuracy) and the ratio of the 19 cases to the total of the 156 records is 12.2%. The summation of the correctly classified percentages in green boxes is equal to the overall classification accuracy of the model (14.7%+14.1%+16%+11.5%+14.7%+14.7%+12.2% = 98.1%).

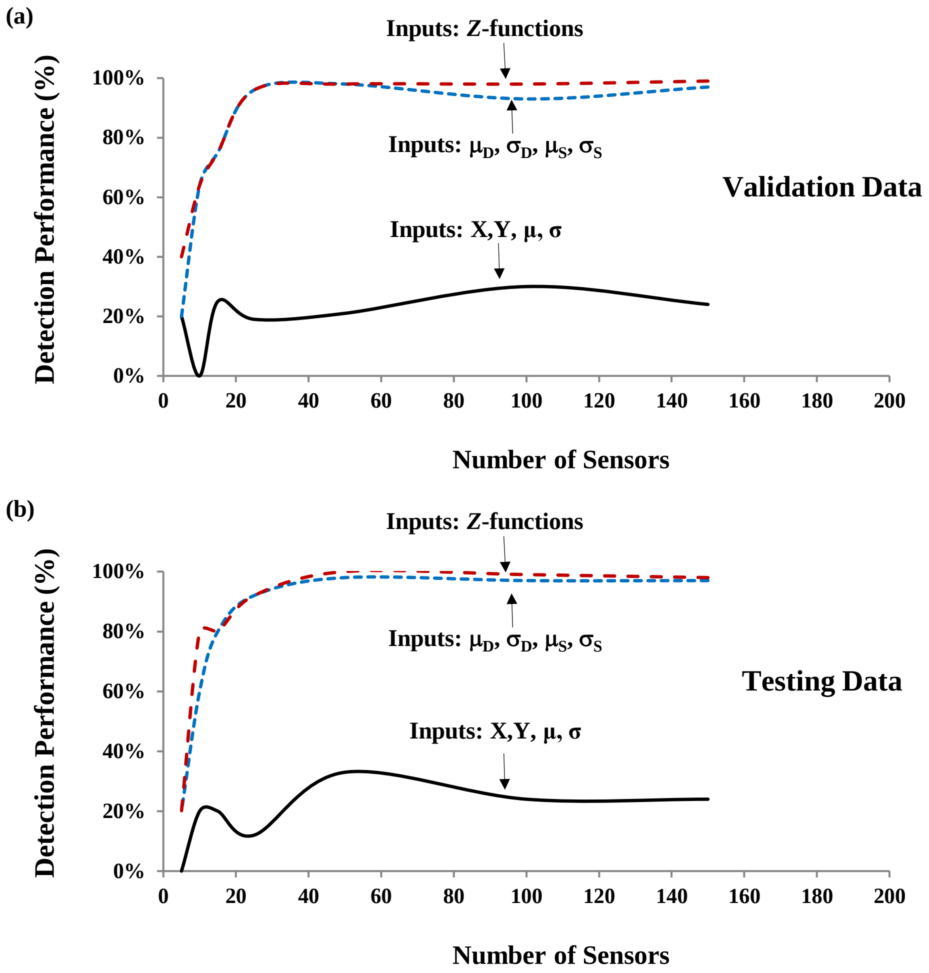


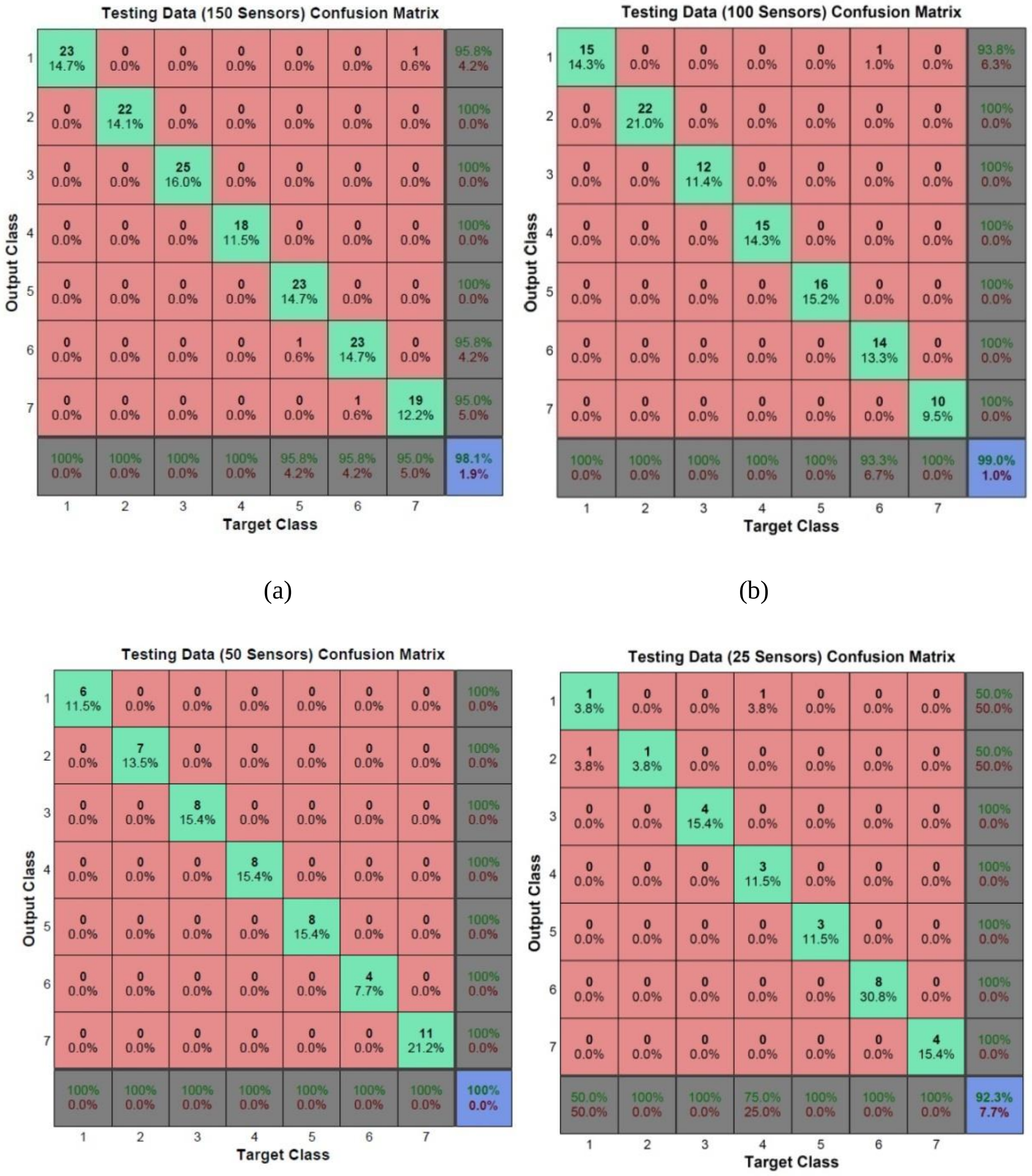
Figure 71. A visual comparison of the performance of the models developed using different sets of input parameters for the beam analysis

Table 16. The damage detection performance for the beam using $\mu_D, \sigma_D, \mu_S, \sigma_S$ as the predictor variables

Number of potential sensors	Damage Detection Performance (%)			Optimal smoothing parameter
	Calibration	Validation	Testing	
150	99%	97%	97%	25E-4
100	100%	93%	97%	1E-4
50	100%	98%	98%	1E-4
25	100%	96%	92%	1E-4
15	96%	75%	80%	1E-4
10	92%	64%	60%	1E-5
5	96%	20%	20%	1E-4
10 (Set)	96%	73%	60%	1E-4
5 (Set)	100%	80%	80%	1E-4

Table 17. The damage detection performance for the beam using the Z-functions as the predictor variables

Number of potential sensors	Damage Detection Performance (%)			Optimal smoothing parameter
	Calibration	Validation	Testing	
150	100%	99%	98%	1E-1
100	100%	98%	99%	1E-1
50	100%	98%	100%	1E-1
25	100%	96%	92%	1E-2
15	97%	75%	80%	1E-4
10	82%	64%	80%	1E-1
5	92%	40%	20%	1E-1
10 (Set)	96%	91%	90%	1E-3
5 (Set)	100%	80%	80%	1E-1



Testing Data (150 Sensors) Confusion Matrix

	1	2	3	4	5	6	7	
1	23 14.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.6%	95.8% 4.2%
2	0 0.0%	22 14.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	25 16.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	18 11.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	23 14.7%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.6%	23 14.7%	0 0.0%	95.8% 4.2%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.6%	19 12.2%	95.0% 5.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	95.8% 4.2%	95.8% 4.2%	95.0% 5.0%	98.1% 1.9%

Testing Data (100 Sensors) Confusion Matrix

	1	2	3	4	5	6	7	
1	15 14.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.0%	0 0.0%	93.8% 6.3%
2	0 0.0%	22 21.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	12 11.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	15 14.3%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 15.2%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 13.3%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 9.5%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	93.3% 6.7%	100% 0.0%	99.0% 1.0%

(a)

(b)

Testing Data (50 Sensors) Confusion Matrix

	1	2	3	4	5	6	7	
1	6 11.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	7 13.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	8 15.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	8 15.4%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 15.4%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 7.7%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	11 21.2%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Testing Data (25 Sensors) Confusion Matrix

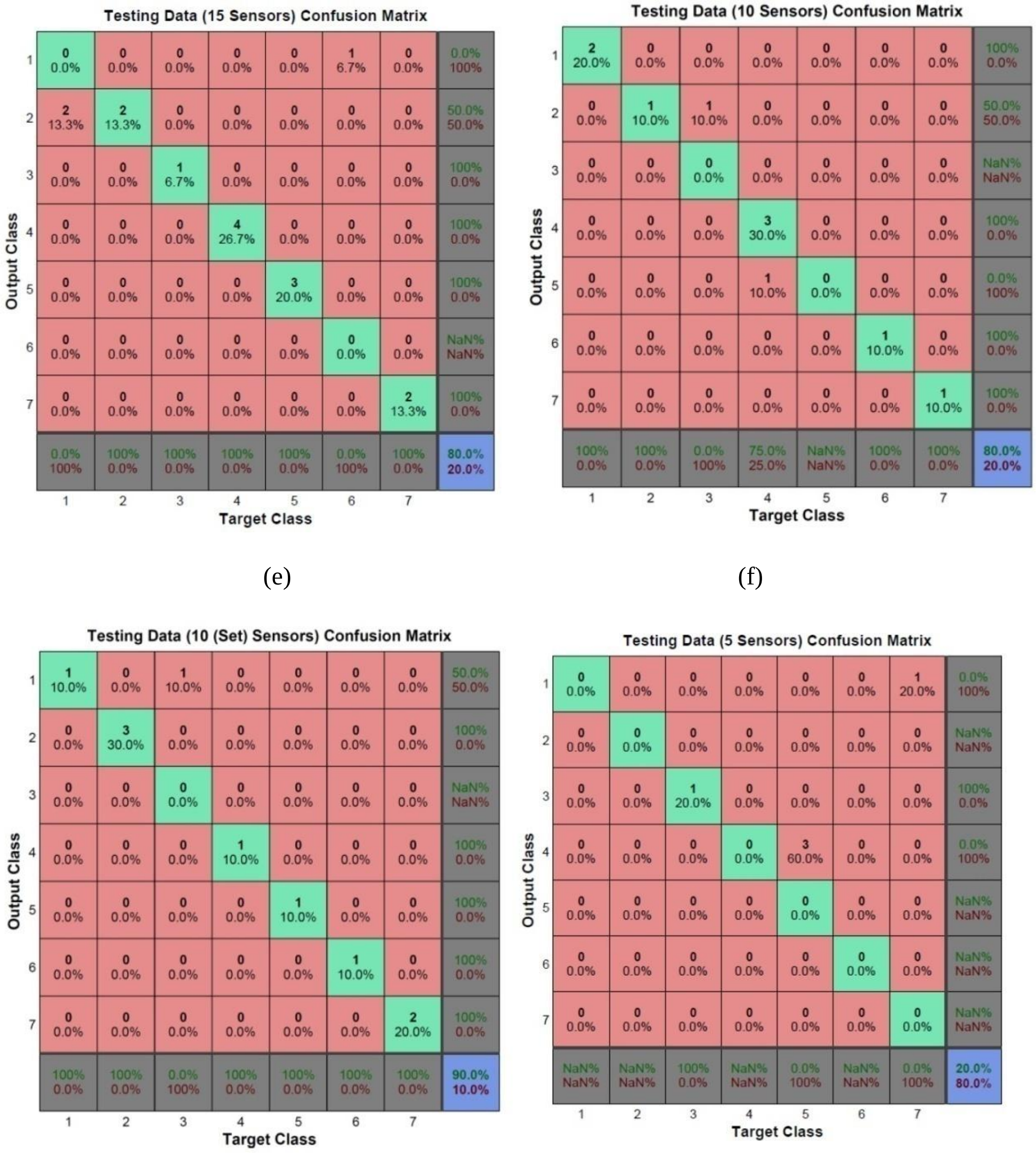
	1	2	3	4	5	6	7	
1	1 3.8%	0 0.0%	0 0.0%	1 3.8%	0 0.0%	0 0.0%	0 0.0%	50.0% 50.0%
2	1 3.8%	1 3.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50.0% 50.0%
3	0 0.0%	0 0.0%	4 15.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	3 11.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 11.5%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 30.8%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 15.4%	100% 0.0%
	50.0% 50.0%	100% 0.0%	100% 0.0%	75.0% 25.0%	100% 0.0%	100% 0.0%	100% 0.0%	92.3% 7.7%

(c)

(d)

Figure 72. Confusion matrixes of the PNN models for the beam plate analysis

Figure 72. (cont'd)



Testing Data (15 Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 6.7%	0 0.0%	0.0% 100%
2	2 13.3%	2 13.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50.0% 50.0%
3	0 0.0%	0 0.0%	1 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	4 26.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 20.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 13.3%	100% 0.0%
	0.0% 100%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	0.0% 100%	100% 0.0%	80.0% 20.0%

Target Class

(e)

Testing Data (10 Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	2 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	1 10.0%	1 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50.0% 50.0%
3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
4	0 0.0%	0 0.0%	0 0.0%	3 30.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	1 10.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 10.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 10.0%	100% 0.0%
	100% 0.0%	100% 0.0%	0.0% 100%	75.0% 25.0%	NaN% NaN%	100% 0.0%	100% 0.0%	80.0% 20.0%

Target Class

(f)

Testing Data (10 (Set) Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	1 10.0%	0 0.0%	1 10.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50.0% 50.0%
2	0 0.0%	3 30.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
4	0 0.0%	0 0.0%	0 0.0%	1 10.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 10.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 10.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 20.0%	100% 0.0%
	100% 0.0%	100% 0.0%	0.0% 100%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	90.0% 10.0%

Target Class

(g)

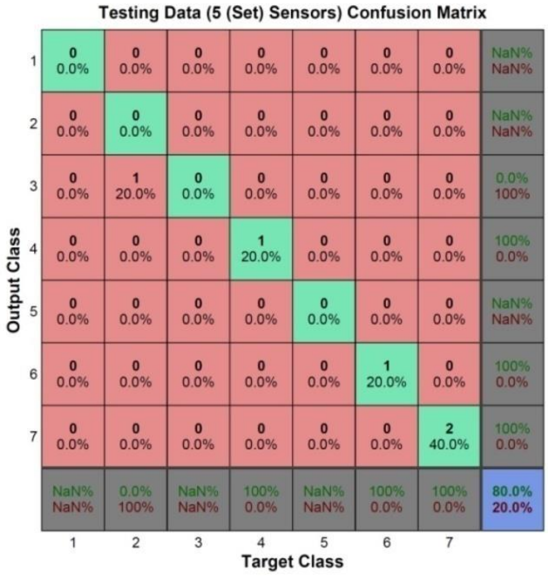
Testing Data (5 Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 20.0%	0.0% 100%
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	0 0.0%	0 0.0%	1 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 60.0%	0 0.0%	0 0.0%	0.0% 100%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	NaN% NaN%	NaN% NaN%	100% 0.0%	NaN% NaN%	0.0% 100%	NaN% NaN%	0.0% 100%	20.0% 80.0%

Target Class

(h)

Figure 72. (cont'd)


Testing Data (5 (Set) Sensors) Confusion Matrix

	1	2	3	4	5	6	7	
1	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	0 0.0%	1 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
4	0 0.0%	0 0.0%	0 0.0%	1 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 20.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 40.0%	100% 0.0%
	NaN% NaN%	0.0% 100%	NaN% NaN%	100% 0.0%	NaN% NaN%	100% 0.0%	100% 0.0%	80.0% 20.0%

(i)

Uncertainty Analysis: As it is known, the model error exists even for a well calibrated FEM model. This error may be caused by the boundary condition, distribution of the structural stiffness, uncertainty of material, etc. Moreover, the real structure might be subjected to environmental effects and operational states such as varying temperature and measurement noise (Yan and Miyamoto, 2003). PNN has noticeable merits in noisy conditions as it describes measurement data in a Bayesian probabilistic approach (Jiang et al., 2011). Despite this capability, adding noise to the data in order to simulate the error of the analytical model and the effect of noise on the actual measurement vectors enhances the reliability of the proposed method. Thus, in this study, the calibration, validation and testing sets are polluted with random noise to simulate the performance of real sensors. To this aim, Gaussian noise with a mean value of zero and standard deviation equal to one standard deviation of the measured data is added to the input vectors. The noise pollution verification phase has been done only for the best models that are developed using the *Z*-functions (see Table 17). Different noise levels are taken for the analysis (5%, 10%, 20%, 30%, and 50%).

The PNN algorithm is run for all the combinations shown in Table 6 with noise-polluted data. Figure 68 visualizes the best classification results for different number of sensors with various noise levels. Comparing the results shown in Tables 17 and Figure 73, it can be observed that increasing the noise level does not influence on the performance of the models on the calibration data. For the validation and testing data, the trends of the results are complicated. In these cases, it can be seen that the identification accuracies of the models gradually decrease with the increase of the noise level. The results for the testing data indicate that nearly all of the models have a good accuracy for noise levels up to 20%. On the other hand, the performance of the models subjected to higher noise levels, in particular 30 and 50%, does not remain satisfactory. This is more evident for the 5 and 10 sensor numbers.

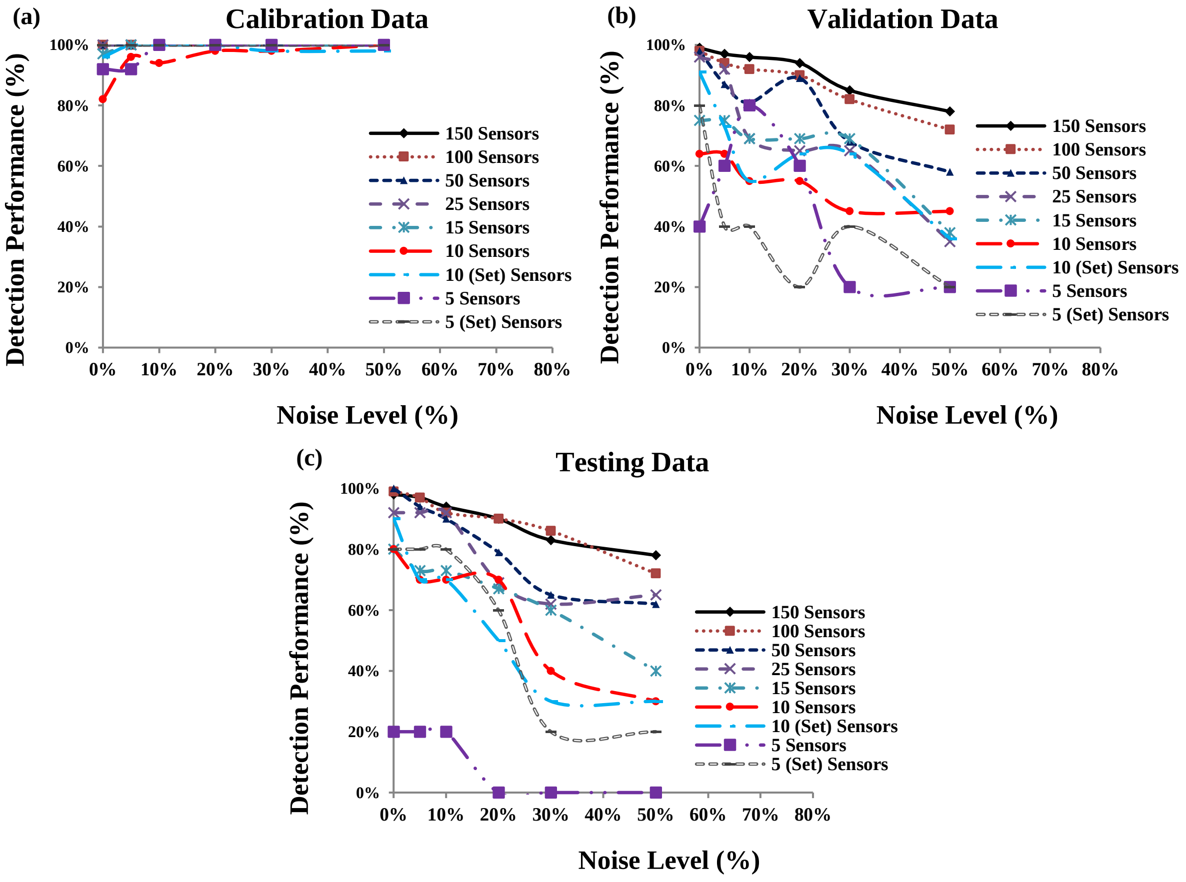
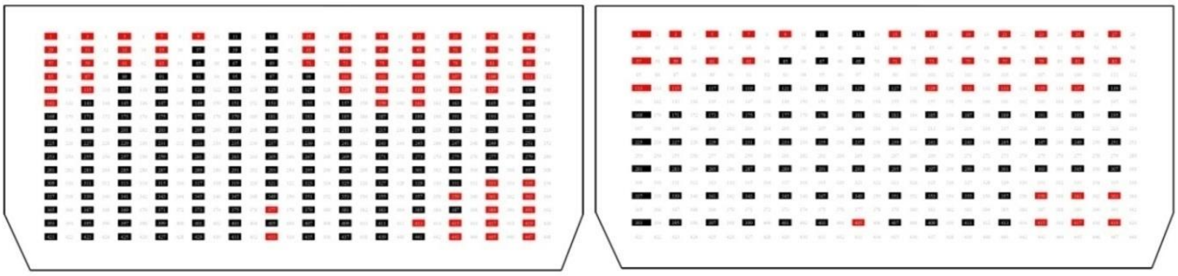
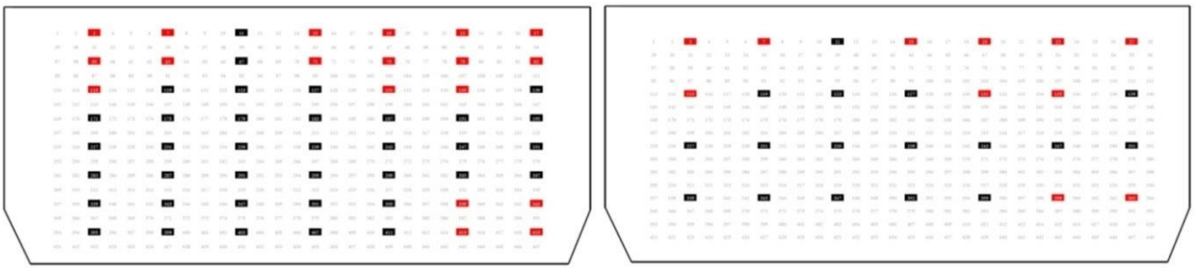


Figure 73. The damage detection accuracy of the best models versus the noise levels for the beam analysis

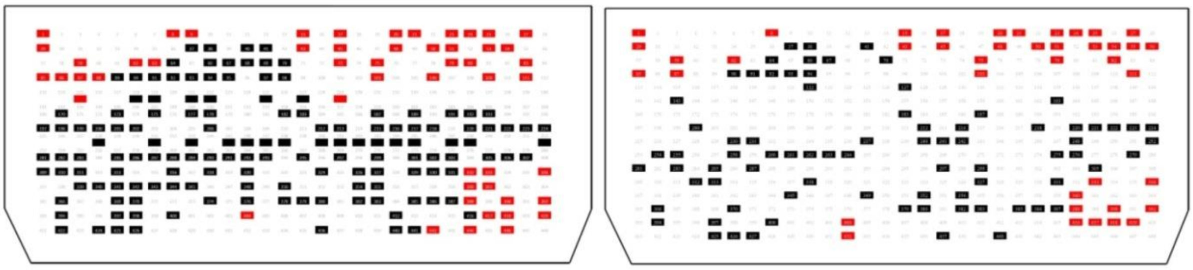
Damage Detection for the Gusset Plate: The observations from the first stage of method verification provide valuable information to tackle the geometrically complicated case of gusset plate. For this case, in addition to the randomize configuration of the sensors, organized positioning of the sensors are also considered. As described in Section 3.2.3.1.2, the information for 448 data acquisition points is extracted for the analysis. However, considering high number of sensors is neither reasonable nor economic in practical applications. Thus, for the second stage of the performance verification, a maximum of 224 sensors is taken into account. The analyses are performed with 224, 112, 56, and 28 organized and randomly selected sensors. Similar to Case I, different pre-determined configuration schemes with 24, 28, 30, 42, 53, 57, and 64 sensors are also considered to assess the importance of locating the sensors at probable critical regions. Figure 74 shows a schematic representation of the sensor configurations. In this figure, black and red cells represent the sensors. The red cells show the sensors that do not sense damage in all of the damage states. Since such sensors do not record any information, they are excluded from the analyses. The black cells define the active sensors. Based on the results obtained in the first stage (Table 15 and Figure 66), it is revealed that X, Y, μ, and σ cannot be solely used for damage detection. Moreover, among different defined input features, *Z*-functions are the most efficient damage indicator variables. Consequently, these four input features are used for detecting the damage states introduced to the gusset plate. The available data for each of the sensor configurations are randomly divided into three calibration (70% of the data), validation (15% of the data), and testing (15% of the data) subsets. The descriptive statistics of the μ and σ values for the entire elements is shown in Table 18. As discussed before, for each of the randomized sensor configurations, a number of repetitions are considered to guarantee that the models with the best performance on both the calibration and validation phases are developed. Evidently, this issue is of great importance for low number of sensors.



