
**POLITECNICO DI MILANO**
**FACULTY OF INDUSTRIAL ENGINEERING**
**MASTER THESIS IN MECHANICAL ENGINEERING**

# FATIGUE CRACK MONITORING OF HELICOPTER FUSELAGE THROUGH SENSOR NETWORK

**Alireza Alizadeh**

764184

*Supervisor:*

**Prof. M. Giglio**

Academic year 2013/2014

# Abstract

Developing a holistic approach for looking at the structure's integrity through real-time Structural Health Monitoring (SHM) defines the core of the project. This work argues the implementation of the real-time SHM for an isolated panel of the rear fuselage of a helicopter. This work initially presents an effective statistical approach to detect structural damage utilizing a novelty detection method based on multidimensional outlier analysis (OA). Going further into the hierarchy of damage identification, together with detection, quantification of the damage intensity and localization of the damage site are performed adopting a multi-layer perceptron (MLP) neural network structure.

Prior to the establishment of algorithms of any method (OA or MLP) a thorough study regarding the identification of the provided database is lunched. Concerning outlier analysis, the necessary database is acquired from experimental tests, consisting of sampled strain measures for both undamaged and damaged scenarios coming from a network of Fiber Bragg Grating (FBG) optical sensors. Due to the high expenditure of experimental test, comprising the costs of both operating equipments and the one-time-use test specimen, the only available experimental database regarding the damaged case is set to be a skin crack propagating from the center of the central bay. To extract the damage sensitive features out of the sensors readings, a normalization is performed where changes in sensor readings caused by damage are separated from those caused by varying operational and environmental conditions. Moreover, an initiative is taken to investigate the influence of possible localized stress variations (this can occur due to localized loads and temperature variations present within the panel) over the damage indices. This is of a great use as it firstly allows for a visual representation of the influence and secondly becomes of a use later in outlier analysis to check the flexibility and efficiency of the novelty detection algorithm.

In this thesis, the outlier analysis (OA) is conducted mainly for lowest level of fault detection (i.e. damage detection) so that the method is simply required to detect signal deviations from normal condition; i.e., the problem is the one of novelty detection. Therefore, the concept of discordancy from the statistical discipline of outlier analysis is used to identify signal deviance from the norm. Since the acquired database of the case study is multivariate, the discordancy test is performed by using Mahalanobis square distance measure to calculate the novelty values relevant to each observation of the database. In applying OA, the novelty values are finally compared against a threshold which allows to group novelty indices as novel or normal corresponding to damaged and undamaged cases respectively.

To develop a diagnostic unit that includes quantification and localization levels of a damage state, as well as the damage detection level, MLP neural networks are implemented. Here, the first challenge is to provide the ANNs with sufficiently large training database which can be considered as a good representative of all possible damage cases that can occur on the panel.

For that reason, a database simulated through a Finite Element (FE) model is used. When applying artificial neural networks as the performing algorithms of diagnostic unit, there are several parameters that are considered at each diagnostic level to optimize the performance of the diagnostic algorithm. The first parameter to be optimized is the structure of an ANN and in specific the number of hidden layers and nodes. Two other optimizing parameters are considered which are “introduction of additive Gaussian noise to the training database” and “optimal training database size”. Considering the classification algorithm for anomaly detection, the concept of “optimal noise” is applied upon redefining the state of damage, while the effect of additive noise in the performance of damage quantification an localization algorithms is simply carried out by checking the corresponding Root Mean Square Error (RMSE) in predictions. Finally, to verify the algorithms applicability to the real diagnostic system, a real experimental centre crack propagation is fed to all three layers of the diagnostic algorithm.

**This thesis is dedicated to my parents.**

*For their endless love, support and encouragement*

# Acknowledgment

I would never have been able to finish my dissertation without the guidance of my committee members, help from friends, and support from my family.

I would like to express my deepest gratitude to my supervisor, Prof. Marco Giglio, for his excellent guidance, patience, and providing me with an excellent atmosphere for doing my research.

I would like to thank Dr. Claudio Sbarufatti, for guiding my research for the past year and helping me to develop my knowledge in the field of structural health monitoring, through the numerous beneficial discussion, and also by providing me every needed resources which helped me to carry my research in best way possible.

I would like