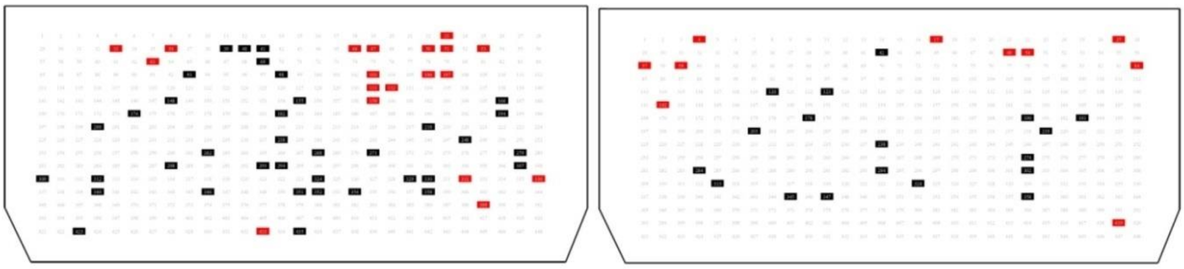
(a) 224 Organized Sensors (156 Active) (b) 112 Organized Sensors (75 Active)



(c) 56 Organized Sensors (37 Active) (d) 28 Organized Sensors (17 Active)



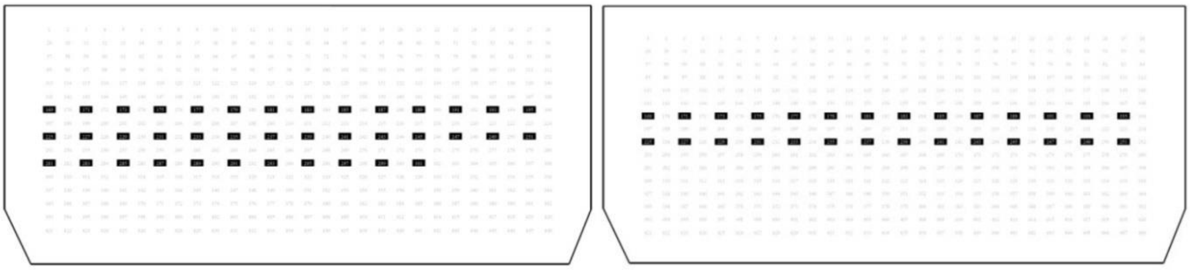
(e) 224 Randomized Sensors (168 Active) (f) 112 Sensors (72 Active)



(g) 56 Randomized Sensors (37 Active) (h) 28 Sensors (17 Active)

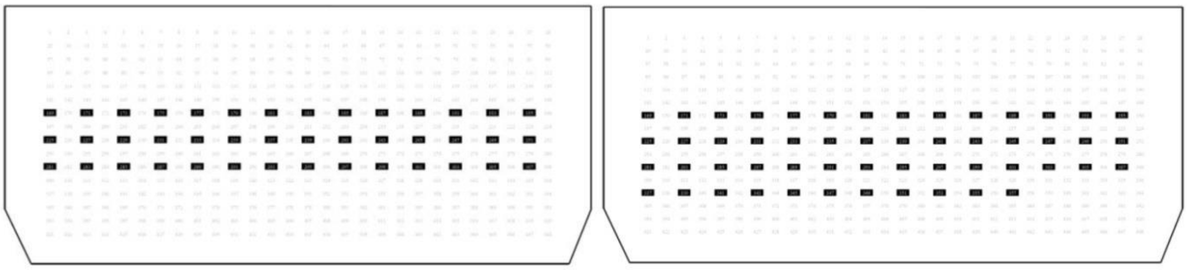
Figure 74. A schematic representation of the sensor configurations for the gusset plate

Figure 74. (cont'd)



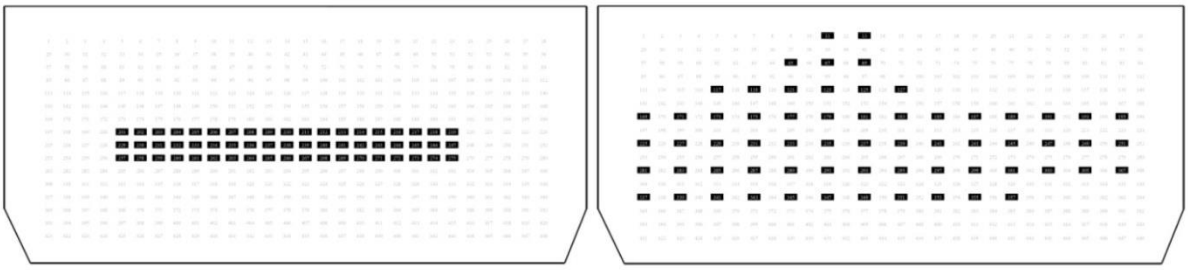
(i) 24 Sensors at Specified Locations

(j) 28 Sensors at Specified Locations



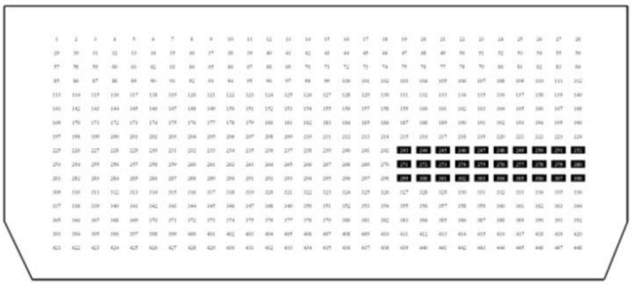
(k) 42 Sensors at Specified Locations

(l) 53 Sensors at Specified Locations



(m) 57 Sensors at Specified Locations

(n) 64 Sensors at Specified Locations



(o) 30 Sensors at Critical Location under the Notch

Table 18. Descriptive statistics of the μ and σ values for the gusset plate

Parameter	μ	σ
Mean	3.28E-03	3.47E-02
Median	3.24E-05	2.30E-05
Standard Deviation	1.90E-01	1.70E-01
Sample Variance	3.63E-02	2.89E-02
Kurtosis	3.69E+01	3.05E+01
Skewness	2.01E-01	5.43E+00
Range	3.12E+00	1.62E+00
Minimum	-2.12E+00	1.74E-08
Maximum	1.00E+00	1.62E+00

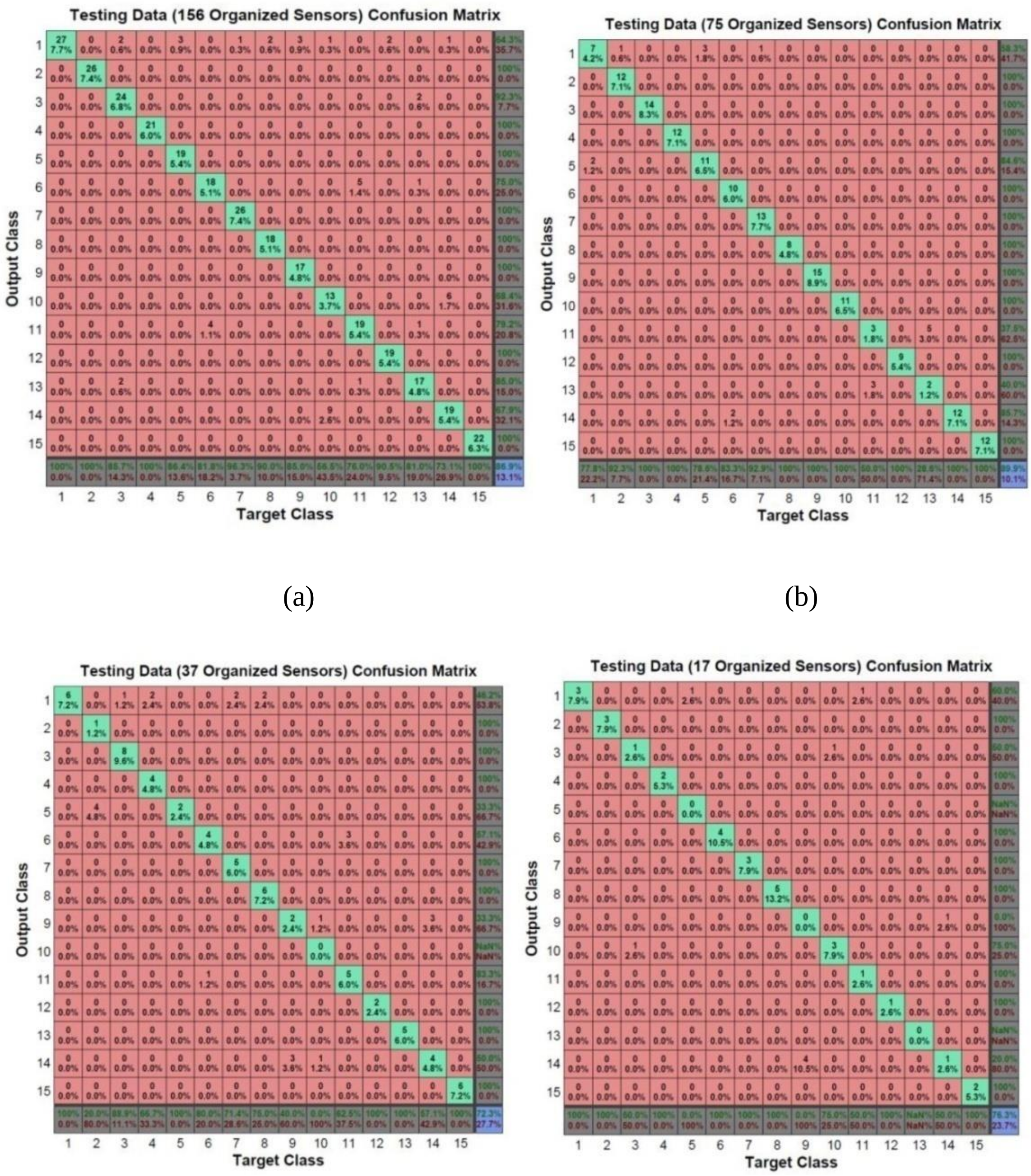
The number of neurons in the pattern layer of the PNN models made with 156, 75, 37, and 17 organized (active) sensors is 1638, 788, 389, and 179, respectively. These numbers for the final models with 168, 72, 37, and 17 randomly selected (active) sensors are 1764, 830, 389, and 179, respectively. Moreover, for the pre-determined configuration schemes with 64, 57, 53, 42, 30, 28, and 24 sensors, the number of neurons in the PNN's pattern layer is 672, 588, 557, 441, 315, 294, and 252, respectively. Neurons in the summation layer correspond to 15 damage patterns. Selection of the η values to find the optimal PNN models is based on extensive trial and error study. Furthermore, the algorithm is run several times to obtain a parameterization of PNN with enough generalization. A Gaussian function is adopted as the window function for the PNN algorithm. The best classification results for different number of sensors are presented in Table 19. In order to have an insight into the area covered by the sensors, the ratio of the area of sensors in each configuration to the area of the plate (Area Ratio) is also shown in Table 19. In order to visualize the detailed classification performance of the derived models on testing data, the corresponding confusion matrixes are given in Figure 75.

As it is seen, the model with 156, 75, 37, and 17 organized sensors have a very good performance on the calibration, validation and testing data. The performance of the models build for 168 and 72 randomly selected sensors is also very satisfying. As expected, the precision of the models decreases with decreasing the number of sensors. However, considering the Area Ratio values, it is seen that even for 168 randomized sensors, only 0.39% of the plate is covered by the sensors (less than 1%). This is indeed a very low rate for achieving such high detection accuracy. On the other hand, for the pre-determined configuration schemes, only 64 and 30 sensors provide acceptable results. This indicates that organized and randomized distribution of the sensors can be a more effective strategy for increasing the detection precision. It is worth mentioning that by decreasing the number of damage classes, the accuracy of the models remarkably increases. For instance, the detection accuracy of the 17 organized sensors for 8 damage classes is equal to 100%, 85% and 90% on the calibration, validation and testing data, respectively. For the 37 active organized sensors, these values are, respectively, equal to 100%, 91% and 86% on the calibration, validation and testing data. Thus, it is possible to reduce the number of sensors and yet have good detection accuracy by decreasing the number of damage classes.

Table 19. The damage detection performance of the PNN method for the gusset plate using the *Z*-functions as the predictor variables

	Number of potential sensors	Damage Detection Performance (%) Calibration	Validation	Testing	Area Ratio	Optimal smoothing parameter
Organized Location	224 (156[*])	100%	87%	87%	0.390%	1E-5
	112 (75)	100%	86%	90%	0.195%	1E-5
	56 (37)	99%	66%	72%	0.097%	1E-4
	28 (17)	100%	68%	76%	0.049%	1E-5
Randomized Location	224 (168)	99%	94%	94%	0.390%	1E-5
	112 (72)	98%	86%	84%	0.195%	1E-5
	56 (37)	99%	46%	53%	0.097%	1E-4
	28 (18)	100%	30%	33%	0.049%	1E-4
Specified Location	24	100%	30%	33%	0.042%	1E-3
	28	100%	37%	29%	0.049%	1E-3
	42	100%	27%	38%	0.073%	1E-4
	53	99%	51%	52%	0.092%	1E-2
	57	99%	51%	52%	0.099%	1E-2
	64	100%	74%	74%	0.111%	1E-2
	30[**]	98%	59%	66%	0.052%	5E-7

* The numbers in the parentheses represent the active sensors for each configuration.
**The sensors are located under the notch.

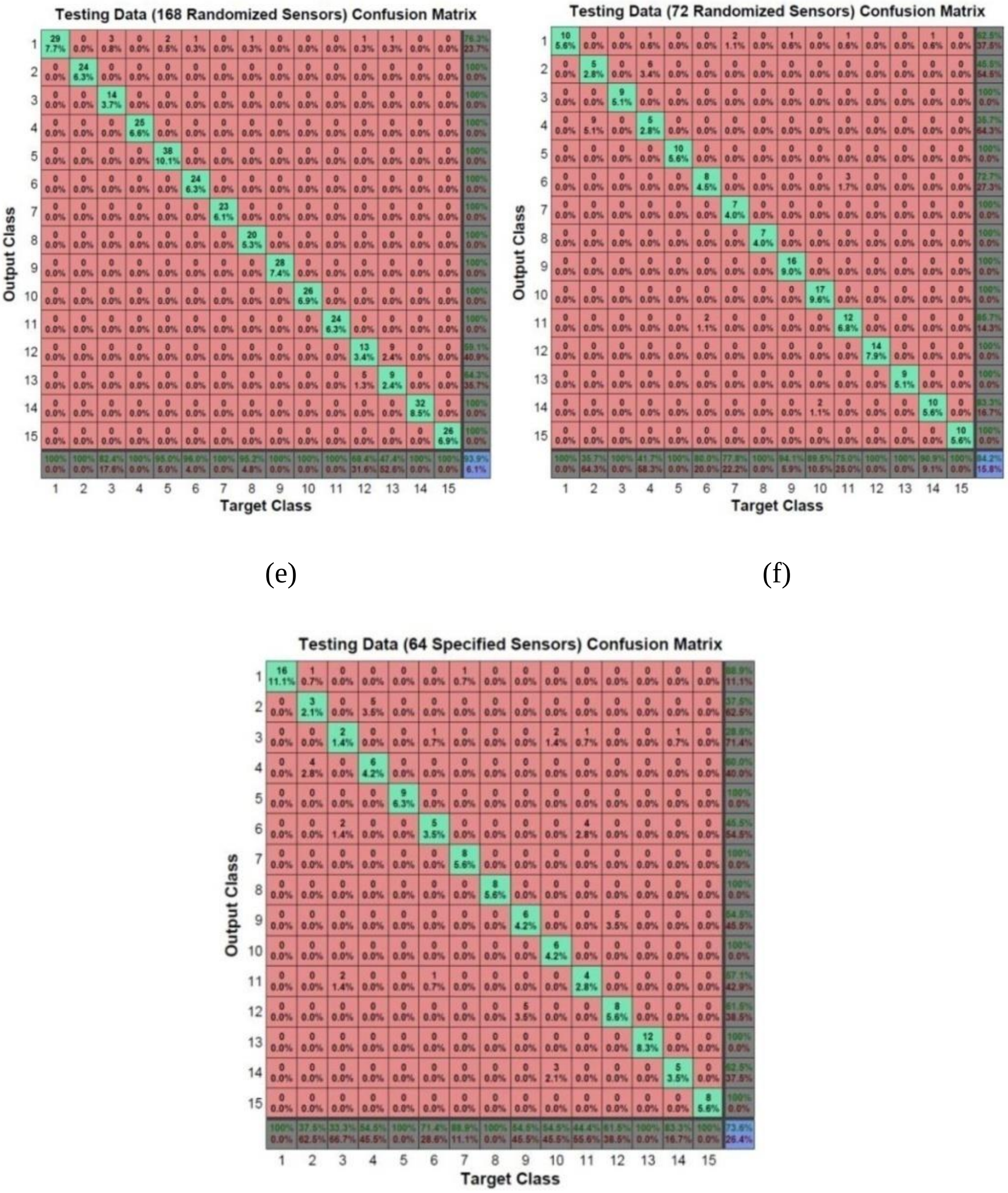


(a) (b)

(c) (d)

Figure 75. Confusion matrixes of the best PNN models for the gusset plate analysis

Figure 75. (cont'd)



(e)

(f)

(g)

Uncertainty Analysis: In order to analyze the uncertainties in predictions, the calibration, validation and testing sets are polluted with random noise. Similar to the case of simply supported beam, Gaussian noise with a mean value of zero and standard deviation equal to one standard deviation of the measured data is added to the input vectors. The noise pollution verification phase has been done only for the best models shown in Table 19. That is, the models for 156, 75, 37, and 17 organized sensors, 168 and 72 randomly selected sensors, and 64 sensors at specified locations are chosen for the uncertainty analysis. Referring to the uncertainty analyses of the beam shown in Figure 68, the performance of the models subjected to 30% and 50% noise levels is not satisfactory. Therefore, for the gusset plate case, only 5%, 10%, 20%, and 30%noise levels are considered. The PNN algorithm is run for all the chosen combinations. Figure 76 presents the best classification results for different number of sensors with various noise levels.

Comparing the results shown in Tables 19 and Figure 71, it is seen that increasing the noise level even up to 30% does not influence the performance of the models build for 168 and 72 randomly selected sensors. Also, the performance of the models for 156 and 75 organized sensors is acceptable up to about 20% noise level. In all cases, it can be seen that the identification accuracies of the models gradually decrease with the increase of the noise level. Besides, the model for 64 sensors at specified locations has an acceptable accuracy only for 10% noise level.

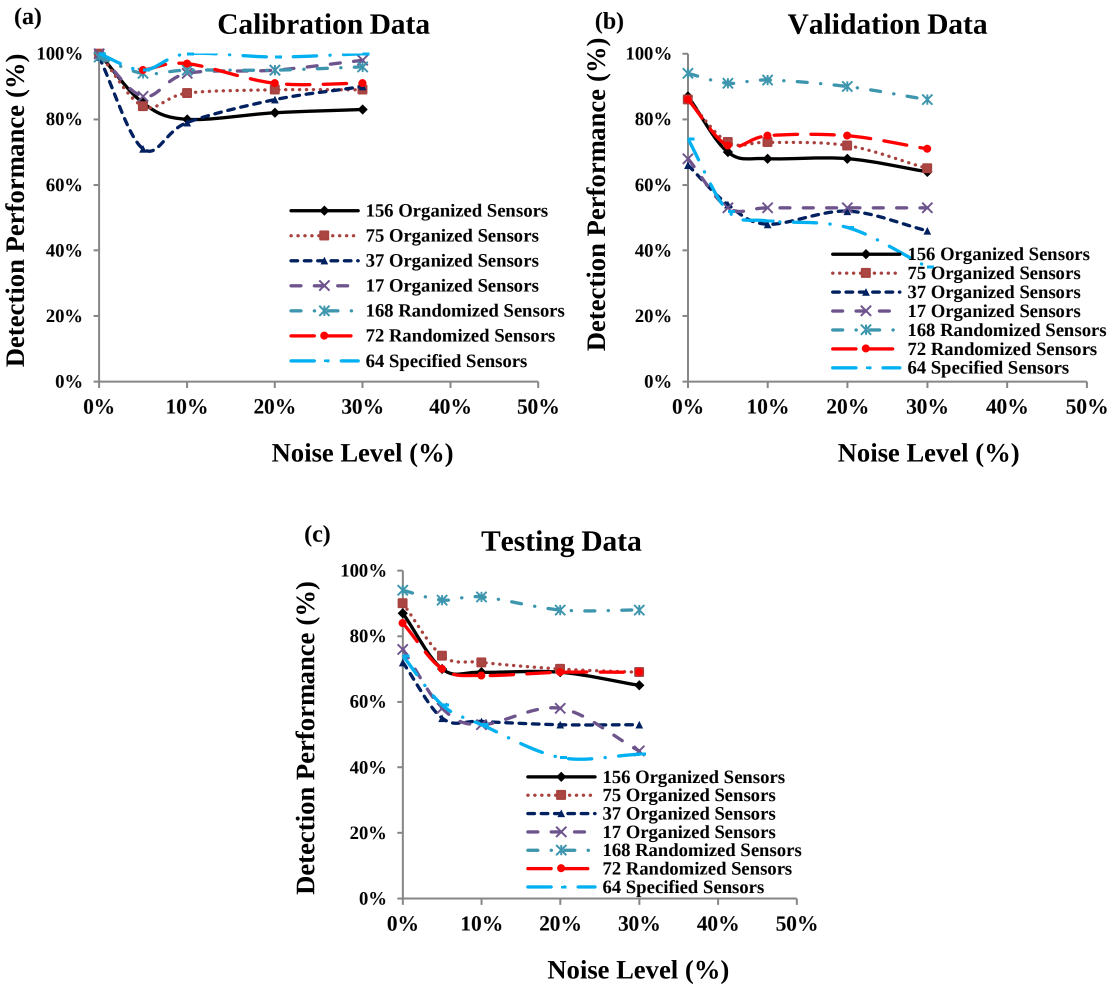


Figure 76. The damage detection accuracy of the best PNN models versus the noise levels for different number of sensors

3.3.3.2 The GPLR Method: Case Study of Bridge Gusset Plate

The 3D FE model used for the evaluation of the PNN method is used in this phase. For this case, the following 7 damage states (classes) are considered:

1: Intact beam (a= 0 mm)

2: a = 10 mm (Initial notch)

3: a = 90 mm

4: a = 170 mm

5: a = 250 mm

6: a = 330 mm

7: a = 410 mm

The same procedure described for PNN is followed to obtain the damage indicator parameters. The generated data sets for each of the sensor configurations are randomly classified into three subsets: (1) calibration, (2) validation, and (3) test subsets. The calibration, validation and testing data are usually taken as 50-70%, 15-25% and 15-25% of all data, respectively. In the present study, 70% of the data vectors are used for the calibration process and about 15% of the data are taken as the validation data. The remaining of the data sets are used for the testing of the obtained models. The GEP parameters are changed for different runs to find the global solution. Various parameters involved in the GEP algorithm are given in Table 20. The parameter setting is based on some previously suggested values (Alavi and Gandomi, 2011) and also after making several preliminary runs and observing the performance behavior. The number of programs in the population is set by the number of chromosomes (population size). The chromosome architectures of the models evolved by GEP include head size and number of genes. The head contains symbols representing both functions and terminals. The head size determines the complexity of each term in the evolved model. In other words, this parameter determines the upper limit for the size of the programs encoded in the gene. The number of terms in the model is determined by the number of genes per chromosome. Each gene codes for a different sub-expression tree or sub-ET. Different optimal levels are considered for the head size and number of genes parameters as tradeoffs between the running time and the complexity of the evolved solutions (Alavi and Gandomi, 2011; Alavi et al., 2013). For the number of genes greater than one, the addition linking function is used to link the mathematical terms encoded in each gene (Alavi and Gandomi, 2011; Alavi et al., 2013). A series of preliminary runs are conducted for high number of sensors to evaluate the importance of the parameters included in Eq. (17) and (18) for damage detection. Similar to PNN, the models developed using X, Y, μ and σ have a very poor performance. Also, it is observed that X, Y, μ, and σ even decrease the detection performance when used with other parameters. The results also indicate that the $\mu_D, \sigma_D, \mu_S, \sigma_S$, and Z-functions parameters do not provide good estimations when used together. Thus, the input parameters for the analyses are categorized into two different groups: (1) $\mu_D, \sigma_D, \mu_S, \sigma_S$, and (2) Z-functions. The results of the preliminary analysis showed that the models built with the Z-functions have a better performance than those made using μ_D, σ_D, μ_S andσ_S as the input parameters. Consequently, for the final analyses, these four input features are used for the detection of the damage states introduced to the gusset plate. Referring to Table 2, there are 3×3×3 = 27 different combinations of the algorithm parameters. A minimum of 3 replications are carried out for each parameter combination. The period of time acceptable for evolution to occur without improvement in best fitness is set via the generations without change parameter. After 2000 generations considered herein, a mass extinction or a neutral gene is automatically added to the model. In this study, basic arithmetic operators and mathematical functions are utilized to get the optimum GEP model. The program is run until there is no longer significant improvement in the performance of the models.

Table 20. Parameter settings for the GEP algorithm

Parameter		Settings
General		
	Chromosomes	30, 50, 100
	Genes	3, 4, 5
	Head Size	10, 15, 20
	Linking Function	Addition
	Function set	+, -, ×, /, √, power, exp
Complexity Increase		
	Generations without Change	2000
	Number of Tries	3
	Max. Complexity	5
Genetic Operators		
	Mutation Rate	0.00138
	Inversion Rate	0.00546

3.3.3.2.1 Implementation and Simulation Results

In the present study, there are 7 damage classes (states). Following the steps of GPLR described in Section 3.2.1, separate GEP models are derived for each class. For 7 damage states and 4 sensor configurations (112, 56, 28, and 8), a total of 28 models are derived. Therefore, the overall number of runs for all the models is at least equal to 27 (number of combinations of the algorithm parameters) × 28 (number of models) = 756. Using Eq. (21), probabilities are assigned to each of the models based on the LR approach. The damage state with higher probability is chosen as the dominant state. On the basis of extensive analyses, the best models for the damage states in each sensor configuration are derived and presented in Tables 21-25. Moreover, the models are provided as Visual Basic for Application (VBA) codes in Appendix. The VBA models can be deployed to Microsoft Excel for easier implementation. Referring to Eq. (21), probabilities (P) to predict each damage state can be calculated using X, α and β given in Tables 21-25. Note that X is a nonlinear combination of the Z-functions in which d0, …, d3 represent $Z_{\mu 2}$, $Z_{\sigma 2}$, $Z_{\mu 1}$, and $Z_{\sigma 1}$, respectively.

Table 21. The derived models for the detection of each damage state using 112 organized sensors (75 active)

Damage State	**Model**
State 1 (Intact, a= 0 mm)	X = ((d2+((2×d0)+ 5.994))-(-7.0×d1)) + (((d0^2)×((d3^3)/d1))+exp(-6.0)) +exp(((d0/d2)^2))^4; α= -9.08512660782129; β= 6.51918988984368E-77;
State 2 (a = 10 mm)	X = ((((-7.91^4)+(d3^2))×(-7.91+d3))×exp(d1)) + (((((-8.0^4)× -10.0)^4)-((-10.0×2.7)-(-8.0×d1)))^2) + ((d0^4)×exp(((((-2.97^4)×d1)×(d3×-2.97)))); α = -35642828.6530698; β = 4.49875667638908E-30;
State 3 (a = 90 mm)	X = (d0/(d2-exp(((d3-d2)-exp(d2))))) + ((((d0+d0)+d0)+(d0+3.0))/((d3/d2)×(d2/-3.0)))+ (d0/(d2-((exp(d2)-d3)-(d2/d1)))); α = -12.159410383184; β = 5.51268762967603E-02;
State 4 (a = 170 mm)	X = ((d1+d0)+(((d3+d0)+d3)×-9.0))+ ((((d0+d2)×d0)+2)/((d3-d0)+(d3^2)))+((((d0×d3)+d2)+((d3-d0)-4))+3); α = -8.03210005727878; β = 7.56920021002415E-02;
State 5 (a = 250 mm)	X = ((((d0/3)×d1)×(d0+3))/((d3^2)×9)) + ((((d1+d2)-d3)×(d3^4))/((d0^3)×(d1^3)))+((-2-d2)^2); α = -11.1545771888252; β = 0.21321778570173;
State 6 (a = 330 mm)	X = ((-8.8^4)×exp((√((d3^4))- √(10.89)))) + (exp(10.989)×((exp(10.989)^4)×((d0^4)× 6.993))) + exp(((((d2^4)-10.01)×(-0.99-d3))×9)); α = -8.6482095125135; β = 1.09885278840212E-32;
State 7 (a = 410 mm)	X = (((exp(d1)+exp(d1))^3)×((d1+5.5055)×(d1+d0)))+((((d1-4.4955)×exp(d1))^3)×((6.3-d2)-(4.4955-d0))) + (exp(((((d1+d0)^4)×(0.986-d3))-d1))^4); α = -8.49403032230808; β = 1.13128375340451E-87;

Table 22. The derived models for the detection of each damage state using 56 organized sensors (37 active)

Damage State	**Model**
State 1 (Intact, a= 0 mm)	X = (((((((-1-d1)-d0)^2)+d2)^3)^3)^4)+d3+d2-6; α = -10.6305379773197; β = 5.5477404146232E-40;
State 2 (a = 10 mm)	X = ((1.55062E+36)-(-7-d1)) + ((((d1×4.95)×(d1^4))^3)/((4.95×4.95)-(d1+8))) + exp((((0.999/d0)×(-6.534×10.10))×((0.999×5.99)×(d3×0.999)))); α = 87499680.6969639; β = 6.15197451294272E-29;
State 3 (a = 90 mm)	X = ((((d0×d2)^2)×(-5.05-d3))+((-5.05-1.8)-(d3+d3))) + exp(((8.18/d2)×((d0-d2)+(3.92×d0)))) + (-1+d0); α = -8.50691174396521; β = 4.99843841351498E-131;
State 4 (a = 170 mm)	X = exp((((d3×d3)+(-6+d1))×(-11.03-(d3-d2))))+ (((d2+((d2-d3)+d1))×((d1+d2)^4))^4) + (((d0+(d2×d2))×((d1+d2)^4))^4); α = -8.99890768675053; β = 6.66222120699797E-33;
State 5 (a = 250 mm)	X = ((((36+2d3)+(d3/d2))^3)^2); + ((d0^3)×((((-d2)-d0)+(d3×-4.0))^3)) + ((((d0^2)×d0)×(d0×d2))+((d1×d2)×exp(6))); α = -10.8370323591921; β = 1.15295630018606E-09;
State 6 (a = 330 mm)	X = -27 + exp((((-440.44-d0)×(d0×d2))-(-74.50×(d2×-5.62)))) + ((((-7×9.09)-d0)×(d2×-7))×((-9.98×-7)×(d0×d1))); α = -8.41137317801545; β = 4.3817590910058E-08;
State 7 (a = 410 mm)	X = 4+d1 + (exp(((-20.08+d0)/(d1-8.1)))×((d1×-10.90)×(d2-d1))) + exp(((exp(d1)× 118.81)×(-10.90×d3-d0^2))); α = -8.43036650035257; β = 9.61588479109158E-138;

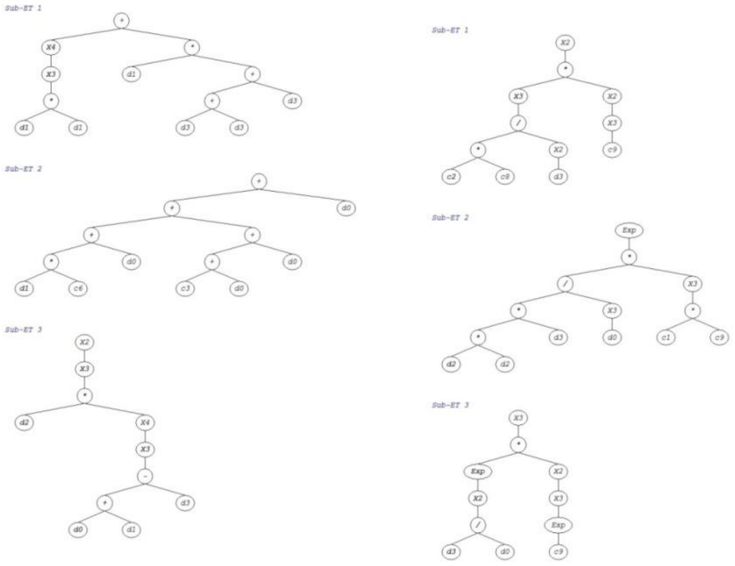
Table 23. The derived models for the detection of each damage state using 28 organized sensors (17 active)

Damage State	**Model**
State 1 (Intact, a= 0 mm)	X = d1^24+d1×3d3 + ((((d1×9.09)+d0)-2+2d0)+d0) + (((d2×((((d0+d1)-d3)^3)^4))^3)^2); α = -9.22728932274076; β = 1.43721533852321E-30;
State 2 (a = 10 mm)	X = (((((-9.989×= 5.988)/(d3^2))^3)×((673.87^3)^2))^2)+ exp(((((d2×d2)×d3)/(d0^3))× 5795.161))+ ((exp(((d3/d0)^2))× 66852549.607)^3); α = -8.17014727523391; β = 9.92492065443564E-54;
State 3 (a = 90 mm)	X = -8.982 + √(exp((((d0/d2)+8)/((d2/d0)^2)))) + (((d1-3)+(d0-6))/64); α = -7.93832502139603; β = 9.93025946235413E-79;
State 4 (a = 170 mm)	X = exp(((d1×exp(d3))×(-5×(d1+5)))) + (((d1+d0)× 729)×((d1+6)+(d0×d1)))+ ((((-7.007^2)×d2)×(2+d2))×((d3-d0)×(-6+d2))); α = -7.96083513694224; β = 2.34283961720364E-06;
State 5 (a = 250 mm)	X = ((-8×((2.002/d1)×d3))×((-8-d0)×(7/d2)))+ ((((d1/d1)-d0)×(d0-d2))/((d3^2)×(d3/9.99)))+ (((((-2/d1)- 7.07)^2)/(d3+(d1+d0)))^2); α = -7.83849444890817; β = 3.72391104592556E-05;
State 6 (a = 330 mm)	X = ((√ ((d3+10))+(d1+d1))-((d2+5.005)×2))+ (exp(((1-d2)-(d0×d3)))+exp(((8.991×8.991)/d0)))+ ((((d1+9)-6)×(-3/d1))+6); α = -8.41298424276477; β = 1.06766764303909E-28;
State 7 (a = 410 mm)	X = ((((84.6)^3)^4)-(((d0^4)+d0)×(d3^4))) + (((exp(d0)×d0)+((3.641+d0)/(d2^4)))^3)+ (d3-exp((((d2^3)+9)-(d1-8.189)))); α = -7.06653202693616; β = 2.5511686740335E-26;

Table 24. The derived models for the detection of each damage state using 8 organized sensors (8 active)

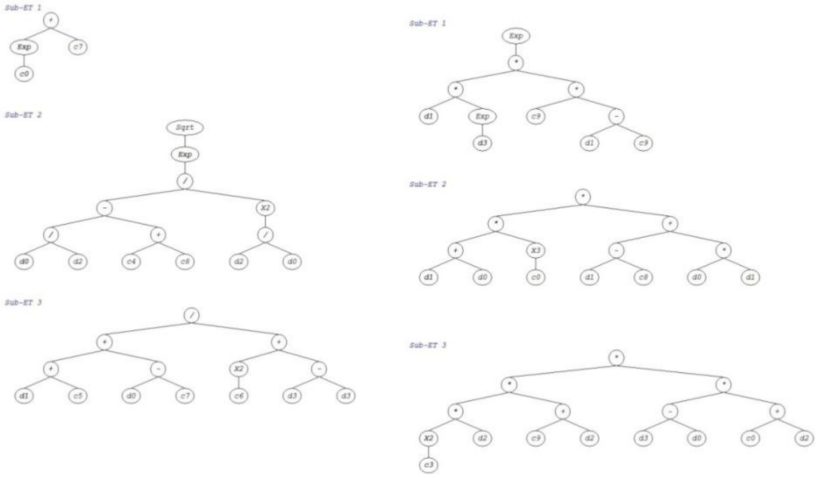
Damage State	**Model**
State 1 (Intact, a= 0 mm)	X =(d1/((exp((d3-d1))/((d2-0.064)+3.988))+((d0/d1)×(d1-d3))))+exp((d1/((((d1-1.0759)^3)× -4.835)×((d3-exp(-7.313))/-14.043))))+exp((d2/(d1-(d3×(((((d2/d1)/ 0.749)+(2.326×d0))×d3)+1))))); α = -8.04669558957269; β = 9.03489419561824E-02;
State 2 (a = 10 mm)	X = ((((((d3^4)^2)×(9.018/d1))+ 164.104)-(-6.066×(-223.207/(d3+d2))))^4)+ (((((d1-d0)+(d0×-7.92))× 9.791)×(((d2+d2)-d1)^3))×(exp((d2-d0))×((9.880^3)^4)))+((((exp(((d2×d2)+(d1/d3)))×((d2+d3)^2))-exp(((d1^3)-(0.092/d1))))^4)×d0); α = -5.34814047260968; β = 1.83957943716362E-16;
State 3 (a = 90 mm)	X = (d1/((exp(exp(d1))/(-9.009-(d3×1.112)))×(((-9.009-d0)-(d3+d1))-(-9.009-d1))))+ (d0×d2)+ (d2-(5.044-((exp((1.998/(d0+d1)))-d2)/(exp((d0+4))^3)))); α = -4.90858192971192; β = 1.29797282256239;
State 4 (a = 170 mm)	X = ((((((d2×d1)-d0)-(d3×d1))+((d0×d0)-(d1×d1)))-d1)+((((-4.035-9.09)^4)^2)^3))+ (d3+(((((((d1-10)-(2.02^4))×((d3^4)- 10))^3)-100)^4)/d1)) + (((((4^4)^4)^4)+(((8.08--27)- 8)×d0))+(((8.08×8)+8)-exp((d0×8.712)))); α = -102965953.285512; β = 3.0258972630069E-31;
State 5 (a = 250 mm)	X = (((d0+d1)+(d1/d0))×(d3-8))+ (((d1+exp(((d0× -6)×(-6/d2))))^3)+((2.236-((d1+d3)×(d0^2)))^2))+ ((((((4^3)×(4-d0))×((4×d2)+(d1+5)))×(4/d3))^4)+d1); α = -8.65825786799137; β = 3.79189016352223E-33;
State 6 (a = 330 mm)	X = (((((((d0-d0)+(d2^4))-d0)×(((2.969+d2)^3)^4))^2)^2)^2)+((((((((d1/d2)^4)-((d2×d0)×d2))^2)/(((d0×d3)+(-3.996×d0))^4))^2)^2)^2)+ ((((exp(d2)- 5.1005)/(3.999-d0))+((d0×d3)^3))+(((d0^2)×exp(d2))+d3^2)); α = -5.53401307008457; β = 1.84854638162239E-48;
State 7 (a = 410 mm)	X = ((-7×((((-8×10.110)× -8)+((4^4)^2))-(exp((d1+d0))×((4^3)×d2))))^3)+exp((((((d0^3)+(d1+d3))×exp(d0))^2)+exp((((3+d1)+(d1+d0))-d3))))+ ((1000^3)+(((((d3^4)×-49)× 8.8)^3)-((exp((8.8-d2))+(1+d0))^2))); α = 55921.7314988659; β = 5.6248130197281E-13;

The derived models present a complex arrangement of operators, variables, and constants that are used to detect damage. As an example, the expression trees (ETs) of the obtained models for one of the sensor configurations (28 sensor layout) are given in Figure 77. In these figures, d0, …, d3 represent $Z_{\mu 2}$, $Z_{\sigma 2}$, $Z_{\mu 1}$, and $Z_{\sigma 1}$, respectively. Also, Xn in the ETs is power to the n. As shown in these figures, the proposed models can be separated into three independent components (subprograms or genes) linked by an addition function. Therefore, each of the evolved subprograms contains important information about the physiology of the final model. Each gene expressed in the final equation is responsible for resolving a particular facet of the problem. Such information provides an opportunity for further scientific discussion at genetic and chromosomal level (Ferreira, 2001).



(a) Intact

(b) Damage State 2

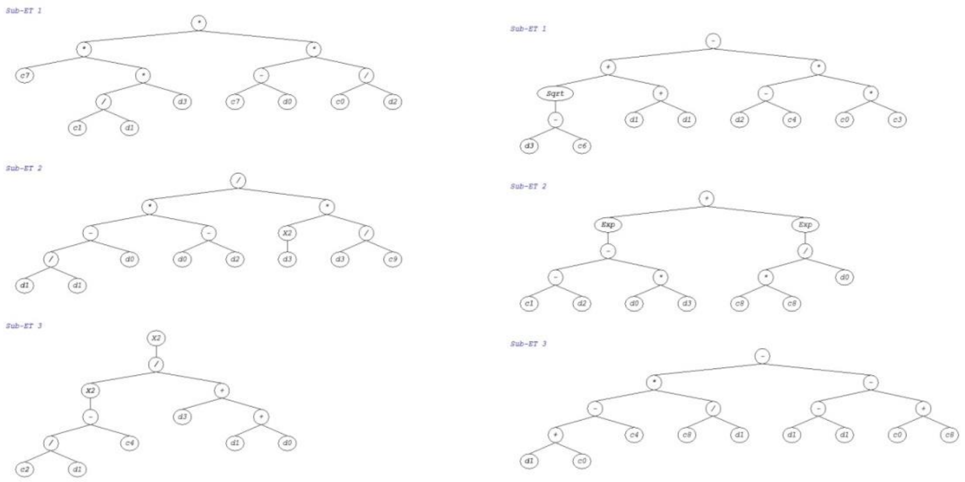


(c) Damage State 3

(d) Damage State 4

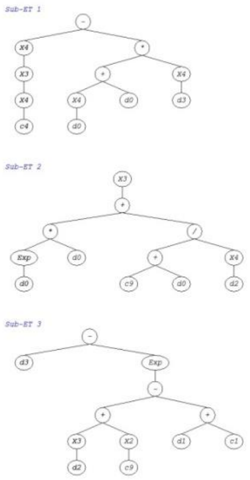
Figure 77. Expression trees of the best models for the detection of each damage state using 28 organized sensors (17 active) (ET = ∑Sub-ETi).

Figure 77. (cont'd)



(e) Damage State 5

(f) Damage State 6



(g) Damage State 7

3.3.3.2.2 Design Example

An illustrative design example is provided to further explain the implementation of the proposed approach for detecting damage. To this aim, the 28 sensor configuration is considered and the goal is to detect damage using one of the sensors (e.g. Sensor # 355). For this case, only 17 sensors sense damage including Sensor # 355. The values of μ and σ for the active sensors are known and the damage state is required. Note that the μ and σ for each sensor configuration are determined following the procedure described in Chapter II, and through curve fitting of the cumulative time histograms. The detection procedure can be divided into the following steps:

Step 1) Calculation of μ_{ave} and σ_{ave}, which are equal to the average of μ and σ of all active 17 sensors, respectively.

Step 2) Calculation of μ_{STD} and $\sigma_{STD}, whic$h are equal to the standard deviation of μ and σ of all active 17 sensors, respectively.

Step 3) Calculation of $Z_{\mu1}, Z_{\sigma1}, Z_{\mu2},$ and $Z_{\sigma2}$which are, respectively, equal to $\frac{\mu_{\#\,355}-\mathrm{\mu_{ave}}}{\mathrm{\mu_{STD}}}$, $\frac{\sigma_{\#\,355}-\sigma_{\mathrm{ave}}}{\sigma_{\mathrm{STD}}}$, $\frac{\mu_{\#\,355}-\mathrm{\mu_{ave}}}{\sigma_{\mathrm{ave}}}$, and $\frac{\sigma_{\#\,355}-\mathrm{\mu_{ave}}}{\sigma_{\mathrm{ave}}}$for Sensor # 355.$\mu_{\#\,355}$ and$\sigma_{\#\,355}$ represent the μ and σ values obtained for Sensor # 355.

Herein, the values of $\mu_{\#\,355,}\sigma_{\#\,355}, Z_{\mu1}, Z_{\sigma1}, Z_{\mu2},$ and $Z_{\sigma2}$ for the given sensor are equal to 36.1με, 14.3με, -999493μ, -999863μ, -242504μ, and -242567μ, respectively.

Step 4) Obtaining the probability of each damage model for the given data.

The models for each damage state in 28 sensor configuration are presented in Table 24. Referring to this table, the probability of each state can be easily estimated. By substituting the $Z_{\mu2}$, $Z_{\sigma2}$, $Z_{\mu1}$, and $Z_{\sigma1}$values for Sensor # 355 into these equations, the probability of each damage state is obtained as follows:

- Probability of State 1 (Intact, a= 0 mm)) = P_1= 0.010%
- Probability of State 2 (a = 10 mm) = P_2= 0.036%
- Probability of State 3 (a = 90 mm) = P_3= 0.031%
- Probability of State 4 (a = 170 mm) = P_4= 99.934%
- Probability of State 5 (a = 250 mm) = P_5= 0.040%
- Probability of State 6 (a = 330 mm) = P_6= 0.022%
- Probability of State 7 (a = 410 mm) = P_7= 0.086%

As it is, the damage state with higher probability is the dominant state. In this case, damage state 4 has the highest probability (P_4= 99.934%) and therefore, the predicted class is 4 for the given sensor information. This example is taken from the testing data and the real state is also damage state 4. Thus, the prediction is in a good agreement with the real damage state of the gusset plate.

3.3.3.2.3 Damage Detection for the Gusset Plate

The performance of the damage models on the calibration, validation and testing data is evaluated following the procedure described in Section 3.2.3.2.1. To facilitate the process, the 7 VBA codes given in Appendix for each sensor configuration are deployed to Microsoft Excel. This provides the possibility of evaluating the damage class of a particular record by applying the argmax (argument of the maximum) function to the probability outputs of all 7 models in each configuration. However, the damage detection rates (DR) for different number of sensors are presented in Table 25. In order to have an insight into the area covered by the sensors, the ratio of the area of sensors in each configuration to the area of the plate (Area Ratio) is also shown in Table 25. In order to visualize the detailed classification performance of the derived models, the corresponding confusion matrixes for the testing data is given in Figure 78.

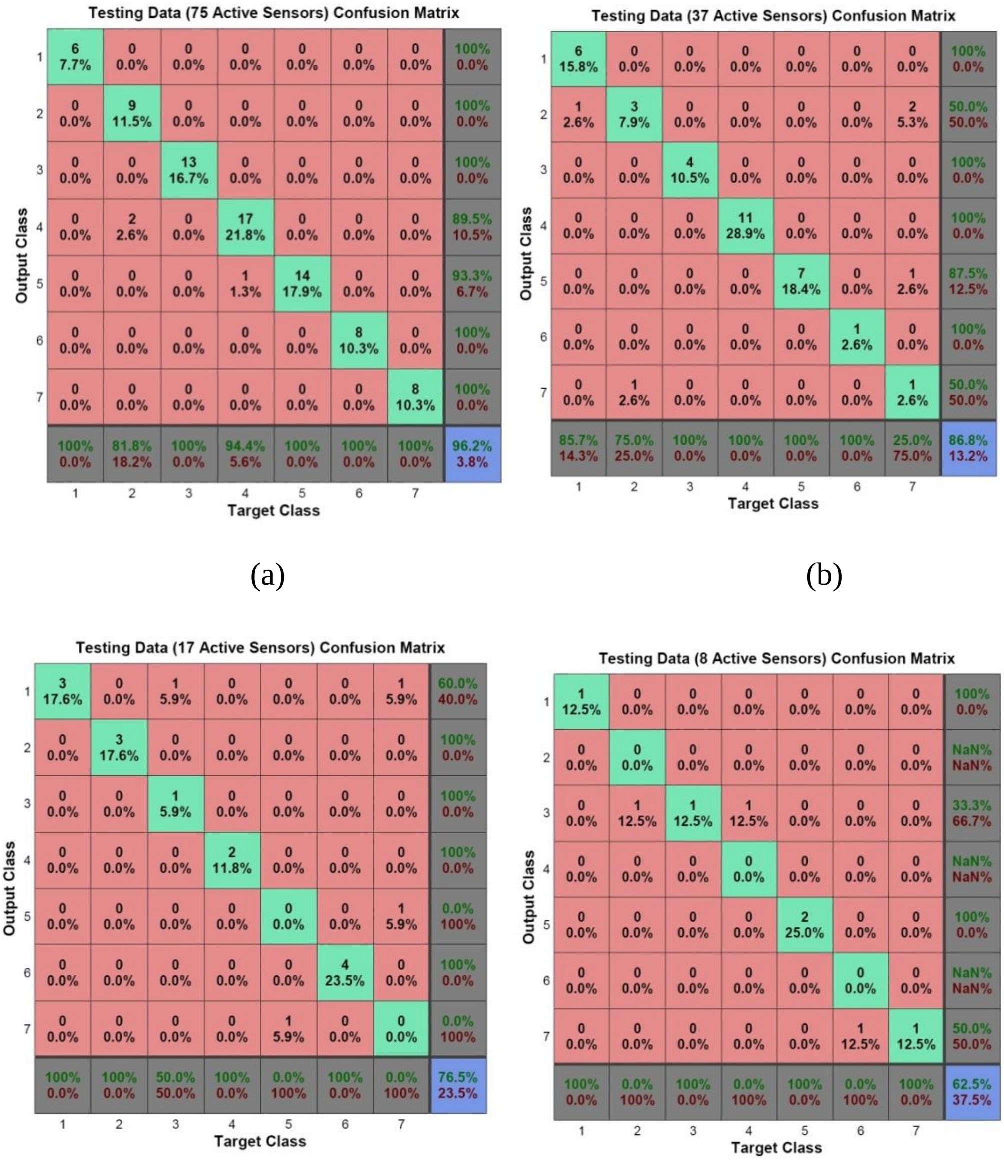
As it is seen in Table 25, all of the models have a good performance on the calibration, validation and testing data. The performance of the model with 75, 37, and 17 active sensors is very satisfying. As expected, the precision of the models decreases with decreasing the number of sensors. However, considering the Area Ratio values, it is seen that even for the total of 112 sensors, only 0.195% of the plate is covered by the sensors (less than 1%). This is indeed a very low rate for achieving such high detection accuracy. Evidently, by decreasing the number of damage classes, the accuracy of the models remarkably increases.

Thus, it is possible to reduce the number of sensors and yet have good detection accuracy by decreasing the number of damage classes.

Table 25. The damage detection performance of the GPLR method for the gusset plate

	Damage Detection Performance			
Number of potential sensors	Calibration	Validation	Testing	Area Ratio
112 (75*)	98%	100%	96%	0.195%
56 (37)	88%	85%	87%	0.097%
28 (17)	83%	78%	77%	0.049%
8	73%	88%	63%	0.014%

* The numbers in the parentheses represent the active sensors for each configuration.



Testing Data (75 Active Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	6 7.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	9 11.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	13 16.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	2 2.6%	0 0.0%	17 21.8%	0 0.0%	0 0.0%	0 0.0%	89.5% 10.5%
5	0 0.0%	0 0.0%	0 0.0%	1 1.3%	14 17.9%	0 0.0%	0 0.0%	93.3% 6.7%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 10.3%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 10.3%	100% 0.0%
	100% 0.0%	81.8% 18.2%	100% 0.0%	94.4% 5.6%	100% 0.0%	100% 0.0%	100% 0.0%	96.2% 3.8%

Target Class

(a)

Testing Data (37 Active Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	6 15.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	1 2.6%	3 7.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 5.3%	50.0% 50.0%
3	0 0.0%	0 0.0%	4 10.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	11 28.9%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 18.4%	0 0.0%	1 2.6%	87.5% 12.5%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 2.6%	0 0.0%	100% 0.0%
7	0 0.0%	1 2.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 2.6%	50.0% 50.0%
	85.7% 14.3%	75.0% 25.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	25.0% 75.0%	86.8% 13.2%

Target Class

(b)

Testing Data (17 Active Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	3 17.6%	0 0.0%	1 5.9%	0 0.0%	0 0.0%	0 0.0%	1 5.9%	60.0% 40.0%
2	0 0.0%	3 17.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	1 5.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	2 11.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 5.9%	0.0% 100%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 23.5%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 5.9%	0 0.0%	0 0.0%	0.0% 100%
	100% 0.0%	100% 0.0%	50.0% 50.0%	100% 0.0%	0.0% 100%	100% 0.0%	0.0% 100%	76.5% 23.5%

Target Class

(c)

Testing Data (8 Active Sensors) Confusion Matrix

Output Class	1	2	3	4	5	6	7	
1	1 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	0 0.0%	1 12.5%	1 12.5%	1 12.5%	0 0.0%	0 0.0%	0 0.0%	33.3% 66.7%
4	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 25.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 12.5%	1 12.5%	50.0% 50.0%
	100% 0.0%	0.0% 100%	100% 0.0%	0.0% 100%	100% 0.0%	0.0% 100%	100% 0.0%	62.5% 37.5%

Target Class

(d)

Figure 78. Confusion matrixes of the best GPLR models for the gusset plate analysis

3.3.3.2.4 Uncertainty Analysis

Similar to the PNN analysis, the calibration, validation and testing sets are polluted with random noise to simulate the performance of real sensors. To this aim, Gaussian noise is added to the input vectors. Different noise levels are considered for the analysis (10%, 20%, and 30%). The GEP algorithm is run for all the combinations shown in Table 25 with noise-polluted data. Figure 79 visualizes the best classification results for different number of sensors with various noise levels. Comparing the results shown in Tables 25 and Figure 79, it can be observed that increasing the noise level does not influence on the performance of the models on the calibration data. For the validation and testing data, the trends of the results are complicated. In these cases, it can be seen that the accuracy of the models gradually decreases with increasing of the noise level. The results for the testing data indicate that nearly all of the models have a good accuracy for noise levels up to 20%. However, the performance of the models on the testing data still remains satisfactory even for 30% noise level.

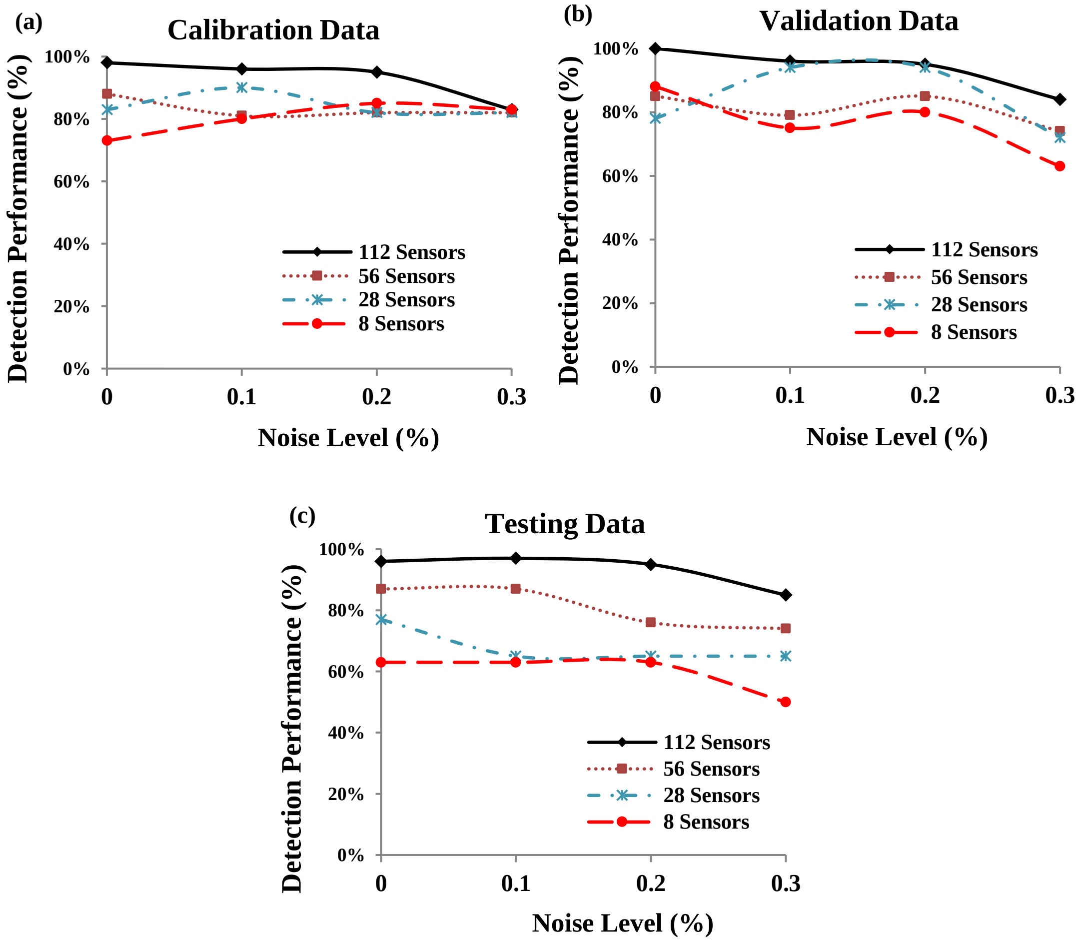


Figure 79. The damage detection accuracy of the best GPLR models versus the noise levels for different number of sensors

3.3.3.2.5 Sensitivity Analysis

Providing an estimation of relative importance of each parameter is an important concern for the aim of model developments or field investigations. As discussed before, among several parameters given in Eq. (18), the role of the Z-functions for detecting damage is much more notable. Herein, a sensitivity analysis is conducted to provide a more in depth understanding of the contribution of these important parameters to the detection of damage. The importance of each variable is evaluated by randomizing its values and then computing the decrease in the performance of the 28 damage models. Thereafter, the results for all variables are normalized. Figure 80 presents the results of the sensitivity analysis. This figure shows the average of the relative importance of the variables in the 7 damage models for each of the 112, 56, 28, and 8 sensor configurations. Moreover, the overall importance of the Z-functions is calculated by averaging of their relative importance values in all of the 28 damage models. Considering the overall importance of the variables, it can be observed from Figure 80 that $Z_{\mu 2}$with an overall importance of 31.4% is the most important parameter to detect damage compared to the other predictor variables. $Z_{\sigma 1}$ and $Z_{\mu 2}$ have a fairly similar importance, while the models seem to be less sensitive to the changes of $Z_{\sigma 2}$ in comparison with other three parameters.

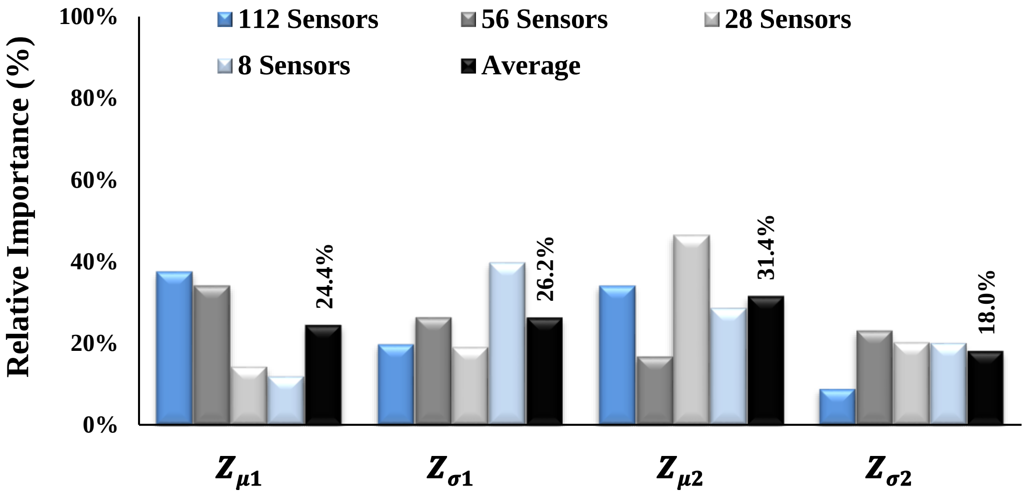


Figure 80. Contributions of the predictor variables in the GPLR damage detection models

3.4 Conclusions

The data fusion in a network of self-powered strain sensors includes statistical and AI analyses. The effect of group of sensors is studied for the numerical simulations with a number of active sensors. For the statistical analysis, failure of pavement systems and crack growth detection in steel plate is investigated numerically and experimentally. A more in depth numerical analysis is done on the distortion-induced fatigue cracking in bridge girders. On this basis, it is shown that the STD of μ and σ of group of sensors has a sound relationship with the damage progression. Based on the results, this parameter increases with the progression of damage. Another important observation is the possibility of localizing the damage and quantifying its severity through the analysis of the PDF shifts. The variations of both μ and σ can be used to localize the damage. In order to evaluate the damage severity, the changing trend of σ is found to be very informative. For the cases where the statistical approach fails to provide decent detections, the GPLR and PNN methods can be used to discover the patterns in a network of sensors. Performance of these methods is verified on a simply supported beam and bridge gusset plate. Among the defined predictor features, the *Z*-functions contain useful information for the detection of the damage state of the investigated system. The GPLR method performs superior to PNN for the case of 112, 56, and 28 organized sensors.

CHAPTER IV

DAMAGE IDENTIFICATION USING A HYBRID NETWORK OF SELF-POWERED ACCELEROMETER AND STRAIN SENSORS

4.1 A Hybrid System for Damage Identification

Hybrid sensor networks are emerging as viable alternatives to traditional SHM systems. These networks enhance the reliability and robustness of the monitoring systems by combining or fusing different sensor types. Information fusion is a key challenge in successful implementation of the multi-sensor networks (Grosse et al., 2008; Adewuyi et al., 2009; Li et al., 2010; Lu et al., 2014). This issue becomes more challenging for hybrid WSNs that are both monitoring systems and autonomous data acquisition nodes. Research in the previous phase of this study has been focused on structural/infrastructure damage identification based on the analysis of the data from the PZT strain sensors. However, a major limitation of these types of sensors is that they can give only partial (local) information about the health status of the structures. Thus, the main goal of this phase of study is to establish a global-local damage detection approach through the analysis of the data from a hybrid network of self-powered accelerometer and strain sensors. To this aim, PZT cantilever-based accelerometers are designed for global damage detection, while PZT strain sensors are employed to provide local damage detection for the host structure. To illustrate the sensing scheme and decision fusion in a self-powered WSN, both numerical and experimental studies are conducted in this project. The experimental studies are carried out on an aluminum plate with bolted connections. The PZT strain sensors and accelerometers are installed on the fabricated fixture. Different damage states are obtained by loosening (or removing) one bolt at a time from the plate. When all bolts are fully tightened, the structure is healthy. A bolt loose condition corresponds to a damaged state. Damage localization requires identification of which of the bolts is loose or missing. The PZT accelerometers are expected to detect changes in the characteristics of the overall structure. The damage details and its approximate location can be identified by the PZT strain sensors. The structure of the proposed hybrid system is schematically shown in Figure 81. Besides, a limited study is performed on the PFG sensors with variable injection rates. The information provided by these sensors can be interpreted through a similar procedure followed for sensors with constant injection rate.

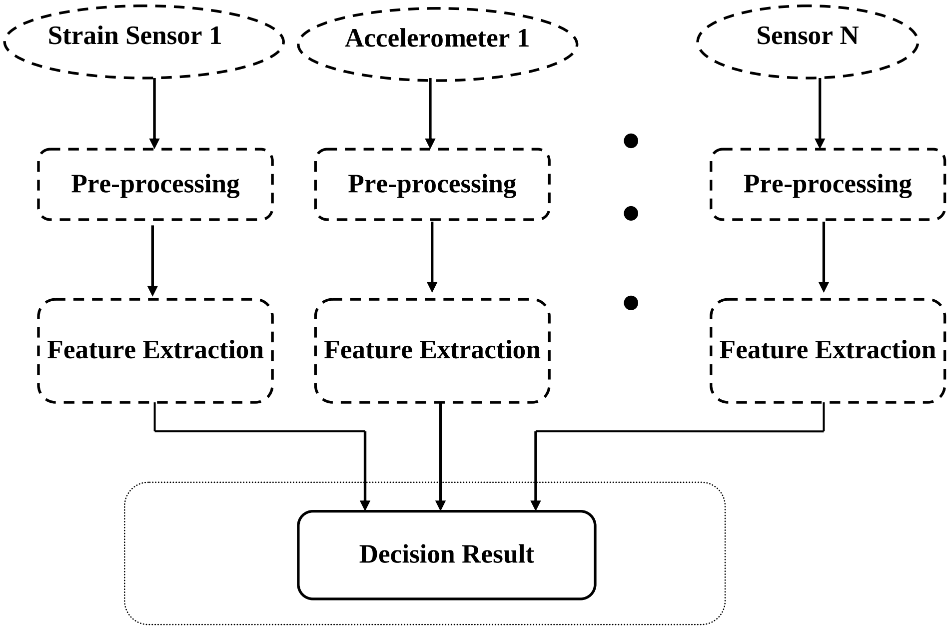


Figure 81. The architecture of the strain sensor and accelerometer hybrid system

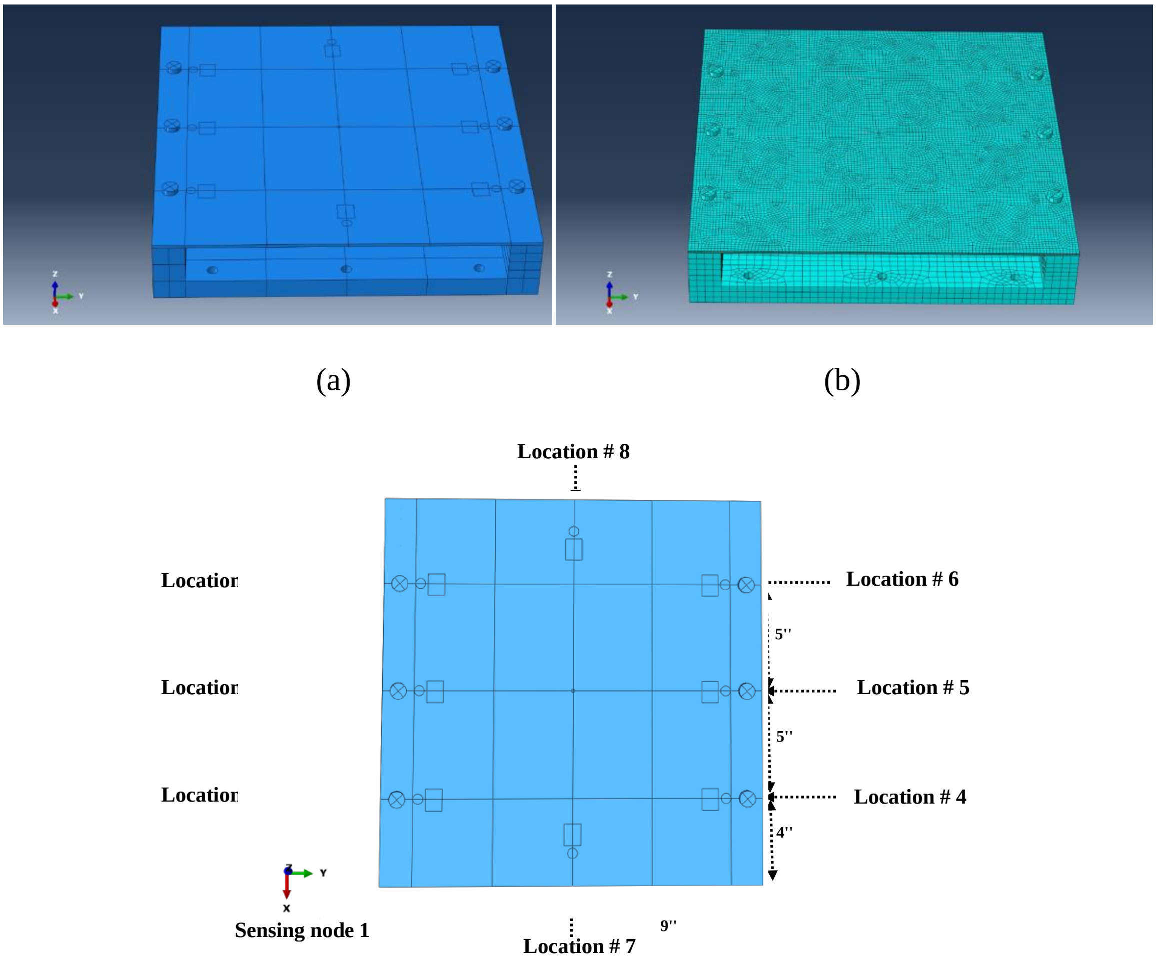
4.2 Numerical Study of a Plate with Bolted Connections

For the numerical study, different 3D FE models are developed for each damage state to analyze the dynamic response of a plate specimen under cyclic loading. The strain data, as well as corresponding features for the data acquisition nodes (potential sensors) are extracted to detect the bolt damage states (fractional torque). ABAQUS/CAE 6.11 is used for the modeling and post-processing of the results. Dynamic implicit procedure is considered for the FE modeling. Following the experimental setup detailed in next section, the locations of the strain sensing nodes are defined by making circular (D = 12 mm) partitions, respectively. These locations are then called by software to extract the average strain amplitude in each partition. The length, width and thickness of the top plate are 457.2 mm (18"), 457.2 mm (18"), 4 mm (5/32"), respectively. Figure 82 shows the assembly of the plate and the meshed geometry.

Damage is introduced by loosening (or removing) one bolt (Bolt # 2 shown in Figure 82) at a time from the plate. 9.5 mm (3/8") stainless steel bolts are considered for the analysis. The nominal bolt torque representing undamaged (healthy) plate is taken 11298 N.mm (100 in. lb). Simulations have been performed with a specific bolt at 100% (healthy), 50%, 10%, and 0% (physically removed) of this nominal torque. The defined damage states are as follows:

- Undamaged: 100% of nominal torque
- 50% Loosened: 50% of nominal torque
- 90% Loosened: 10% of nominal torque
- Missing Bolt: 0% of nominal torque

In order to simulate the bolt damage states (fractional torque), the corresponding torque is converted to axial bolt "clamp" force and applied to Bolt # 2. The coefficient of friction between bolt and plate is taken 0.2.



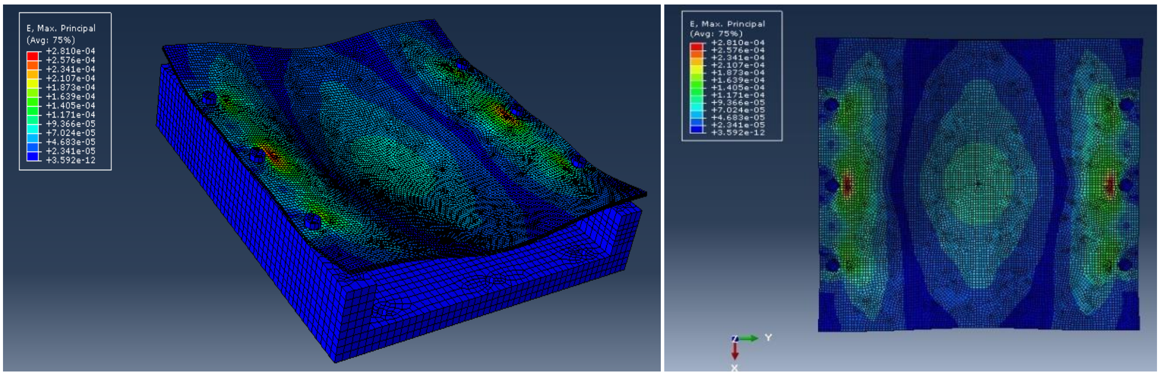
(a) (b)

(c)

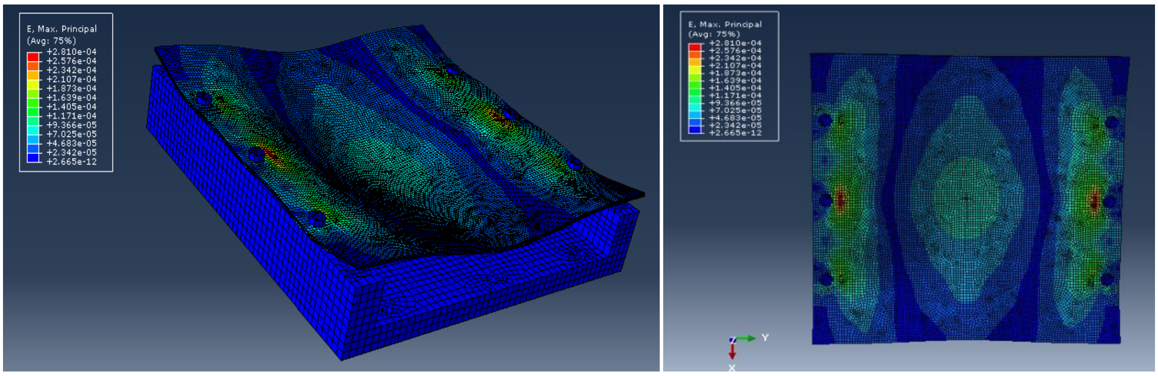
Figure 82. Assembly and meshing of the plate with bolted connections

The displacements and the in-plane rotations of the lower fixture are constrained to satisfy the boundary conditions. The simulations are performed at 8 Hz loading frequency for a 0.32 mm displacement applied to the center of the plate. The undamaged plate is modeled using 73483 linear hexahedral elements of type C3D8R. Mesh refining is adopted to capture high stress and strain concentration around the bolts. The used material for the fixture is aluminum with E = 69 GPa, ν = 0.33, and density = 2700 kg/m^3.

Figure 83 and Table 26 show the maximum principal strains obtained from the FE analyses for different damage states. The maximum strains are experienced at Sensing node 1 followed by Sensing node 5 for the undamaged state. Sensing nodes 7 and 8 give the lowest strain values. While the strain value at Sensing node 2 decreases by reducing the torque on Bolt # 2, it has an increasing trend for the other sensors. That is to say, by loosening Bolt # 2, other fastening bolts will experience higher stress concentration. However, it can be seen in Figures 23(a) and (b), and Table 26 that there is no change in the strain values when Bolt # 2 is tightened more than 50% of the nominal torque. The most eminent strain changes can be seen at Sensing node 2 where the strain value drops from 256.20 με to 82.11 με by transitioning from the 50% Loosened to 90% Loosened damage state.



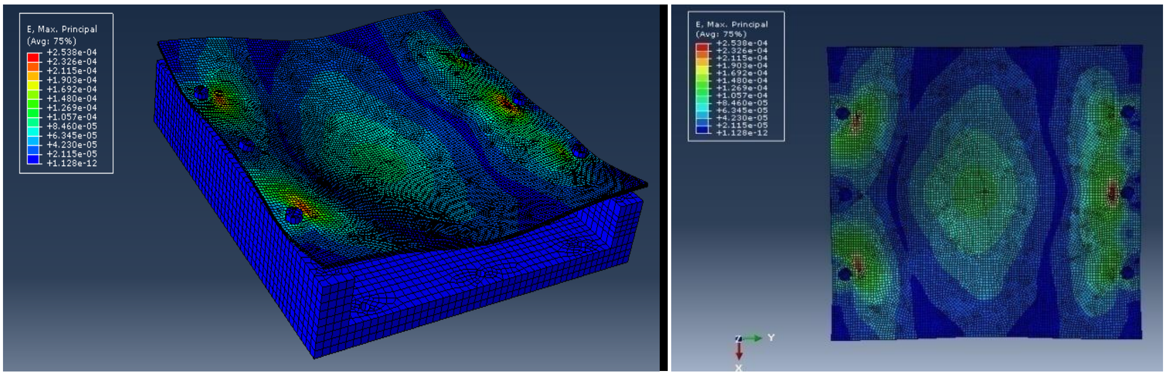
(a) Undamaged



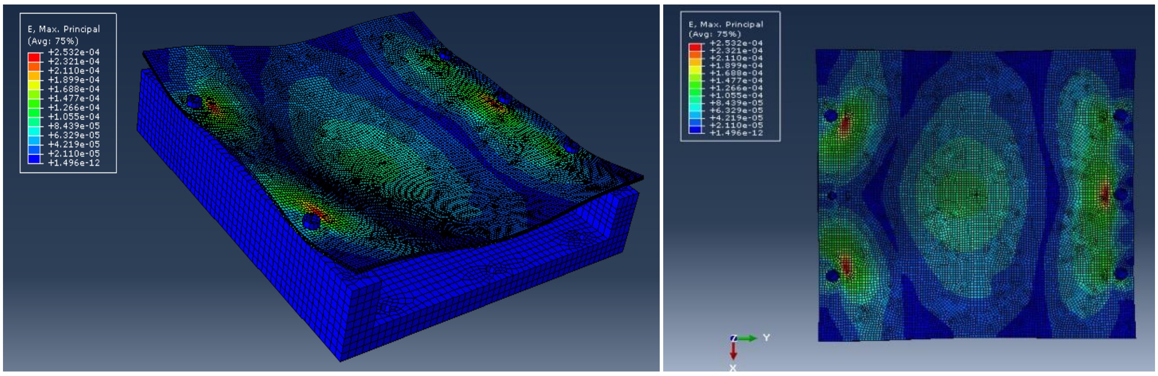
(b) 50% Loosened

Figure 83. The FE results for the plate with bolted connections

Figure 83. (cont'd)



(c) 90% Loosened



(d) Missing Bolt

Table 26. Maximum principal strains for different damage states for the plate with bolted connections

	Maximum Principal Strain (με)			
Sensor #	Undamaged	50% Loosened	90% Loosened	Missing Bolt
Sensing Node 1	138.08	138.07	176.16	182.77
Sensing Node 2	256.21	256.20	82.11	75.85
Sensing Node 3	143.01	143.03	183.18	189.44
Sensing Node 4	140.26	140.26	143.69	148.40
Sensing Node 5	246.11	250.84	252.80	252.81
Sensing Node 6	142.29	142.28	144.36	150.19
Sensing Node 7	40.57	40.57	37.76	36.53
Sensing Node 8	40.84	40.84	37.84	36.88

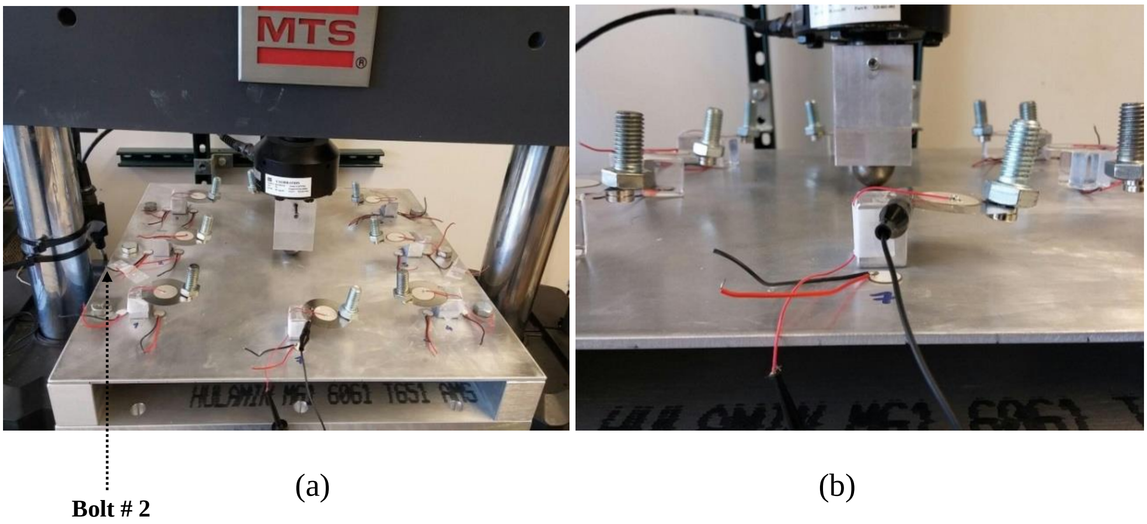
4.3 Experimental Study of the Plate with Bolted Connections

For the experimental study, an aluminum plate with bolted connections is fabricated.

The PZT strain sensors and accelerometers are installed on the structure. The strain sensors are PZT-5A ceramic discs and accelerometers are PZT-5H ceramic round bimorph cantilevers with attached proof mass. The experimental setup is shown in Figure 84. The dimensions of the fixture and bolt sizes are the same as the FE model. The lower fixture is fully connected to the MTS shaft. A set 8 PZT ceramic discs (D = 12 mm) and round bimorph cantilevers are attached to the plate surface (see Figure 84). PZTs 1 to 6 are installed near the bolts and PZTs 7 and 8 are placed near the edge of the plate to check their sensitivity to damage. The round bimorph flexing transducers used in this study are piezoelectric ceramic disks adhered to a metal plates of brass (Figure 85). The resonant frequency, resonant resistance, and capacitance of the bimorphs are, respectively, 3.2kHz±0.5 kHz, 500ohm, and 26000pF± 30% at 100 Hz. The bimorph cantilevers are tightly glued to the supporting block, which is made of aluminum and has dimensions of 25.4mm×25.4mm×19.8mm. Under larger deflections, the mechanical strains cause the PZT to lose its piezoelectric properties (Elvin et al., 2006). On the other hand, about 7 volts should be generated by PZTs to activate the available sensors. In order to find a tradeoff between the maximum allowable tip displacements, piezoelectric failure strains, and amount of generated energy to activate the sensors, preliminary experiments are carried out using different tip masses, and loading frequencies and amplitudes. As a result, the tests are conducted at 8 Hz loading frequency, 0.32 mm displacement and a tip mass of 42 g. A 9.5 mm (3/8") bolt and two neodymium magnets are used as tip masses. The point load is applied to the center of the plate using a steel sphere with 25.4 mm (1") diameter. Before starting the test, a preload equal to 0.2 kN is applied to the plate to ensure it is seated on the fixture. Thereafter, 100 cyclic displacements are applied. The PZT outputted voltage is read on NI 9220 data acquisition system in parallel with the PFG sensor.

The structural damage considered in this experiment is a single fastener failure (Bolt # 2). Detection of this type of damage with piezoelectric transducers has been the focus of some studies (Olson et al., 2006; Zein-Sabatto et al., 2011). Four different classes of structural damages are defined. The 100% torque value for the fastening bolts is 11298 N.mm (100 in. lb). Classes 1–4 of damage correspond to Undamaged (100% of nominal torque), 50% Loosened (50% of nominal torque), 90% Loosened (10% of nominal torque), and Missing Bolt (0% of nominal torque/physically removed). A calibrated torque wrench is used to apply the required torque values to Bolt # 2. Responses are collected from PZT strain sensors and accelerometers for multiple measurements.

Tables 27 and 28 show the maximum voltage outputted by the PZT strain sensors and accelerometers for different damage states. As seen in Table 27, the output of Strain sensor 2 decreases by reducing the torque. The other sensors generate more voltage due to damage progression as they experience higher strains. This is an expected case and is in agreement with the FE results. Sensors 7 and 8 deliver the lowest voltage values and are not sensitive to the changes of the boundary condition. There is a 306% decrease in the voltage delivered by Strain sensor 2 due to reduction of the nominal torque from 50% to 90%. This is the highest change among all sensors and clearly shows the damage location. Referring to Table 28, the voltage delivered by Accelerometers 1 to 6 decreases due to damage progression. This shows that base excitation decreases in the entire system when Bolt # 2 is loosened or removed. Accordingly, less voltage is delivered by the accelerometers. Interestingly, Accelerometers 7 to 8 are sensitive to the changes of the boundary condition and generate higher voltage as Bolt # 2 is loosened. Note that, unlike the FE results, strain sensors and accelerometers are sensitive to the reduction of the torque from the Undamaged to 50% Loosened phase.



(a) (b)

Figure 84. Test setup and sensors locations for the plate with bolted connections

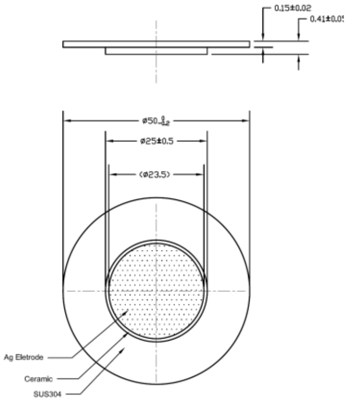


Figure 85. The round piezoelectric bimorph actuator manufactured by Multicomp Inc.

Table 27. Maximum voltage delivered by the PZT strain sensors installed on the plate with bolted connections

	Maximum voltage (volt)			
Sensor #	Undamaged	50% Loosened	90% Loosened	Missing Bolt
PZT Strain Sensor 1	9.23	9.33	9.88	10.11
PZT Strain Sensor 2	8.96	8.53	2.10	1.38
PZT Strain Sensor 3	7.50	7.57	7.95	7.98
PZT Strain Sensor 4	8.46	8.59	8.66	8.82
PZT Strain Sensor 5	9.49	9.55	9.60	9.72
PZT Strain Sensor 6	8.68	8.95	9.00	9.18
PZT Strain Sensor 7	1.32	1.31	1.33	1.23
PZT Strain Sensor 8	1.18	1.16	1.16	1.17

Table 28. Maximum voltage delivered by the PZT accelerometers installed on the plate with bolted connections

	Maximum voltage (volt)			
Sensor #	Undamaged	50% Loosened	90% Loosened	Missing Bolt
PZT Accelerometer 1	9.12	9.00	8.60	8.33
PZT Accelerometer 2	9.72	9.30	8.59	8.10
PZT Accelerometer 3	9.57	9.33	8.72	8.42
PZT Accelerometer 4	9.81	9.46	9.44	9.40
PZT Accelerometer 5	10.26	10.07	9.96	9.84
PZT Accelerometer 6	10.35	9.99	9.84	9.50
PZT Accelerometer 7	8.65	8.80	9.03	9.22
PZT Accelerometer 8	7.92	7.99	8.18	8.53

4.4 Damage Growth Detection Based on the FE Results

The maximum of the strain value extracted from the FE simulations is 256.21 με. Therefore, the upper threshold is set to 260 με. On this basis, the preselected strain levels are taken 30.0, 68.3, 106.7, 145.0, 183.3, 221.7, and 260.0 με. Following the procedure described in Section 2.1, the μ and σ values are extracted and used to plot the PDFs corresponding to each sensor (Figure 86). As seen in Figures 86(a)-(c), the shape of PDFs sensing nodes close to Bolt # 2 notably changes due to damage progression (Sensing nodes 1-3). For Sensing nodes 1 and 3, PDFs shift to the left and expand which implies an increase in strains due to the loosening of Bolt # 2. An inverse trend can be observed for Sensing node 1 (Figure 86(b)) as the strain values are reducing at this node. Referring to Figure 86, the most significant changes pertain to Sensing node 2. However, for most cases, the FE analysis only captures the transition from the 50% Loosened to 90% Loosened damage states. The other observation is that Sensing nodes 7 and 8 cannot detect any changes in the system (Figures 86(g) and (h)).

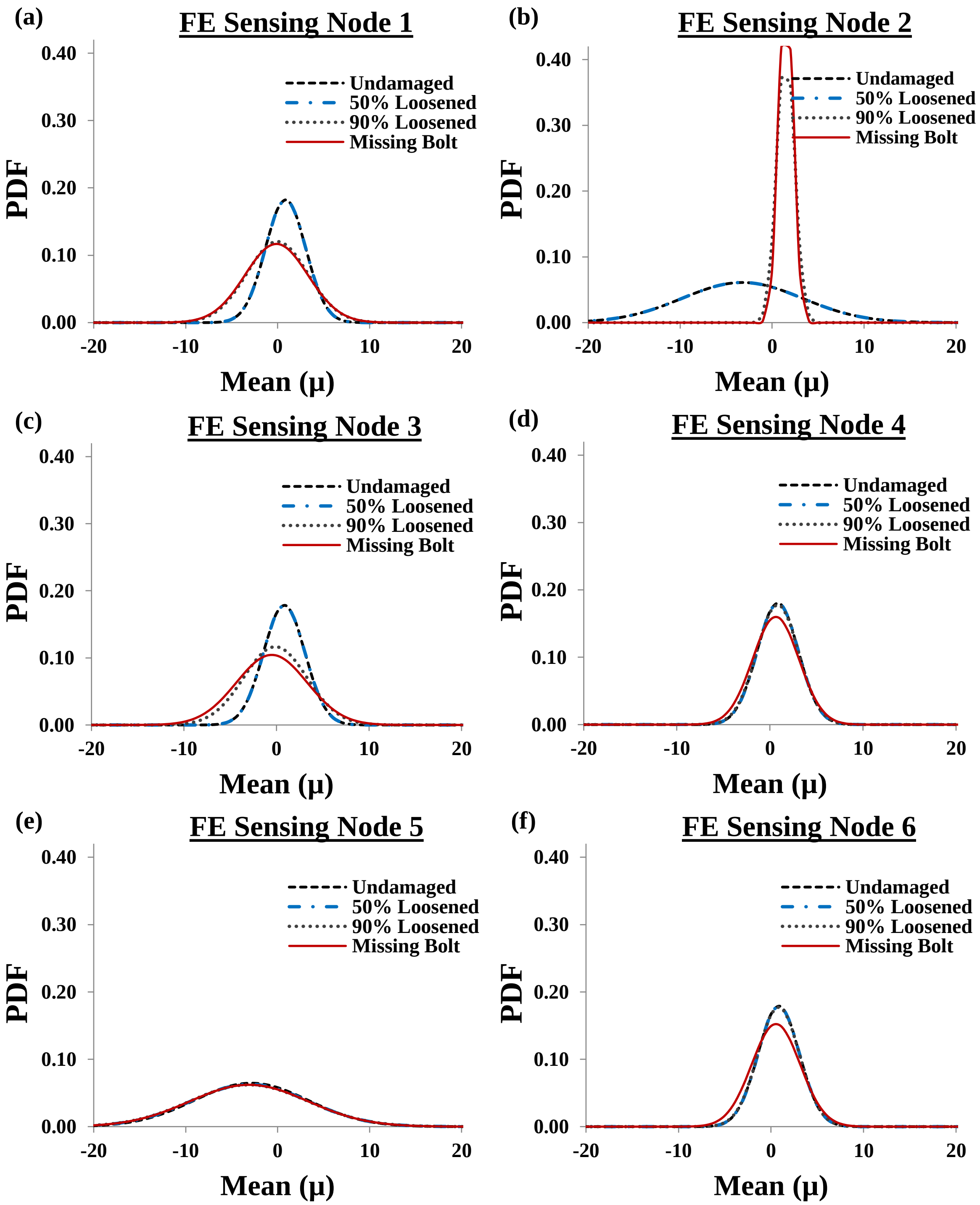
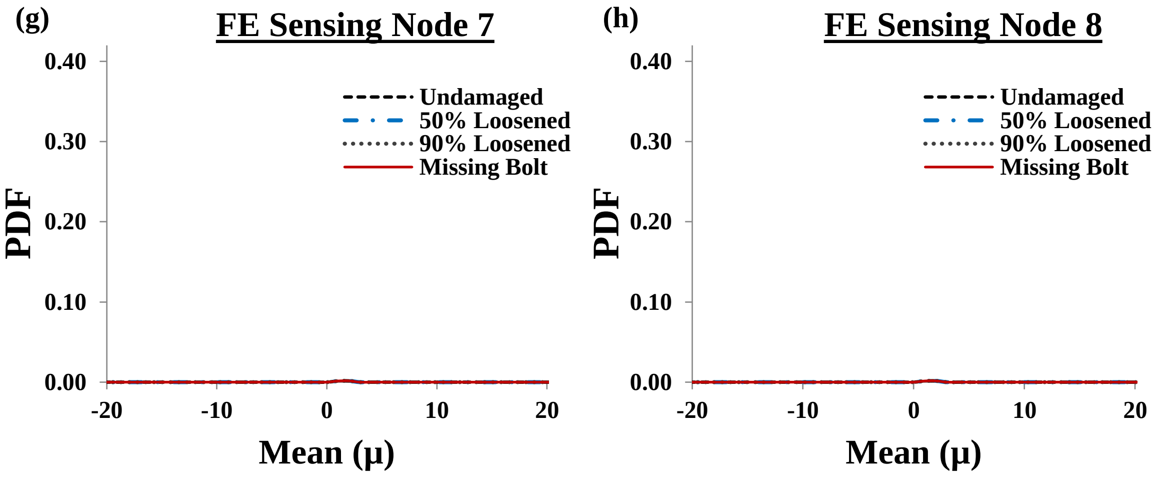


Figure 86. Change of PDFs curves due to damage progression based on the FE simulations of the plate with bolted connections

Figure 86. (cont'd)



4.5 Damage Growth Detection Based on the Experimental Results

The SWS used for this phase of study contains 7 floating-gates that are programmed to trigger at 7.2, 7.7, 7.9, 8.7, 9.05, 9.7, and 10.2 volts, respectively. These levels are used to obtain the μ, σ, and PDF plots for each sensor. The PDF plots for the PZT strain sensors and accelerometers are shown in Figure 87. As seen in Figure 87, the shapes of PDFs change due to damage progression, except for Strain sensors 7 and 8. For Strain sensors 1, 3, 4, 5, and 6 that are experiencing higher strains, the estimated values of PDF shown in Y-axes are decreasing. Therefore, the plots move to left and expand (Figures 87(a), (e)), (g)), (i), and (k)). This is contrary to the trends captured by Strain sensor 2 (Figure 87(c)). Accelerometers 1 to 8 are experiencing lower excitations when Bolt # 2 is loosened or removed. Thus, their corresponding PDF values increase and the plots shift to right and contract by transitioning from Undamaged to Missing Bolt mode (Figures 87(b), (d), (f)), (h), (j), (l), (n) and (p)).

Comparing the results obtained by the PZT strain sensors and accelerometers, two important points can be inferred. First, all of the accelerometers provide a fairly more consistent behavior than the strain sensors. This is more evident for Accelerometers 7 to 8 that provide sound trends and capture the mode changes from Undamaged to Missing Bolt. In this context, Strain sensors 7 and 8 are not sensitive to the changes of the boundary condition.

The other important observation is that the PDFs corresponding to the strain sensors give a good insight into the location of damage. Comparing the measurements from Strain sensor 2 and Accelerometer 2 (Figures 87 (b) and (c)), it can be seen that Strain sensor 2 provides a much more eminent change of μ and σ with damage progression. In this case, the μ and σ of Strain sensor 2 change 905% and 216% by transitioning from the Undamaged to Missing Bolt mode, respectively. These rates are much lower for Accelerometer 2 and are equal to 272% and 179% for μ and σ, respectively. Note that, among the strain sensors, only Strain sensor 2 which is the closest to the damage zone experiences such fluctuations. This is while most of the accelerometers seem to be sensitive to the changes in the system (e.g. Accelerometers 2 and 8) and therefore might not be used for determining exact location of damage. The results clearly indicate that it is feasible to monitor changes in the characteristics of the overall structure even with one accelerometer (e.g. Accelerometer 7 or 8), while several PZT strain sensors might be needed to localize the damage.

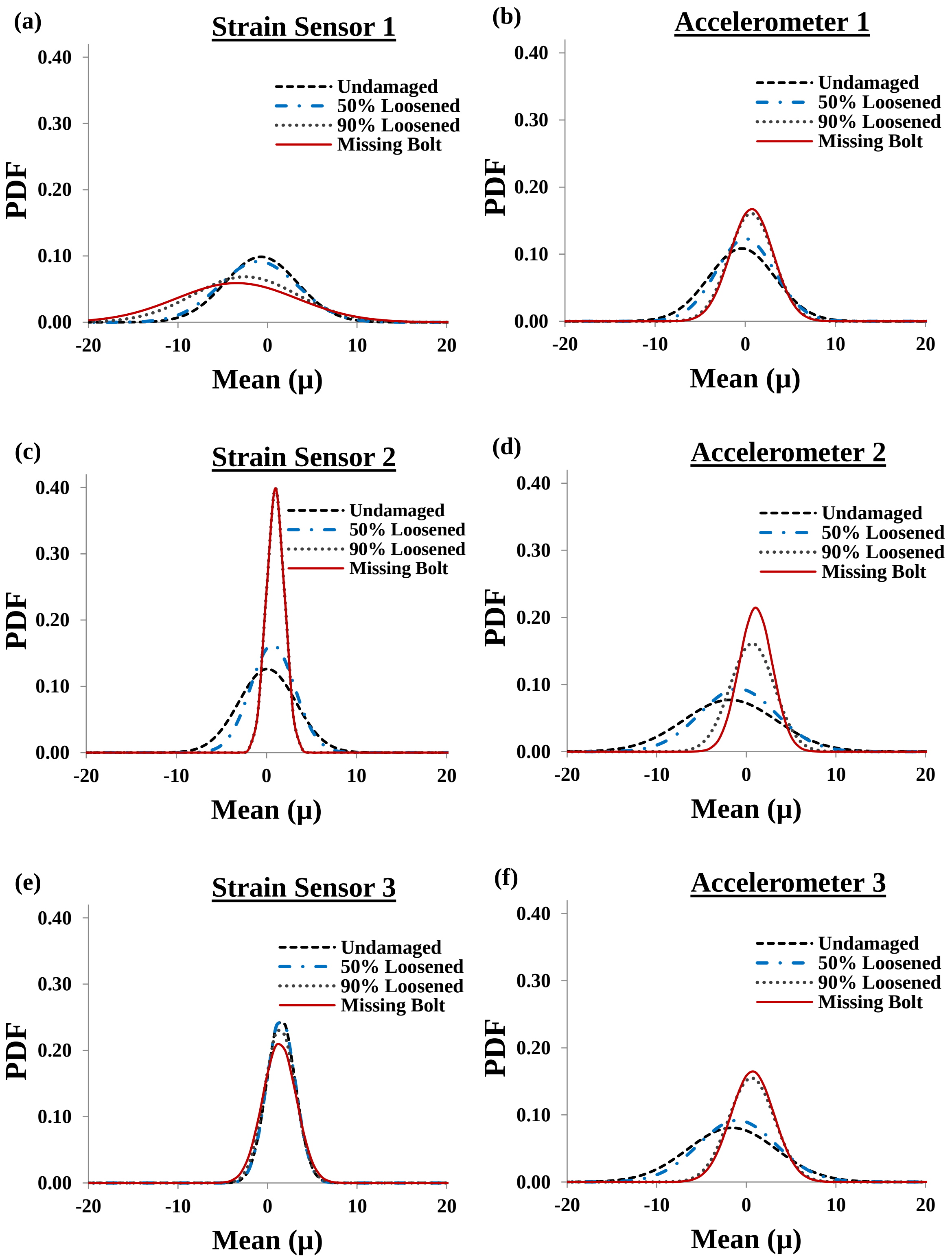


Figure 87. Change of PDFs curves due to damage progression based on the outputs of the PZT strain sensors and accelerometers installed on the plate with bolted connections

Figure 87. (cont'd)

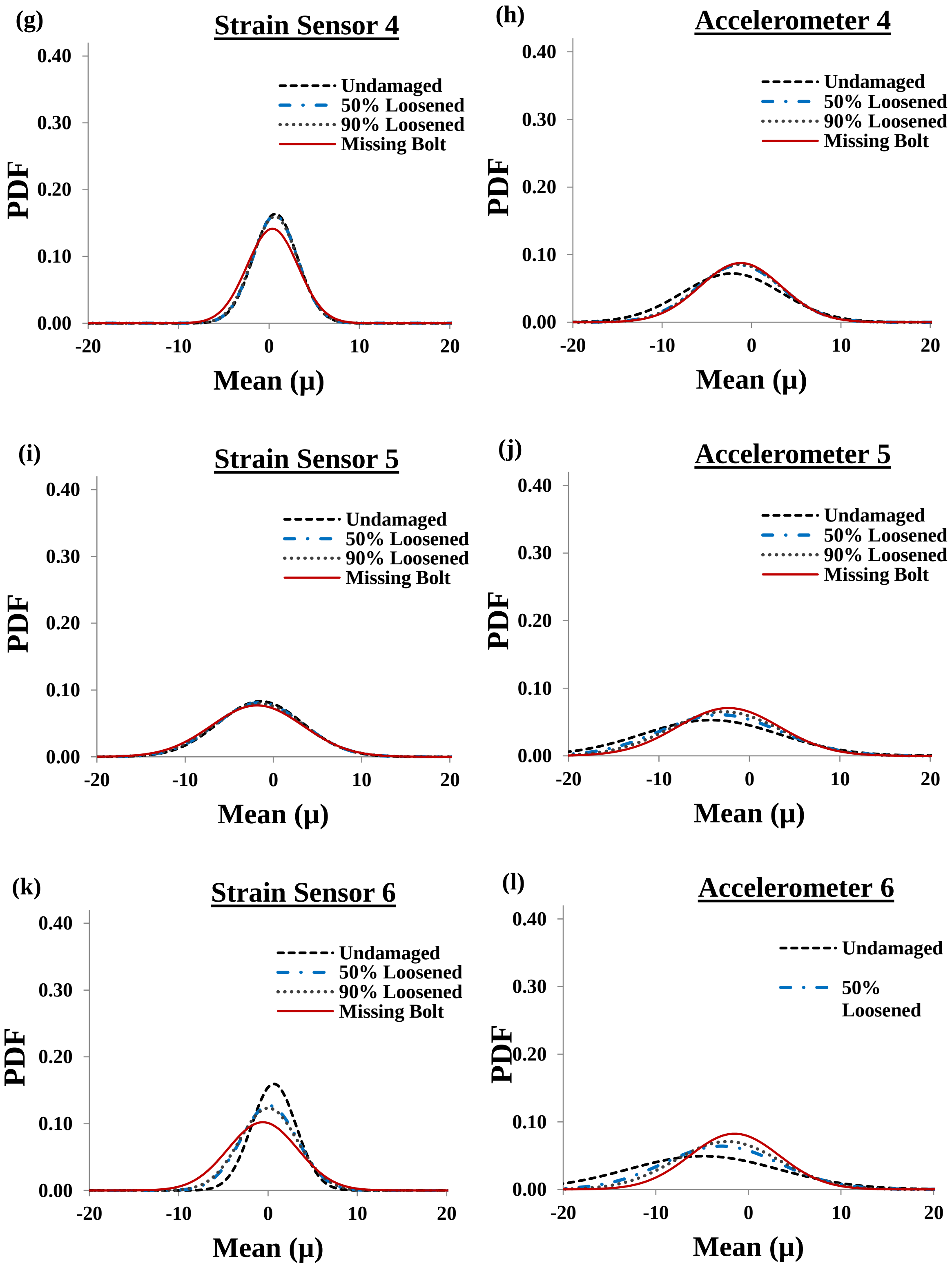
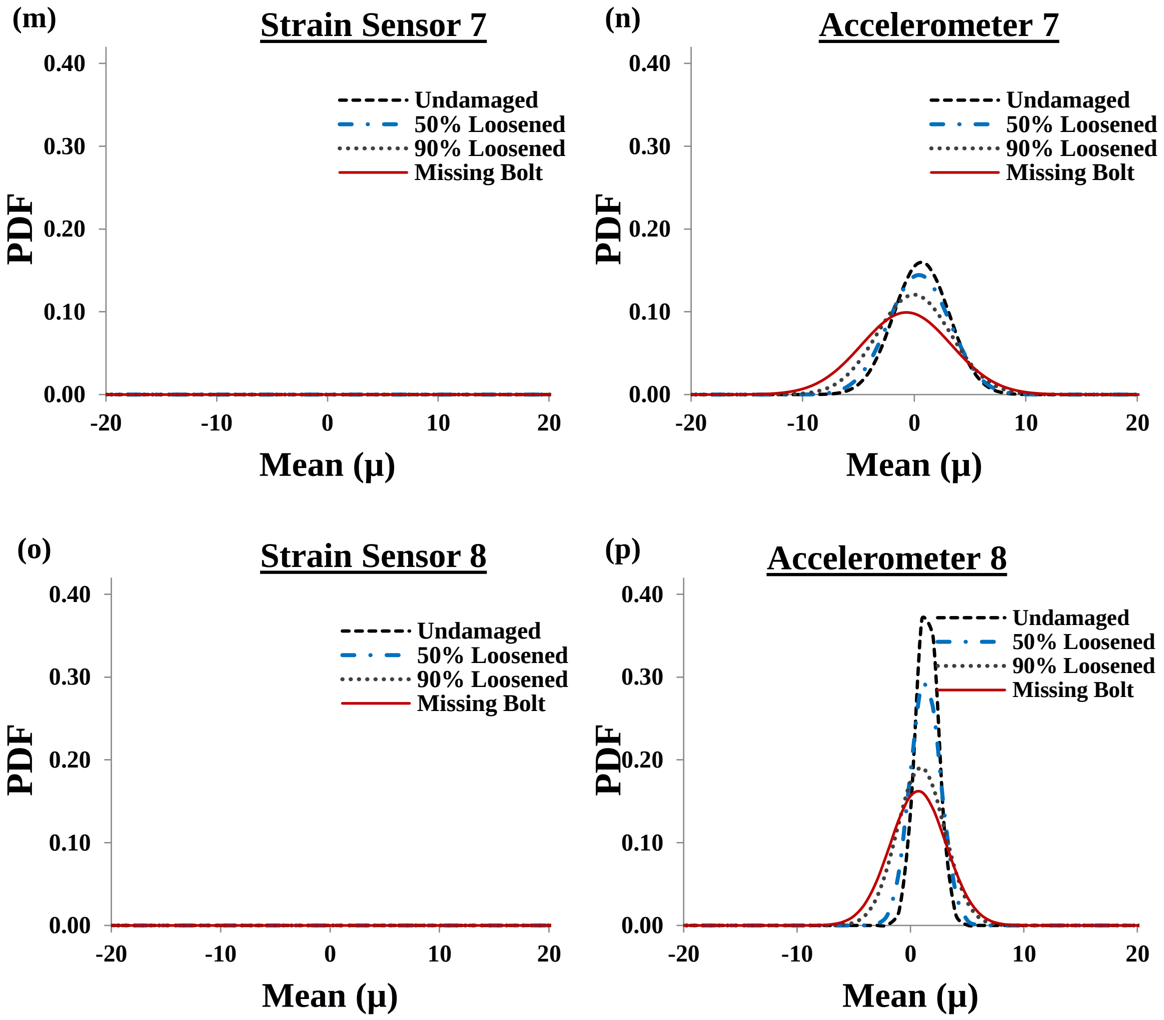


Figure 87. (cont’d)



4.6 Discussion

As discussed before, the prototypes of the PFG sensor can have floating-gates with constant and variable electron injection rates. The injection rate controls the injection of the electrons into the gate and therefore the voltage droppage across it. The cumulative time at specific pre-defined voltage thresholds is proportional to this droppage (see Eq. (5)). Based on the results presented in this research, the output of the sensor with the constant injection rates can be characterized by a Gaussian CDF. In this case, µ and σ accounting for the load and frequency variability define the SWS output data. However, analysis of the sensor outputs still remains challenging for the case of a sensor with variable injection rates. For this case, a similar approach to the one defined for sensors with constant injection rates can be followed by fitting curves to the voltage droppage histograms and extracting new features. Herein, a limited study is carried out on the results provided sensors with variable injection rates. A more comprehensive study will be the focus of future research. For each of the tests, the initial voltage of the gates is set to 1.2 volts. Then, the voltage is read after applying 100 cycles. After each test, the sensor is tunneled and injected to reset all gates to almost the same voltage.

For brevity, only the results for the Strain sensor 2 and Accelerometer 2 located close to Bolt # 2 are presented. Figures 88 and 89 show the voltage changes across the floating-gates of the SWS for the Strain sensor 2 and Accelerometer 2, respectively. The activation thresholds for Gate 1, ..., Gate 7 are 7.2, 7.7, 7.9, 8.7, 9.05, 9.7, and 10.2 volts, respectively. As seen in these figures, the recorded droppage of the sensor voltage highly depends on the damage state. As soon as the voltage generated by a PZT exceeds a threshold corresponding to one of the gates, the procedure of electron injection initiates, and subsequently the voltage of that gate starts decreasing. Conversely, if the voltage generated by the PZT transducer is below the gate injection threshold, the injection stops, and therefore the charge on the floating-gate will not change. As an example, the voltage delivered by the Strain sensor 2 dropped from 8.96 to 1.38 volts due to the damage progression from the Undamaged to Missing Bolt mode (see Table 27). At this location, the stress concentration decreases by loosening Bolt # 2. As long as the voltage delivered by PZT is above the 7.2 volts, at least one of the gates is injecting (Figure 88). In this case, as soon as the damage approaches the 90% Loosened phase, the strain decreases and the voltage amplitude drops to 2.1 volts which is below the injection thresholds for all of the gates. Hence, the channels shut off. Gates 5-7 are activated for voltage amplitudes higher than 9 volts. Since the maximum voltage delivered by Strain sensor 2 for all of the damage states is 8.96 volts, none of these gates are activated and the initial voltage of these gates (1.2 volts) remains constant.

A similar behavior can be observed from Figure 89 for Accelerometer 2. This accelerometer is experiencing lower excitations as Bolt # 2 is loosened. Referring to Figures 89(a), (c), (e), (g), and (i), Gates 1 to 5 have experienced a drop in the charge. This is because the voltage generated by Accelerometer 2 exceeds their thresholds for some of the damage states. Gate 6 is activated only for the Undamaged phase in which the voltage of Accelerometer 2 exceeds 9.7 volts (Figures 89(k)). The maximum voltage delivered by Accelerometer 2 for all of the damage states is 9.72 volts and therefore Gate 7 is not activated.

Apparently, the gate activation can be considered as an indicator of damage occurrence. However, the other important observation from Figures 88 and 89 is that the voltage droppage rate for each gate is also a good indicator of damage progression. For instance, consider the response of Gate 1 to the voltage generated by Accelerometer 2 for the Undamaged, 50% Loosened, 90% Loosened, and Missing Bolt states (Figures 89(a)). Since the excitation was higher for the Undamaged mode than that for the other modes, the injection time was higher, and therefore the voltage on the gate dropped more for this mode. The same is true for the 50% Loosened-90% Loosened, and 90% Loosened-Missing Bolt cases. The reduction in voltage droppage rates due to bolt loosening or removal are clearly presented in Figures 88 ((b), (d), (f), and (h)) and Figures 89 ((b), (d), (f), (h), (j), and (l)).

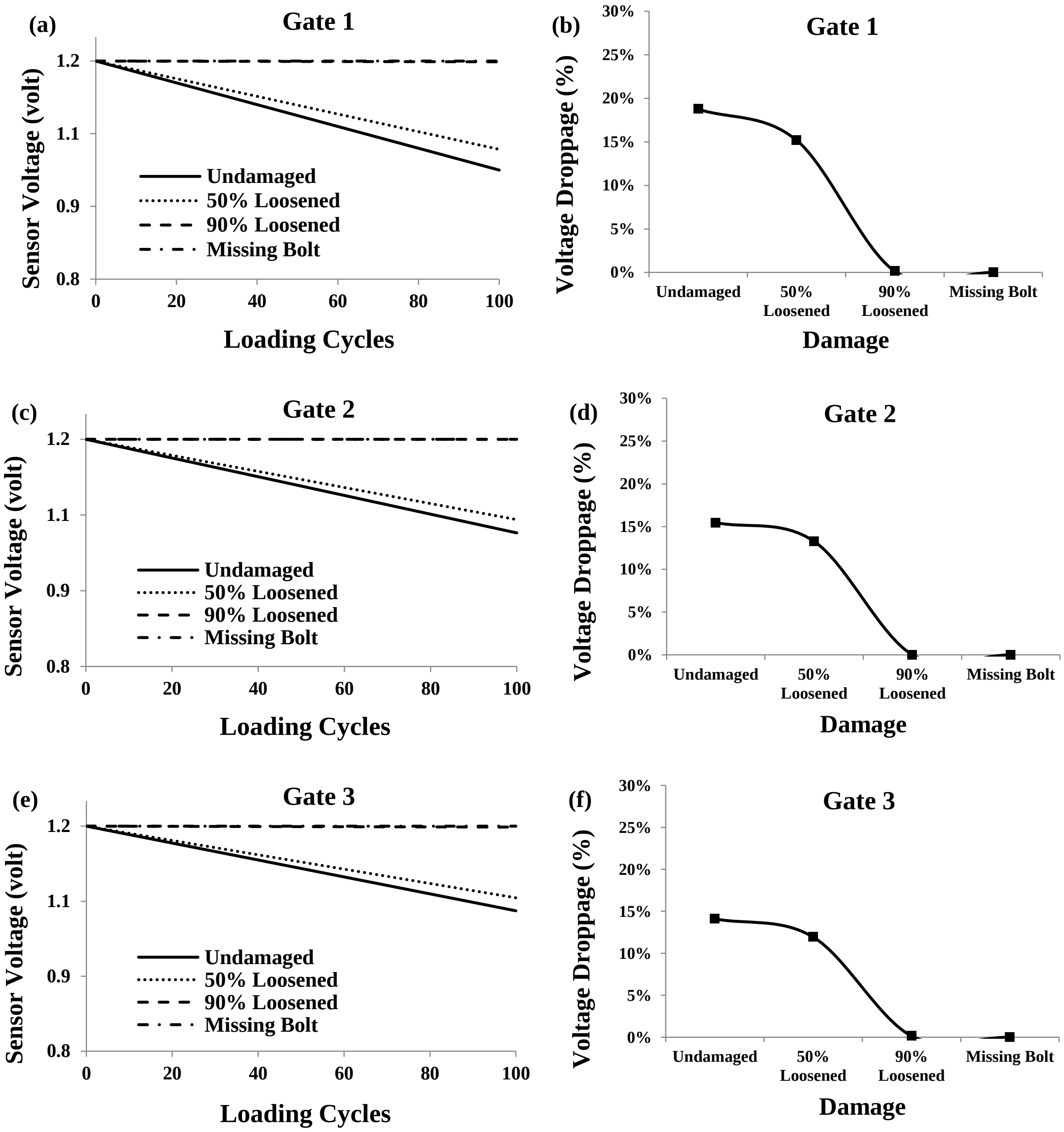


Figure 88. Voltage changes across the floating-gates of the SWS: PZT strain sensor 2 installed on the plate with bolted connection

Figure 88. (cont'd)

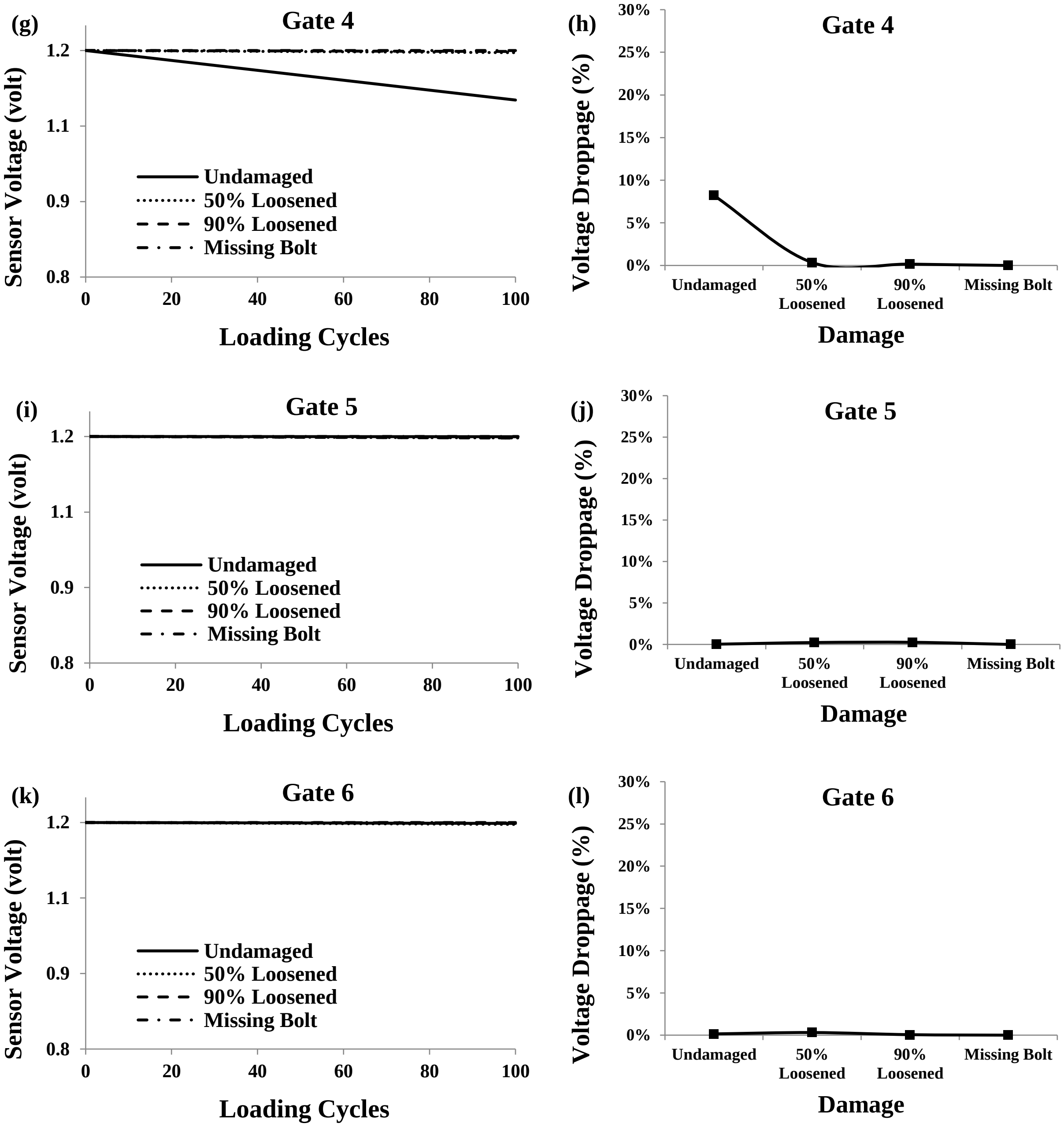
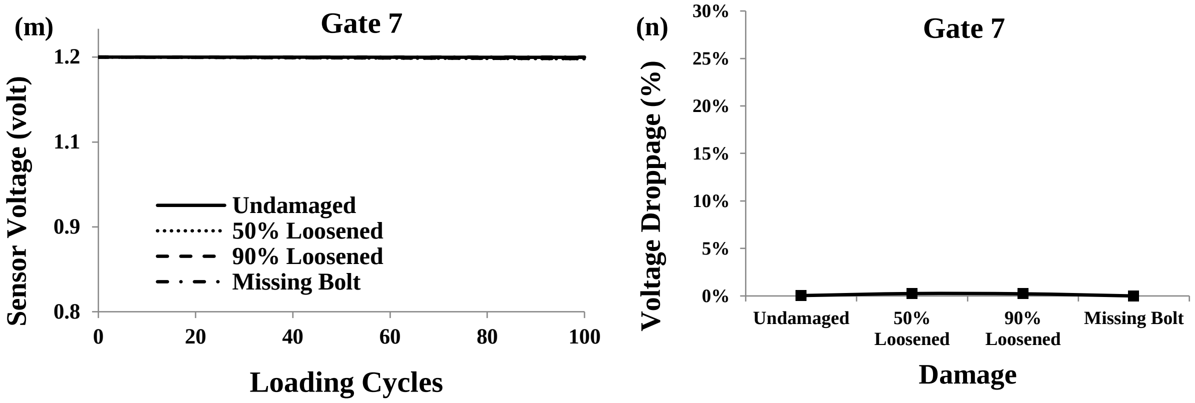


Figure 88. (cont'd)



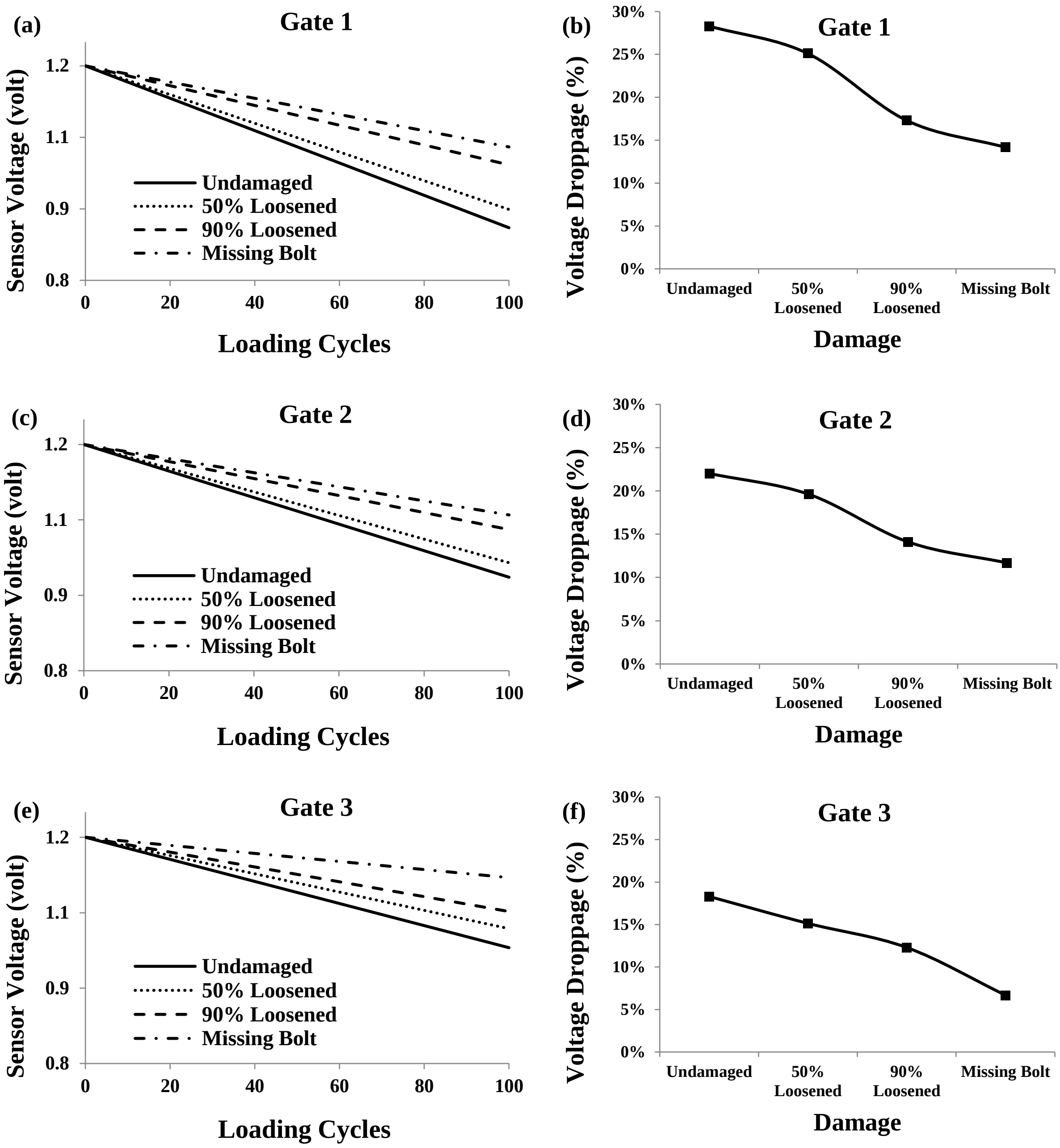
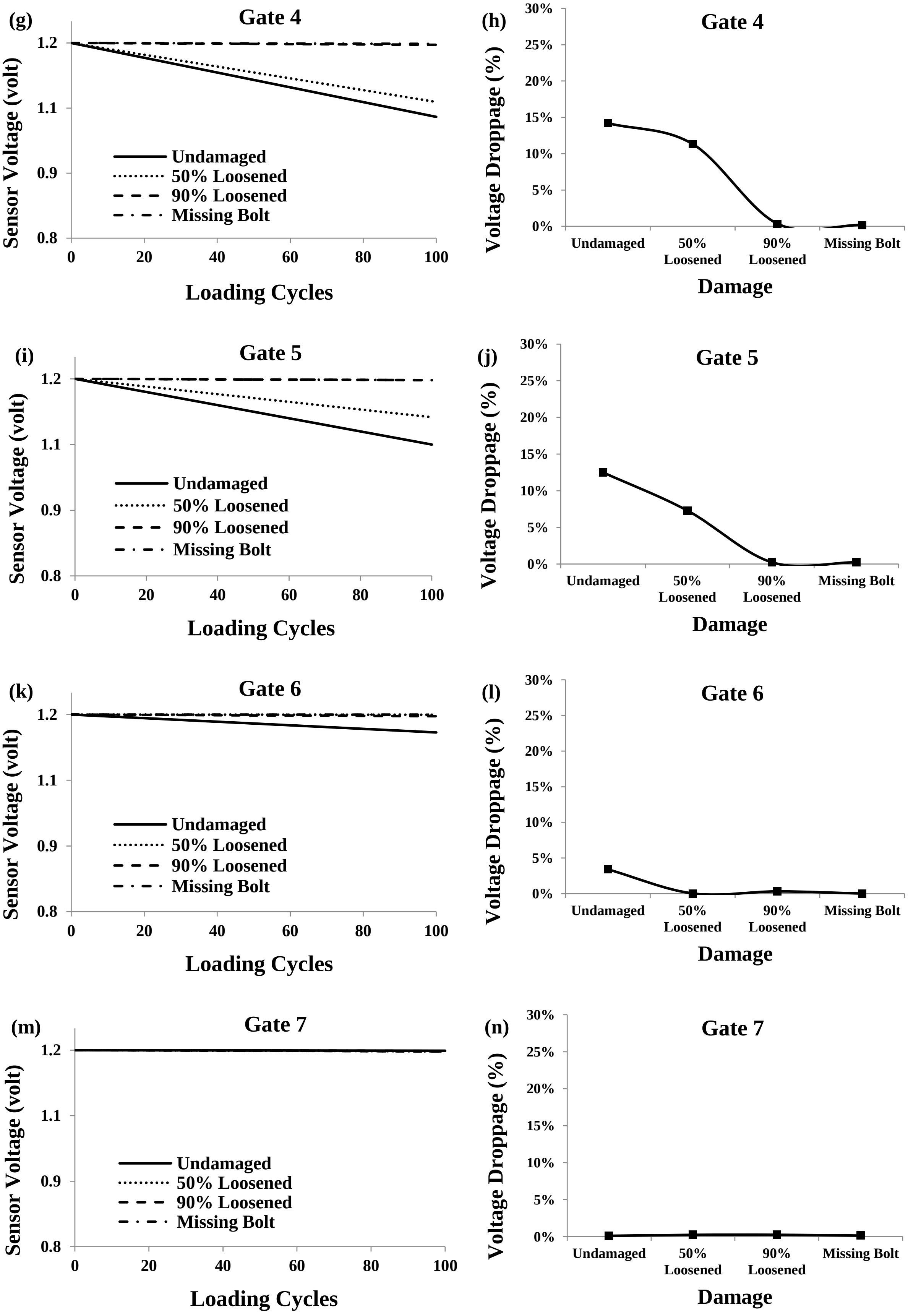


Figure 89. Voltage changes across the floating-gates of the SWS: PZT accelerometer 2 installed on the plate with bolted connection

Figure 89. (cont'd)



CHAPTER V

SUMMARY AND CONCLUSION

This study presents a framework for the damage identification in civil infrastructure based on advanced data mining and self-powered monitoring approaches. It focuses on two main technical challenges for successful development of Smart Cities: (1) knowledge discovery/data mining, and (2) innovative power solutions for heterogeneous WSNs. Thus, it can be an integral part of the next generation of Smart Civil Infrastructure that is capable of self-diagnosis of damage before the occurrence of any failure. The proposed framework is established through the integration of statistical, AI and FE methods to interpret the limited data stored on-board the SWS. A new probabilistic AI-based technique, called GPLR, is proposed for multi-state damage classification system. Data obtained from the FE simulations and experimental study of hybrid SWS networks is used to calibrate the sensor-specific data interpretation algorithms. The framework consists of multilevel data interpretation strategies for the structural/infrastructure damage identification. The damage identification process is based on the data from individual self-powered strain sensors, data fusion in a network of self-powered strain sensors, and data fusion in a hybrid network of self-powered accelerometer and strain sensors. For each of these levels, several damage indicator features are extracted upon the simulation of the compressed data stored in memory chips of the SWS. The first strategy is focused on finding a reasonable relationship between the damage indicator features of point sensors and damage progression. However, a major limitation of this type of analysis is that the strain sensors give only partial information about the health status of the structures. The data fusion stages are based on the information provided by a group of sensors, termed as "group effect". Moreover, combination of the data from a network of accelerometer and strain sensors results in developing an integrated global-local damage detection approach. Extensive numerical and experimental studies are carried out to evaluate the performance of the proposed damage detection approach. Several infrastructure systems are analyzed with different type of damages. The investigated cases are failure of simply supported beam under three-point bending, continuous health monitoring of pavement systems, failure of gusset plate of the I-35W highway bridge in Minneapolis, Minnesota, distortion-induced fatigue cracking in steel bridge girders, and crack growth detection in steel plates under a uniaxial tension mode. 3D dynamic FE models are developed for each of the cases. The experimental studies are carried out on a steel plate subjected to an in-plane tension, an aluminum plate with bolted connections, and on asphalt concrete specimens in three-point bending mode. PZT-5A ceramic discs and PZT-5H bimorph accelerometers are placed on the surface of the plates to measure the delivered voltage in each damage phase. For the asphalt experiments, a new miniaturized spherical packaging system is designed and tested to protect the PZT-5A ceramic discs embedded inside the specimen. For some of the investigated cases, uncertainty analyses are performed through the contamination of the damage indicator features with different noise levels. The results indicate that the proposed I/SHM systems are efficiently capable of detecting different damage states. However, the following points can also be inferred from the results presented in this research:

- Among different probability distributions evaluated in this study, the Gaussian CDF is found to be the most efficient for the characterization of the data from the SWS with constant injection rate.
- The FE simulation and laboratory testing results for the pavement system, steel plate and steel bridge girders indicate that damage can be identified using individual self-powered strain sensors.

- Based on the individual sensor analysis, the PDFs shift to left (μ decreases) and their width increases (σ increases) due to the damage progression. Variation of σ is found to be a better damage indicator.
- The PDFs corresponding to the sensors that are closer to critical locations are experiencing more prominent shifts compared to those for the sensors far from the damage zone. This issue provides the possibility of localizing the damage and quantifying its severity.
- Based on the data fusion in a network of self-powered strain sensors, the group effect can be efficiently used for damage identification. On this basis, the STD of μ and σ of group of sensors has a sound relationship with the damage progression. This parameter increases with the progression of damage.
- For the cases where the statistical approach fails to provide good detections, the GPLR and PNN methods can be utilized to capture the patterns in a network of sensors. The performance of the proposed GPLR method is much better than PNN for the studied cases of 112, 56, and 28 sensor configurations.
- The proposed GPLR method is a hybrid multi-class classification system that assigns probabilities to model scores. It provides simple yet accurate classification models that can be programmed on-board the sensors. Unlike the PNN models, the derived GPLR-based models can readily be used via spreadsheet programming. Besides, the straightforward structure of the GPLR method allows for its future application to a variety of multi-class classification problems in engineering domain.
- Among several extracted damage indicator features, the *Z*-functions contain more useful information for the detection of the damage state.

- Based on the analysis of the plate with bolted connections, it is observed that the PZT bimorph accelerometers provide a fairly more consistent behavior than the PZT strain sensors.
- While some of the PZT strain sensors are not sensitive to the changes of the boundary condition, the bimorph accelerometers capture the mode changes from Undamaged to Missing Bolt.
- The PDFs corresponding to the strain sensors give a better insight into the location of damage compared to the accelerometers. In this context, the changes of μ and σ of the strain sensor adjacent to the damaged bolt are much more eminent than those for the accelerometers.
- It is feasible to monitor changes in the characteristics of the overall structure even with one accelerometer, while several PZT strain sensors might be needed to localize the damage.

Although the efficiency of the proposed frameworks is verified for some structures but their applicability is not limited to the investigated cases. In fact, these integrated sensing systems can be modified to become building blocks of future medical, mechanical, civil, transportation, and aerospace long-term sensing technologies. However, there are still some challenges that are the focus of future research as follows:

- Considering different type of damages, as well as different type of structures for the case study of damage identification.
- Although the FE simulation and laboratory results still remain satisfactory, verification of the proposed approach with full-scale experiments would be an interesting topic for future study.
- Studying the performance of the whole sensing system including wireless communication more in-depth.

- Performance evaluation of the proposed damage detection approaches for materials with plastic behavior.
- Development of a process to find the optimal number of the memory cells of the wireless sensors to achieve the most appropriate strain pattern.
- Development of a more comprehensive approach for optimal sensor placement (OSP) using robust optimization algorithms.
- The GPLR and PNN algorithms are supervised learning methods based on the labeled data. Future research may focus on detecting damage for the cases where the target classes are unknown. To this aim, unsupervised learning algorithms such as self-organizing map (SOM) seem to be the most efficient tools.
- Verification of the long-term performance of the sensing system for real-life structure and environmental conditions. The obtained results are based on the experiments carried out at room temperature (25°C). In this context, future research can focus on the determination of the effects of ambient temperature on the performance of the proposed system.
- Analysis of the outputs of sensors that have floating-gates with variable injection rates. In this case, voltage droppage across the floating gates may be analyzed rather than obtaining the corresponding cumulative time histograms.
- Developing a new framework based on the obtained features to predict the remaining life of civil infrastructure systems.

APPENDIX

The VBA Codes for Different Sensors Configurations

(Note: More, d(0), …, d(3) represent $Z_{\mu 2}$, $Z_{\sigma 2}$, $Z_{\mu 1}$, and $Z_{\sigma 1}$, respectively.)

A.1.: The VBA Codes for 128 Organized Sensors

```
' State 1 (Intact, a= 0 mm)
'-------------------------
Function gepModel1(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C4 As Double = 5.994
   Const G1C8 As Double = 0#
   Const G1C7 As Double = -7#
   Const G2C5 As Double = -6#

   Dim y As Double
   y = 0#

   y = ((d(2) + ((d(0) + d(0)) + G1C4)) - ((G1C8 * G1C8) + (G1C7 * d(1))))
   y = y + (((d(0) ^ 2) * ((d(3) ^ 3) / d(1))) + Exp(G2C5))
   y = y + (Exp(((d(0) / d(2)) ^ 2)) ^ 4)
   Const SLOPE As Double = 6.51918988984368E-77
   Const INTERCEPT As Double = -9.08512660782129

   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel1 = probabilityOne
End Function
```

```
' State 2 (a = 10 mm)
'-------------------------
Function gepModel2(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C9 As Double = -7.91208
   Const G2C6 As Double = -10#
   Const G2C1 As Double = 2.7
   Const G2C2 As Double = -8#
   Const G3C9 As Double = -2.97
   Dim y As Double
   y = 0#
   y = ((((G1C9 ^ 4) + (d(3) ^ 2)) * (G1C9 + d(3))) * Exp(d(1)))
   y = y + (((((G2C2 ^ 4) * G2C6) ^ 4) - ((G2C6 * G2C1) - (G2C2 * d(1)))) ^ 2)
   y = y + ((d(0) ^ 4) * Exp((((G3C9 ^ 4) * d(1)) * (d(3) * G3C9))))
   Const SLOPE As Double = 4.49875667638908E-30
   Const INTERCEPT As Double = -35642828.6530698

   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel2 = probabilityOne
End Function
```

```
' State 3 (a = 90 mm)
'-------------------------
Function gepModel3(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G2C3 As Double = -3#
   Dim y As Double
   y = 0#
   y = (d(0) / (d(2) - Exp(((d(3) - d(2)) - Exp(d(2))))))
   y = y + ((((d(0) + d(0)) + d(0)) + (d(0) - G2C3)) / ((d(3) / d(2)) * (d(2) / G2C3)))
   y = y + (d(0) / (d(2) - ((Exp(d(2)) - d(3)) - (d(2) / d(1)))))
   Const SLOPE As Double = 5.51268762967603E-02
   Const INTERCEPT As Double = -12.159410383184
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel3 = probabilityOne
End Function
```

```
' State 4 (a = 170 mm)
'-------------------------
Function gepModel4(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C9 As Double = -9#
   Const G2C9 As Double = 10#
   Const G2C1 As Double = 8#
   Const G3C9 As Double = -3#
   Const G3C3 As Double = 4#
   Dim y As Double
   y = 0#
   y = ((d(1) + d(0)) + (((d(3) + d(0)) + d(3)) * G1C9))
   y = y + ((((d(0) + d(2)) * d(0)) + (G2C9 - G2C1)) / ((d(3) - d(0)) + (d(3) ^ 2)))
   y = y + ((((d(0) * d(3)) + d(2)) + ((d(3) - d(0)) - G3C3)) - G3C9)
   Const SLOPE As Double = 7.56920021002415E-02
   Const INTERCEPT As Double = -8.03210005727878
```

```
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel4 =
probabilityOne
End Function


' State 5 (a = 250 mm)
'-------------------------
Function gepModel5(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C9 As Double = 3#
   Const G3C6 As Double = -
2#
   Dim y As Double
   y = 0#
   y = ((((d(0) / G1C9) * d(1))
* (d(0) + G1C9)) / ((d(3) ^ 2) *
(G1C9 ^ 2)))
   y = y + ((((d(1) + d(2)) -
d(3)) * (d(3) ^ 4)) / ((d(0) ^ 3)
* (d(1) ^ 3)))
   y = y + ((G3C6 - d(2)) ^ 2)
   Const SLOPE As Double =
0.21321778570173
   Const INTERCEPT As
Double = -11.1545771888252
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel5 =
probabilityOne
End Function

' State 6 (a = 330 mm)
'-------------------------
Function gepModel6(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C3 As Double = -
8.8
   Const G1C9 As Double =
10.89
   Const G2C6 As Double =
10.989
   Const G2C4 As Double =
6.993
   Const G3C0 As Double = 9#
   Const G3C8 As Double =
10.01
   Const G3C6 As Double = -
0.99
   Dim y As Double
   y = 0#
   y = ((G1C3 ^ 4) *
Exp((Sqr((d(3) ^ 4)) -
Sqr(G1C9))))
   y = y + (Exp(G2C6) *
((Exp(G2C6) ^ 4) * ((d(0) ^ 4)
* G2C4)))
   y = y + Exp(((((d(2) ^ 4) -
G3C8) * (G3C6 - d(3))) *
G3C0))
   Const SLOPE As Double =
1.09885278840212E-32
   Const INTERCEPT As
Double = -8.6482095125135

   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel6 =
probabilityOne
End Function


' State 7 (a = 410 mm)
'-------------------------
Function gepModel7(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C3 As Double =
5.5055
   Const G2C1 As Double =
6.3
   Const G2C5 As Double =
4.4955
   Const G3C0 As Double =
0.98604593604099

   Dim y As Double
   y = 0#
   y = (((Exp(d(1)) +
Exp(d(1))) ^ 3) * ((d(1) +
G1C3) * (d(1) + d(0))))
   y = y + ((((d(1) - G2C5) *
Exp(d(1))) ^ 3) * ((G2C1 -
d(2)) - (G2C5 - d(0))))
   y = y + (Exp(((((d(1) + d(0))
^ 4) * (G3C0 - d(3))) - d(1))) ^
4)
   Const SLOPE As Double =
1.13128375340451E-87
   Const INTERCEPT As
Double = -8.49403032230808
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel7 =
probabilityOne
End Function
```

A.2.: The VBA Codes for 58 Organized Sensors

```
' State 1 (Intact, a= 0 mm)
'-------------------------
Function gepModel1(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C6 As Double = -
1#
   Const G3C6 As Double = -
6#
   Dim y As Double
   y = 0#
   y = (((((((G1C6 - d(1)) -
d(0)) ^ 2) + d(2)) ^ 3) ^ 3) ^ 4)
   y = y + d(3)
   y = y + (d(2) + G3C6)
   Const SLOPE As Double =
5.5477404146232E-40
   Const INTERCEPT As
Double = -10.6305379773197
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel1 =
probabilityOne
End Function
```

' State 2 (a = 10 mm)
'-------------------------

```
Function gepModel2(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C9 As Double = -
6#
   Const G1C7 As Double = 8#
   Const G1C1 As Double = 9#
   Const G1C2 As Double = -
5556.20791027609
   Const G1C4 As Double =
79.92
   Const G1C3 As Double =
119.709264982805
   Const G2C3 As Double =
4.95
   Const G2C6 As Double = -
8#
   Const G3C4 As Double =
0.999
   Const G3C3 As Double = -
6.534
   Const G3C5 As Double =
10.0999394001515
   Const G3C7 As Double =
5.99338861320479
   Dim y As Double
   y = 0#

   y = ((((G1C2 + G1C4) *
(G1C3 ^ 4)) ^ 3) - ((G1C9 +
G1C7) - (G1C1 + d(1))))
   y = y + ((((d(1) * G2C3) *
(d(1) ^ 4)) ^ 3) / ((G2C3 *
G2C3) - (d(1) - G2C6)))
   y = y + Exp((((G3C4 / d(0))
* (G3C3 * G3C5)) * ((G3C4 *
G3C7) * (d(3) * G3C4))))
   Const SLOPE As Double =
6.15197451294272E-29
   Const INTERCEPT As
Double = 87499680.6969639
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel2 =
probabilityOne
End Function
```

' State 3 (a = 90 mm)
'-------------------------

```
Function gepModel3(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C5 As Double = -
5.05
   Const G1C3 As Double =
1.8
   Const G2C5 As Double =
8.180991819
   Const G2C8 As Double =
3.9204
   Const G3C5 As Double =
3.9996
   Const G3C3 As Double = -
3#
   Dim y As Double
   y = 0#
   y = ((((d(0) * d(2)) ^ 2) *
(G1C5 - d(3))) + ((G1C5 -
G1C3) - (d(3) + d(3))))
   y = y + Exp(((G2C5 / d(2))
* ((d(0) - d(2)) + (G2C8 *
d(0)))))
   y = y + (((G3C5 - G3C5) -
Exp((G3C3 - G3C5))) + d(0))
   Const SLOPE As Double =
4.99843841351498E-131
   Const INTERCEPT As
Double = -8.50691174396521
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel3 =
probabilityOne
End Function
```

' State 4 (a = 170 mm)
'-------------------------

```
Function gepModel4(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C7 As Double = -
6#
   Const G1C2 As Double = -
6.533993466
   Const G1C1 As Double =
4.5
   Dim y As Double
   y = 0#
   y = Exp((((d(3) * d(3)) +
(G1C7 + d(1))) * ((G1C2 -
G1C1) - (d(3) - d(2)))))
   y = y + (((d(2) + ((d(2) -
d(3)) + d(1))) * ((d(1) + d(2)) ^
4)) ^ 4)
   y = y + (((d(0) + (d(2) *
d(2))) * ((d(1) + d(2)) ^ 4)) ^
4)
   Const SLOPE As Double =
6.66222120699797E-33
   Const INTERCEPT As
Double = -8.99890768675053

   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel4 =
probabilityOne
End Function
```

' State 5 (a = 250 mm)
'-------------------------

```
Function gepModel5(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C2 As Double = 6#
   Const G2C1 As Double = -
4#
   Const G2C7 As Double = -
1#
   Const G3C4 As Double = 6#
   Dim y As Double
   y = 0#
   y = (((((G1C2 ^ 2) + (d(3) +
d(3))) + (d(3) / d(2))) ^ 3) ^ 2)
   y = y + ((d(0) ^ 3) *
((((G2C7 * d(2)) - d(0)) + (d(3)
* G2C1)) ^ 3))
   y = y + ((((d(0) ^ 2) * d(0))
* (d(0) * d(2))) + ((d(1) * d(2))
* Exp(G3C4)))
   Const SLOPE As Double =
1.15295630018606E-09
   Const INTERCEPT As
Double = -10.8370323591921
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
```

```
   gepModel5 =
probabilityOne
End Function

' State 6 (a = 330 mm)
'--------------------------
Function gepModel6(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C6 As Double = -
3#
   Const G2C6 As Double = -
440.44
   Const G2C3 As Double = -
80.7999192
   Const G2C1 As Double =
6.2937
   Const G2C7 As Double = -
5.6245322133
   Const G3C4 As Double = -
7#
   Const G3C0 As Double = -
9.98001
   Const G3C7 As Double = -
7#
   Const G3C8 As Double =
9.09
   Dim y As Double
   y = 0#
   y = (G1C6 ^ 3)
   y = y + Exp((((G2C6 - d(0))
* (d(0) * d(2))) - ((G2C3 +
G2C1) * (d(2) * G2C7))))
   y = y + ((((G3C7 * G3C8) -
d(0)) * (d(2) * G3C4)) *
((G3C0 * G3C7) * (d(0) *
d(1))))
   Const SLOPE As Double =
4.3817590910058E-08
   Const INTERCEPT As
Double = -8.41137317801545
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel6 =
probabilityOne
End Function

' State 7 (a = 410 mm)
'--------------------------
Function gepModel7(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C2 As Double = 4#
   Const G2C6 As Double =
4.846643505
   Const G2C7 As Double = -
20.08012008002
   Const G2C2 As Double = -
8.1
   Const G3C6 As Double = -
10.90087909911
   Dim y As Double
   y = 0#
   y = (G1C2 + d(1))
   y = y + (Exp(((G2C7 + d(0))
/ (d(1) + G2C2))) * ((d(1) *
G2C6) * (d(2) - d(1))))
   y = y + Exp(((Exp(d(1)) *
(G3C6 * G3C6)) * ((G3C6 *
d(3)) - (d(0) * d(0)))))
   Const SLOPE As Double =
9.61588479109158E-138
   Const INTERCEPT As
Double = -8.43036650035257

   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel7 =
probabilityOne

End Function
```

A.3.: The VBA Codes for 28 Organized Sensors

```
' State 1 (Intact, a= 0 mm)
'--------------------------
unction gepModel1(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G2C6 As Double =
9.09
   Const G2C3 As Double = -
2#
   Dim y As Double
   y = 0#
   y = ((((d(1) * d(1)) ^ 3) ^ 4)
+ (d(1) * ((d(3) + d(3)) +
d(3))))
   y = y + ((((d(1) * G2C6) +
d(0)) + ((G2C3 + d(0)) + d(0)))
+ d(0))
   y = y + (((d(2) * ((((d(0) +
d(1)) - d(3)) ^ 3) ^ 4)) ^ 3) ^ 2)
   Const SLOPE As Double =
1.43721533852321E-30
   Const INTERCEPT As
Double = -9.22728932274076
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel1 =
probabilityOne
End Function


' State 2 (a = 10 mm)
'--------------------------
Function gepModel2(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C9 As Double = -
673.8718720734
   Const G1C2 As Double = -
9.989001
   Const G1C8 As Double =
5.988006
   Const G2C1 As Double = -
3.5964
   Const G2C9 As Double =
4.995
   Const G3C9 As Double =
3.003
   Dim y As Double
   y = 0#
   y = (((((G1C2 * G1C8) /
(d(3) ^ 2)) ^ 3) * ((G1C9 ^ 3) ^
2)) ^ 2)
   y = y + Exp(((((d(2) * d(2))
* d(3)) / (d(0) ^ 3)) * ((G2C1 *
G2C9) ^ 3)))
   y = y + ((Exp(((d(3) / d(0)) ^
2)) * ((Exp(G3C9) ^ 3) ^ 2)) ^
3)
```

```
   Const SLOPE As Double = 9.92492065443564E-54
   Const INTERCEPT As Double = -8.17014727523391
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel2 = probabilityOne
End Function

' State 3 (a = 90 mm)
'--------------------------
Function gepModel3(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C7 As Double = -9#
   Const G1C0 As Double = -4#
   Const G2C4 As Double = -3#
   Const G2C8 As Double = -5#
   Const G3C5 As Double = -3#
   Const G3C7 As Double = 6#
   Const G3C6 As Double = 8#
   Dim y As Double
   y = 0#
   y = (Exp(G1C0) + G1C7)
   y = y + Sqr(Exp((((d(0) / d(2)) - (G2C4 + G2C8)) / ((d(2) / d(0)) ^ 2))))
   y = y + (((d(1) + G3C5) + (d(0) - G3C7)) / ((G3C6 ^ 2) + (d(3) - d(3))))

   Const SLOPE As Double = 9.93025946235413E-79
   Const INTERCEPT As Double = -7.93832502139603
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel3 = probabilityOne
End Function

' State 4 (a = 170 mm)
'--------------------------
Function gepModel4(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C9 As Double = -5#
   Const G2C0 As Double = 9#
   Const G2C8 As Double = -6#
   Const G3C9 As Double = 2#
   Const G3C0 As Double = -6#
   Const G3C3 As Double = -7.007
   Dim y As Double
   y = 0#
   y = Exp(((d(1) * Exp(d(3))) * (G1C9 * (d(1) - G1C9))))
   y = y + (((d(1) + d(0)) * (G2C0 ^ 3)) * ((d(1) - G2C8) + (d(0) * d(1))))
   y = y + ((((G3C3 ^ 2) * d(2)) * (G3C9 + d(2))) * ((d(3) - d(0)) * (G3C0 + d(2))))
   Const SLOPE As Double = 2.34283961720364E-06
   Const INTERCEPT As Double = -7.96083513694224
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel4 = probabilityOne
End Function


' State 5 (a = 250 mm)
'--------------------------
Function gepModel5(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C7 As Double = -8#
   Const G1C0 As Double = 7#
   Const G1C1 As Double = 2.002
   Const G2C9 As Double = 9.99
   Const G3C4 As Double = 7.07
   Const G3C2 As Double = -2#

   Dim y As Double
   y = 0#
   y = ((G1C7 * ((G1C1 / d(1)) * d(3))) * ((G1C7 - d(0)) * (G1C0 / d(2))))
   y = y + ((((d(1) / d(1)) - d(0)) * (d(0) - d(2))) / ((d(3) ^ 2) * (d(3) / G2C9)))
   y = y + (((((G3C2 / d(1)) - G3C4) ^ 2) / (d(3) + (d(1) + d(0)))) ^ 2)
   Const SLOPE As Double = 3.72391104592556E-05
   Const INTERCEPT As Double = -7.83849444890817
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel5 = probabilityOne
End Function

' State 6 (a = 330 mm)
'--------------------------
Function gepModel6(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C4 As Double = -5.005
   Const G1C0 As Double = -1#
   Const G1C3 As Double = -2#
   Const G1C6 As Double = -10#
   Const G2C1 As Double = 1#
   Const G2C8 As Double = 8.991
   Const G3C4 As Double = 6#
   Const G3C8 As Double = -3#
   Const G3C0 As Double = 9#
   Dim y As Double
   y = 0#
   y = ((Sqr((d(3) - G1C6)) + (d(1) + d(1))) - ((d(2) - G1C4) * (G1C0 * G1C3)))
   y = y + (Exp(((G2C1 - d(2)) - (d(0) * d(3)))) + Exp(((G2C8 * G2C8) / d(0))))
```

```
    y = y + ((((d(1) + G3C0) - G3C4) * (G3C8 / d(1))) - ((d(1) - d(1)) - (G3C0 + G3C8)))
    Const SLOPE As Double = 1.06766764303909E-28
    Const INTERCEPT As Double = -8.41298424276477
    Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
    gepModel6 = probabilityOne
End Function

' State 7 (a = 410 mm)
'-------------------------
Function gepModel7(ByRef r As Range) As Double
    Dim n As Long
    Dim d(0 To 3) As Double
    For n = 0 To 3
d(n) = CDbl(r(n + 1))
    Next
    Const G1C4 As Double = 3.03303
    Const G2C9 As Double = -3.64193677659767
    Const G3C1 As Double = -8.189181
    Const G3C9 As Double = -3#
    Dim y As Double
    y = 0#
    y = ((((G1C4 ^ 4) ^ 3) ^ 4) - (((d(0) ^ 4) + d(0)) * (d(3) ^ 4)))
    y = y + (((Exp(d(0)) * d(0)) + ((G2C9 + d(0)) / (d(2) ^ 4))) ^ 3)
    y = y + (d(3) - Exp((((d(2) ^ 3) + (G3C9 ^ 2)) - (d(1) + G3C1))))
    Const SLOPE As Double = 2.5511686740335E-26
    Const INTERCEPT As Double = -7.06653202693616
    Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
    gepModel7 = probabilityOne
End Function
```

A.4.: The VBA Codes for 8 Organized Sensors

```
' State 1 (Intact, a= 0 mm)
'-------------------------
Function gepModel1(ByRef r As Range) As Double
    Dim n As Long
    Dim d(0 To 3) As Double
    For n = 0 To 3
d(n) = CDbl(r(n + 1))
    Next
    Const G1C6 As Double = -3.988011996
    Const G1C5 As Double = -0.064138015508733
    Const G2C5 As Double = -4.8357485514243
    Const G2C1 As Double = -1.07594840239699
    Const G2C2 As Double = -7.3134014033839
    Const G2C4 As Double = 7.82547315603315
    Const G2C9 As Double = 7.96731376567705E-03
    Const G2C8 As Double = -5.14143681709626
    Const G3C0 As Double = 1.000996997003
    Const G3C4 As Double = 0.749237501255573
    Const G3C5 As Double = 2.32623176257729
    Dim y As Double
    y = 0#
    y = (d(1) / ((Exp((d(3) - d(1))) / ((d(2) + G1C5) - G1C6)) + ((d(0) / d(1)) * (d(1) - d(3)))))
    y = y + Exp((d(1) / ((((d(1) + G2C1) ^ 3) * G2C5) * ((d(3) - Exp(G2C2)) / ((G2C2 / G2C4) - (G2C9 - G2C8))))))
    y = y + Exp((d(2) / (d(1) - (d(3) * (((((d(2) / d(1)) / G3C4) + (G3C5 * d(0))) * d(3)) + G3C0)))))

    Const SLOPE As Double = 9.03489419561824E-02
    Const INTERCEPT As Double = -8.04669558957269
    Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
    gepModel1 = probabilityOne
End Function

' State 2 (a = 10 mm)
'-------------------------
Function gepModel2(ByRef r As Range) As Double
    Dim n As Long
    Dim d(0 To 3) As Double
    For n = 0 To 3
d(n) = CDbl(r(n + 1))
    Next
    Const G1C0 As Double = -6.06606
    Const G1C1 As Double = 10.201
    Const G1C6 As Double = 3.003
    Const G2C2 As Double = 9.791199
    Const G2C3 As Double = -9.8802099
    Const G2C8 As Double = -7.92
    Const G3C7 As Double = 9.22787331553004E-02
    Dim y As Double
    y = 0#
    y = ((((((d(3) ^ 4) ^ 2) * ((G1C6 ^ 2) / d(1))) + Sqr(Exp(G1C1))) - (G1C0 * ((G1C0 ^ 3) / (d(3) + d(2))))) ^ 4)
    y = y + (((((d(1) - d(0)) + (d(0) * G2C8)) * G2C2) * (((d(2) + d(2)) - d(1)) ^ 3)) * (Exp((d(2) - d(0))) * ((G2C3 ^ 3) ^ 4)))
    y = y + ((((Exp(((d(2) * d(2)) + (d(1) / d(3)))) * ((d(2) + d(3)) ^ 2)) - Exp(((d(1) ^ 3) - (G3C7 / d(1))))) ^ 4) * d(0))
    Const SLOPE As Double = 1.83957943716362E-16
    Const INTERCEPT As Double = -5.34814047260968
    Dim probabilityOne As Double
```

```
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel2 = probabilityOne
End Function

' State 3 (a = 90 mm)
'-------------------------
Function gepModel3(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C9 As Double = -9.009
   Const G1C7 As Double = -1.112111
   Const G3C9 As Double = 5.04495
   Const G3C7 As Double = 4#
   Const G3C0 As Double = -2.002
   Dim y As Double
   y = 0#
   y = (d(1) / ((Exp(Exp(d(1))) / (G1C9 - (d(3) * G1C7))) * (((G1C9 - d(0)) - (d(3) + d(1))) - (G1C9 - d(1)))))
   y = y + (d(0) * d(2))
   y = y + (d(2) - (G3C9 - ((Exp(((G3C7 + G3C0) / (d(0) + d(1)))) - d(2)) / (Exp((d(0) + G3C7)) ^ 3))))
   Const SLOPE As Double = 1.29797282256239
   Const INTERCEPT As Double = -4.90858192971192
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel3 = probabilityOne
End Function

' State 4 (a = 170 mm)
'-------------------------
Function gepModel4(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C0 As Double = -4.03596
   Const G1C5 As Double = 9.09
   Const G2C7 As Double = 10#
   Const G2C1 As Double = 2.02
   Const G3C3 As Double = 8#
   Const G3C0 As Double = 8#
   Const G3C5 As Double = 8.08
   Const G3C2 As Double = 8.712
   Const G3C1 As Double = 4#
   Const G3C8 As Double = -27#
   Dim y As Double
   y = 0#
   y = ((((((d(2) * d(1)) - d(0)) - (d(3) * d(1))) + ((d(0) * d(0)) - (d(1) * d(1)))) - d(1)) + ((((G1C0 - G1C5) ^ 4) ^ 2) ^ 3))
   y = y + (d(3) + (((((((d(1) - G2C7) - (G2C1 ^ 4)) * ((d(3) ^ 4) - G2C7)) ^ 3) - (G2C7 ^ 2)) ^ 4) / d(1)))
   y = y + (((((G3C1 ^ 4) ^ 4) ^ 4) + (((G3C5 - G3C8) - G3C0) * d(0))) + (((G3C5 * G3C0) + G3C3) - Exp((d(0) * G3C2))))
   Const SLOPE As Double = 3.0258972630069E-31
   Const INTERCEPT As Double = -102965953.285512
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel4 = probabilityOne
End Function


' State 5 (a = 250 mm)
'-------------------------
Function gepModel5(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C5 As Double = -8#
   Const G2C8 As Double = 5#
   Const G2C2 As Double = -6#
   Const G3C7 As Double = 4#
   Const G3C0 As Double = -5#
   Dim y As Double
   y = 0#
   y = (((d(0) + d(1)) + (d(1) / d(0))) * (d(3) + G1C5))
   y = y + (((d(1) + Exp(((d(0) * G2C2) * (G2C2 / d(2))))) ^ 3) + ((Sqr(G2C8) - ((d(1) + d(3)) * (d(0) ^ 2))) ^ 2))
   y = y + ((((((G3C7 ^ 3) * (G3C7 - d(0))) * ((G3C7 * d(2)) + (d(1) - G3C0))) * (G3C7 / d(3))) ^ 4) + d(1))
   Const SLOPE As Double = 3.79189016352223E-33
   Const INTERCEPT As Double = -8.65825786799137
   Dim probabilityOne As Double
probabilityOne = 1# / (1# + Exp(-(SLOPE * y + INTERCEPT)))
   gepModel5 = probabilityOne
End Function

' State 6 (a = 330 mm)
'-------------------------
Function gepModel6(ByRef r As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C6 As Double = 2.969703
   Const G2C1 As Double = -3.996
   Const G3C9 As Double = 5.1005
   Const G3C0 As Double = 3.999996
   Const G3C8 As Double = 0#
   Dim y As Double
   y = 0#
   y = (((((((d(0) - d(0)) + (d(2) ^ 4)) - d(0)) * (((G1C6 + d(2)) ^ 3) ^ 4)) ^ 2) ^ 2) ^ 2)
   y = y + ((((((((d(1) / d(2)) ^ 4) - ((d(2) * d(0)) * d(2))) ^ 2) / (((d(0) * d(3)) + (G2C1 * d(0))) ^ 4)) ^ 2) ^ 2) ^ 2)
   y = y + ((((Exp(d(2)) - G3C9) / (G3C0 - d(0))) +
```

```
((d(0) * d(3)) ^ 3)) + (((d(0) ^
2) * Exp(d(2))) + ((G3C8 *
d(1)) + (d(3) ^ 2))))
   Const SLOPE As Double =
1.84854638162239E-48
   Const INTERCEPT As
Double = -5.53401307008457
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel6 =
probabilityOne
End Function


' State 7 (a = 410 mm)
'--------------------------

Function gepModel7(ByRef r
As Range) As Double
   Dim n As Long
   Dim d(0 To 3) As Double
   For n = 0 To 3
d(n) = CDbl(r(n + 1))
   Next
   Const G1C8 As Double = -
7#
   Const G1C5 As Double = -
8#
   Const G1C7 As Double =
10.1101
   Const G1C0 As Double = 4#
   Const G1C1 As Double = -
1#
   Const G2C7 As Double = 3#
   Const G3C1 As Double =
1000#
   Const G3C3 As Double =
8.8
   Const G3C0 As Double = 1#
   Const G3C5 As Double = 7#
   Const G3C7 As Double = -
7#
   Dim y As Double
   y = 0#
   y = ((G1C8 * ((((G1C5 *
G1C7) * G1C5) + ((G1C0 ^ 4)
^ 2)) - (Exp((d(1) + d(0))) *
((G1C1 ^ 3) * d(2))))) ^ 3)
   y = y + Exp((((((d(0) ^ 3) +
(d(1) + d(3))) * Exp(d(0))) ^ 2)
+ Exp((((G2C7 + d(1)) + (d(1)
+ d(0))) - d(3)))))
y = y + ((G3C1 ^ 3) + (((((d(3)
^ 4) * (G3C5 * G3C7)) *
G3C3) ^ 3) - ((Exp((G3C3 -
d(2))) + (G3C0 + d(0))) ^ 2)))
   Const SLOPE As Double =
5.6248130197281E-13
   Const INTERCEPT As
Double = 55921.7314988659
   Dim probabilityOne As
Double
probabilityOne = 1# / (1# +
Exp(-(SLOPE * y +
INTERCEPT)))
   gepModel7 =
probabilityOne
End Function
```

BIBLIOGRAPHY

BIBLIOGRAPHY

AASHTO, Standard Specifications for Highway Bridges. American Association of State Highway and Transportation Officials (AASHTO), 16th Ed. Ishington, D.C, 1996.

Adeli H., Panakkat A., A probabilistic neural network for earthquake magnitude prediction. Neural Networks, 2009, 22, 1018-1024.

Adewuyi A.P., Wu Z., Serker N.H.M.K., Assessment of vibration-based damage identification methods using displacement and distributed strain measurements. Structural Health Monitoring, 2009,8, 443–461.

Alavi A. H., M. Ameri, A. H. Gandomi, M. R. Mirzahosseini, Formulation of flow number of asphalt mixes using a hybrid computational method. Construction and Building Materials, 2011, 25(3), 1338-1355.

Alavi A.H., A.H. Gandomi, H. Chahkandi Nejad, A. Mollahasani, A. Rashed, Design equations for prediction of pressuremeter soil deformation moduli utilizing expression programming systems. Neural Computing & Applications 2013, 23(6), 1771-1786.

Alavi A.H., Gandomi A.H., A Robust Data Mining Approach for Formulation of Geotechnical Engineering Systems. Engineering Computations 2011, 28(3), 242-274.

Alavi A.H., Hasni H., Lajnef N., Chatti K., Faridazar F., Continuous Health Monitoring of Pavement Systems Using Smart Sensing Technology. Construction and Building Materials, 2016c, 114, 719–736.

Alavi A.H., Hasni H., Lajnef N., Chatti K., Faridazar F. An Intelligent Structural Damage Detection Approach Based on Self-Powered Wireless Sensor Data. Automation in Construction, 2016a, 62, 24–44.

Alavi A.H., Hasni H., Lajnef N., Chatti K., Faridazar F., Damage Detection Using Self-Powered Wireless Sensor Data: An Evolutionary Approach. Measurement 2016b, 82, 254–283.

Al-Qadi I.L., Wang H., Pavement Damage Due to Different Tire and Loading Configurations on Secondary Roads. NEXTRANS University Transportation Center, West Lafayette, Ind, 2010.

Al-Qadi I.L., Loulizi A., Elseifi M., Lahouar S., The Virginia smart road: The impact of pavement instrumentation on understanding pavement performance. Journal of the Association of Asphalt Paving Technologists, 2004, 73, 427–465.

ASTM E1820-09, Standard test method for measurement of fracture toughness. American Society for Testing and Materials, USA, 2009.

Badr A., Karlaftis A.G., Duration model estimation for pavement rehabilitation and service life. Advances and Applications in Statistics, 2012, 31, 1-19.

Banzhaf W., Nordin P., Keller R., Francone F., Genetic programming: An introduction on the automatic evolution of computer programs and its application. Morgan Kaufmann, 1998.

Bhargava A., Roddis W.M.K., Finite Element Analysis of Fatigue Prone Details of the Tuttle Creek Bridge, K-TRAN: KS-07-5, Final Report, Kansas Department of Transportation, Topeka KS, 2007.

Burton A.R., Minegishi K., Kurata M., Lynch J.P., Free-standing carbon nanotube composite sensing skin for distributed strain sensing in structures. Proceeding of SPIE 9061, 2014, 906123.

Chakrabartty S., Asynchronous Event-Based Self-Powering, Computation, and Data Logging. Chapter 14, N. Elvin, A. Erturk (Eds.) Advances in Energy Harvesting Methods, Springer Science+Business Media New York, 2013.

Coello C.C.A., Lamont G.B., Van Veldhuizen D.A., Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation. 2nd ed. Springer, New York, 2007.

Doebling S.W., Farrar C.R., Prime M.B., Shevitz D.W., Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in Their Vibration Characteristics: A Literature Review. Technical Report LA-13070-MS, Los Alamos National Laboratory, NM, 1996.

Elewa M.A., Influence of Secondary Components on the Serviceability of Steel Girder Highway Bridges. Ph.D. Dissertation, Michigan State University, East Lansing, MI, 2004.

Elkordy M.F, Chang K.C., Lee G.C., Neural networks trained by analytically simulated damage states. Journal of Computing in Civil Engineering (ASCE), 1993, 7(2), 130 - 145.

Elvin N.G., Lajnef N., Elvin A.A., Feasibility of structural monitoring with vibration powered sensors. Smart Materials and Structures, 2006, 15(4), 977-986,

Elvin N., Elvin A., Choi D. H., A self-powered damage detection sensor. Journal of Strain Analysis, 2003, 38(2), 115 -124.

Elvin N., Elvin A., Spector M., A self-powered mechanical strain energy sensor. Smart Materials and Structures, 2001, 10, 293–299.

Elvin N., Leung C.K.Y., A fast iterative boundary element technique for solving closed crack problems. Engineering Fracture Mechanics, 1999, 63(5), 631-650.

Fan W., Qiao P., Vibration-based damage identification methods: a review and comparative study. Structural Health Monitoring, 2011, 10(1), 83–111.

Farrar C. R., Park G., Allen D.W., Todd M., Sensor Network Paradigms for Structural Health Monitoring. Structural Health Monitoring, 2006, 13, 210-255.

Farrar C.R., Historical Overview of Structural Health Monitoring. Lecture Notes on Structural Health Monitoring Using Statistical Pattern Recognition, Los Alamos Dynamics, Los Alamos, NM, 2001.

Ferreira C., Gene expression programming: a new adaptive algorithm for solving problems. Complex Systems, 2001, 13, 87–129.

FHWA, Status of the Nation's highways, bridges, and transit: 2004 Conditions and performance report to congress. Federal Highway Administration (FHWA), Ishington, DC, 2004. http://www.fhwa.dot.gov/policy/2004cpr/index.htm (Last Accessed: Oct. 19, 2015).

Fisher J.W., Barthelemy B. M., Mertz D. R., Edinger J.A., National Cooperative Highway Research Program Report 227: Fatigue Behavior of Full-Scale Welded Bridge Attachments. Transportation Research Board, National Research Council, Ishington, D.C, 1980.

Fisher, J.W., Menzemer C.C., Fatigue Cracking in Welded Steel Bridges. Transportation Research Record, 1990, 1282, 111-117.

Fisher, J.W., Fatigue and Fracture in Steel Bridges: case Studies. John Wiley & Sons. Inc., 1984.

Flood I., Christophilos P., Modeling construction processes using artificial neural networks. Automation in Construction, 1996, 4(4), 307-320.

Fogel L.J., Owens A.J., Walsh M.J., Artificial Intelligence through Simulated Evolution. Wiley, New York, 1996.

Gandomi A.H., Alavi A.H., Mirzahosseini M.R., Moghadas Nejad F., Nonlinear Genetic-Based Models for Prediction of Flow Number of Asphalt Mixtures. ASCE Journal of Materials in Civil Engineering, 2011, 23, 1-18.

Gang N., Yang B.S., Pechta M., Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. Reliability Engineering & System Safety, 2010, 95(7), 786–796.

Ghuzlan, K.A., Carpenter, H. Fatigue damage analysis in asphalt concrete mixtures using the dissipated energy approach. Canadian Journal of Civil Engineering, 2006, 33(7), 890-901.

Goel, A., Das, A. Nondestructive testing of asphalt pavements for structural condition evaluation: A state of the art. Nondestructive Testing and Evaluation. 2008, 23(2), 121–140.

Goh A.T.C., Probabilistic neural network for evaluating seismic liquefaction potential. Canadian Geotechnical Journal, 2002, 39, 219-232.

Grosse C., McLaskey G., Bachmaier S., Glaser S.D., Krüger M., A hybrid wireless sensor network for acoustic emission testing in SHM. Proceeding of SPIE 6932, 2008.

Hafiang W., Fatigue performance evaluation of Westrack asphalt mixtures based on viscoelastic analysis of indirect tensile test. PhD dissertation, North Carolina State University, 2001.

Hall D.L., Llinas J., Introduction to multisensor data fusion. Handbook of multisensor data fusion, 1st ed., Boca Raton: CRC press pp.1-15, 2001.

Hao S., I-35W Bridge Collapse. Journal of Bridge Engineering ASCE, 2010, 608- 614.

Harvey D., Todd M., Automated Selection of Damage Detection Features by Genetic Programming. Proceeding of SPIE 8695, Health Monitoring of Structural and Biological Systems2013, 86950J, San Diego, California, USA, 2013.

Holt R., Hartmann J., Adequacy of the U10 & L11 Gusset Plate Designs for the Minnesota Bridge No. 9340 (I-35 over the Mississippi River). Interim Report, Federal Highway Administration, 2008.

Hou Z., Noori M., Amand R.S., Wavelet-based approach for structural damage detection. American Society of Civil Engineers, Journal of Engineering Mechanics, 2000, 126(7), 677–683.

Huang C., Lajnef N., Chakrabartty S., Self-calibration and characterization of self-powered floating-gate usage monitors with single electron per second operational limit. IEEE Transactions on Biomedical Circuits and Systems, 2010, 57, 556-567.

Huang N.E., Shen Z., Long S.R., A new view of nonlinear water waves: the Hilbert spectrum. Annual Review of Fluid Mechanics 1999, 31, 417–475.

Hung W.C., Hwang C., Liou J.C., Lin Y.S., Yang H.L., Modeling aquifer-system compaction and predicting land subsidence in central Taiwan. Engineering Geology, 2012, 147–148, 78-90.

Jiang S.F., Fu C., Zhang C., A hybrid data-fusion system using modal data and probabilistic neural network for damage detection. Advances in Engineering Software, 2011, 42, 368–374.

Juntunen D., Study of Michigan’s Link Plate and Pin Assembly. Michigan Department of Transportation (MDOT). Research Report No. R-13581998, 1998.

Kohavi R., Foster P., Glossary of terms. Machine Learning, 1998, 30, 271-274.

Koutsopoulos H., Downey A. Primitive based classification of pavement cracking images. Journal of Transportation Engineering (ASCE), 1993, 3(402), 402–418.

Koza J.R., Genetic programming, on the programming of computers by means of natural selection. MIT Press, Cambridge (MA), 1992.

Lajnef N., Chatti K., Chakrabartty S., Rhimi M., Sarkar P., Smart Pavement Monitoring System. Report: FHWA-HRT-12-072, Federal Highway Administration (FHWA), Ishington, DC, 2013.

Lajnef N., Rhimi M., Chatti K., Mhamdi L., Toward an Integrated Smart Sensing System and Data Interpretation Techniques for Pavement Fatigue Monitoring. Computer-Aided Civil and Infrastructure Engineering, 2011, 26, 513–523.

Lajnef N., Chakrabartty S., Elvin N., A Piezo-powered Floating-gate Sensor Array for Long-term Fatigue Monitoring in Biomechanical Implants. IEEE Transaction on Biomedical Circuits and Systems, 2008, 2(3), 164-172.

Lee J.J., Yun C.B., Damage localization for bridges using probabilistic neural networks. KSCE Journal of Civil Engineering, 2007, 11, 111–120.

Lee J.J., Lee J.W., Yi J.H., Yun C.B., Jung H.Y., Neural networks-based damage detection for bridges considering errors in baseline finite element models. Journal of Sound and Vibration, 2005, 280, 555–578

Li P., Olmi C., Song G., (Hybrid Piezo-Based Wireless Sensor Networks for Civil Structural Health Monitoring. Earth and Space, 2010, 2405-2411.

Liao M., Okazaki T., A Computational Study of the I-35W Bridge Collapse. Final Report, CTS Project # 2008049, Center for Transportation Studies, University of Minnesota, Minneapolis, MN, 2009.

Liao M., Okazaki T., Ballarini R., Schultz A.E., Galambos T.V., Nonlinear Finite-Element Analysis of Critical Gusset Plates in the I-35W Bridge in Minnesota. Journal of Structural Engineering (ASCE), 2011, 59-69.

Lu W., Teng J., Cui Y., Damage Identification for Large Span Structure Based on Multiscale Inputs to Artificial Neural Networks. The Scientific World Journal, 2014, Article ID 540806.

Lynch J.P., K.J. Loh, A summary review of wireless sensors and sensor networks for structural health monitoring. Shock and Vibration Digest, 2006, 38, 91-128.

Lynch, J. P., Overview of Wireless Sensors for Real-time Health Monitoring of Civil Structures. Proceedings of the 4th International Workshop on Structural Control, New York, NY, 2004, 189–194.

Lynch J.P., Design of a Wireless Active Sensing Unit for Localized Structural Health Monitoring, Journal of Structural Control and Health Monitoring, 2005,12(3-4), 405–423.

Lynch J. P., Sundararajan A., Law K. H., Carryer E., Farrar C. R., Sohn H., Allen D.W., Nadler B., Wait J. R., Design and Performance Validation of a Wireless Sensing Unit for Structural Health Monitoring Applications. Structural Engineering and Mechanics, 2004, 17(3), 393–408.

Lynch J.P., Sundararajan A., Law K.H., Kiremidjian A.S., Kenny T., Carryer E., Embedment of Structural Monitoring Algorithms in a Wireless Sensing Unit. Journal of Structural Engineering and Mechanics, 2003d, 15(3), 285–297.

Malekzadeh M., Atia G., Catbas F.N., Performance-based structural health monitoring through an innovative hybrid data interpretation framework. Journal of Civil Structural Health Monitoring, 2015, 5(3), 287-305.

Marasteanu M.O., Buttlar W., Bahia H., Williams C., Moon K.H., Teshale E.Z., Falchetto A.C., Turos M., Investigation of Low Temperature Cracking in Asphalt Pavements, National Pooled Fund Study -Phase II. Final Report 2012-232012, University of Minnesota, 2012.

Masri S.F., Chassiakos A.G, Caughey T.K., Identification of nonlinear dynamic systems using neural networks. Journal of Applied Mechanics, Transactions ASME, 1993, 60, 123-133.

Mills J.P., Newton I., Peirson G.C., Pavement deformation monitoring in a rolling load facility. The Photogrammetric Record Journal, 2001, 17(97), 7–24.

Moes N., Dolbow J., Belytschko T., A finite element method for crack growth without remeshing. International Journal for Numerical Methods in Engineering, 1999, 46(1), 131–150.

Mohajeri M.H., Manning P.J., ARIA: An operating system of pavement distress diagnosis by image processing. Transportation Research Record 1311, Transportation Research Board, Ishington, DC, 1991, 120–130.

Ni Y.Q., Jiang S.F., Ko J.M., Application of adaptive probabilistic neural network to damage detection of Tsing Ma suspension bridge. Proceeding of SPIE 4337, Health Monitoring and Management of Civil Infrastructure Systems, 2001.

NTSB, Structural and local failure study of gusset plate in minneapolis bridge collapse. Report - NTSBC070010, National Transportation Safety Board (NTSB), Ishington, DC, 2008.

Olson S.E., DeSimio M., Derriso M.M., Fastener Damage Estimation in a Square Aluminum Plate. Structural Health Monitoring, 2006, 5, 173–183.

Oltean M., C. Grosşan, A comparison of several linear genetic programming techniques. Advances in Complex Systems, 2003, 14(4), 1–29.

Plati C., Georgiou P., Loizos A., Use of infrared thermography for assessing HMA paving and compaction. Transportation Research Part C: Emerging Technologies, 2014, 46, 192-208.

Poli R., Langdon W.B., McPhee N.F., Koza J.R., Genetic programming: An introductory tutorial and a survey of techniques and applications. Technical report [CES-475], University of Essex, UK, 2007.

Potter J. F., Mayhew H.C., Mayo A.P., Instrumentation of the full scale experiment on A1 trunk road at Conington, Huntingdonshire. Transport Research Laboratory (Road Research Laboratory), Wokingham, Berkshire, U.K., 1969.

Quadri S.A., Sidek O., Factors Affecting Data Fusion Performance in an Inertial Measurement Unit. Journal of Control Engineering and Technology, 2013, 3, 107-110.

Quek S.T., Wang Q., Zhang L., Ong K.H., Practical issues in the detection of damage inbeams using wavelets. Smart Materials and Structures, 2001, 10, 1009–1017.

Rahimi M., Shah H., Sukhatme G.S., Heideman J., Estrin D., Studying the Feasibility of Energy Harvesting in a Mobile Sensor Network. Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei. Taiwan, 2003, 19-24.

Rhimi M., Lajnef N., Chatti K., Faridazar F., A Self-powered Sensing System for Continuous Fatigue Monitoring of In-service Pavements. International Journal of Pavement Research and Technology, 2012, 5(5), 303-310.

Rollings R. S., Pittman D.W., Field instrumentation and performance monitoring of rigid pavements. Journal of Transportation Engineering (ASCE), 1992, 361–370.

Rosenstrauch P.L., Sanayei M., Brenner B.R., Capacity Analysis of Gusset Plate Connections Using the Whitmore, Block Shear, Global Section Shear, and Finite Element Methods. Engineering Structures, 2013, 48, 543–557.

Roundy S., Wright P.K., Rabaey J., A Study of Low Level Vibrations as a Power Source for Wireless Sensor Nodes. Computer Communications, 2002, 26, 1131-1144.

Schwefel H.P., Evolution strategy and numerical optimization, Dissertation. Technical University of Berlin, Berlin, Germany, 1975.

Shahin M. A., Jaksa M. B., Neural network prediction of pullout capacity of marquee ground anchors. Computers and Geotechnics, 2005, 32(3), 153-163.

Shi Z.Y., Law S.S., Zhang L.M., Structural damage localization from modal strain energy change. Journal of Sound and Vibration, 1998, 218(5), 825–844.

Singh N.G., Joshi M., Optimization of location and number of sensors for structural health monitoring using genetic algorithm. Materials Science Forum, 2009, 33, 359-67.

Sirohi J., Chopra I., Fundamental Understanding of Piezoelectric Strain Sensors. Journal of Intelligent Material Systems and Structures, 2001, 11, 246-257.

Sohn H., Farrar C. R., Hemez F.M., Czarnecki J.J., Shunk D.D., Stinemates D.W., Nadler B.R., A Review of Structural Health Monitoring Literature: 1996–2001. Report Number LA-13976-MS, Los Alamos National Laboratory, Los Alamos, NM, 2004.

Specht D., Probabilistic Neural Networks. Neural Networks, 1990, 3, 109-118.

Spencer B.F.Jr., Opportunities and Challenges for Smart Sensing Technology. Proceedings of the International Conference on Structural Health Monitoring and Intelligent Infrastructure, Tokyo, Japan, 2003, 65–71.

Spencer B.F.Jr., Ruiz-Sandoval M.E., Kurata N., Smart Sensing Technology: Opportunities and Challenges. Journal of Structural Control and Health Monitoring, 2004, 11(4), 349–368.

Staszewski W.J., Boller C., Tomlinson G.R., Health Monitoring of Aerospace Structures: Smart Sensor Technologies and Signal Processing. Wiley, New York, 2004.

Stetson K.A., Harrison I.R., Redesign of structural vibration modes by finite element inverse perturbation. Journal of Engineering for Power, 1981, 103, 319–325.

Su Z., Wang X., Cheng L., Yu L., Chen Z., On selection of data fusion schemes for structural damage evaluation. Structural Health Monitoring, 2009, 8(3), 223–241.

Sundaram B.A., K. Ravisankar, R. Senthil, S. Parivalla, Wireless sensors for structural health monitoring and damage detection techniques. Current Science, 2013, 104(11), 1496-1505.

Szewezyk P., Hajela P., Damage detection in structures based on feature - sensitive neural networks. Journal of Computing in Civil Engineering (ASCE), 1994, 8(2), 163-178.

Unal M., Onat M., Demetgul M., Kucuk H., Fault diagnosis of rolling bearings using a genetic algorithm optimized neural network. Measurement, 2014, 58, 187-196.

Watters D.G., Jayaweera P., Bahr A.J., Huestis D.L., Priyantha N., Meline R., Reis R., Parks D., Smart Pebble: Wireless Sensors for Structural Health Monitoring of Bridge Decks. Proceedings of SPIE 5057, 2003.

Weise T., Global Optimization Algorithms – Theory and Application. Germany: it-weise.de (self-published). [Online]. Available: http://www.it-weise.de (2009)

Wu X., Gahboussi J., Garrett J.H., Use of neural networks in detection of structural damage. Computers & Structures, 1992, 42(4), 649 - 659.

Xue W., Wang L., Wang D., Druta C., Pavement Health Monitoring System Based on an Embedded Sensing Network. Journal of Materials in Civil Engineering (ASCE), 2014, 26(10), 04014072.

Yan B., Miyamoto A., Application of probabilistic neural network and static test data to the classification of bridge damage patterns. Proceedings of SPIE 5057, 2003, 606-617.

Yoo P.J., Al-Qadi I.L., Effect of Transient Dynamic Loading on Flexible Pavements. Transportation Research Record, Transportation Research Board of the National Academies, Ishington, DC, 2007, 129-140.

Youn H., Gu Z., Predicting Korean lodging firm failures: An artificial neural network model along with a logistic regression model. International Journal of Hospitality Management, 2010, 29(1), 120–127.

Yun C.B., Min J., Smart Sensing, Monitoring, and Damage Detection for Civil Infrastructures. KSCE J Civil Eng, 2011, 15(1), 1-14.

Zein-Sabatto S., Mikhail M., Bodruzzaman M., DeSimio M., Information and Decision Fusion Systems for Aircraft Structural Health Monitoring. Proceedings of IEEE, Nashville, TN, 2011, pp. 395-400.

Zhao J., Ivan J.N., DeWolf J.T., Structural Damage Detection Using Artificial Neural Networks. Journal of Infrastructure Systems (ASCE), 1998, 4(3), 93-101.

Zhao Y., Roddis W.M.K., Fatigue Prone Steel Bridge Details: Investigation and Recommended Repairs, K-TRAN: KU-99-2. Final Report, Kansas Department of Transportation, Topeka KS, 2004.

Zong Z.H., Wang T.L., Huang D.Z., State-of-the-art Report of Bridge Health Monitoring. Journal of Fuzhou University (Natural Science), 2002, 30, 127–152.

Zou Y., Tong L., Steven G.P., Vibration-based model-dependent damage (delamination) identification and health monitoring for composite structures-A review. Journal of Sound and Vibration, 2000, 230(2), 357–378.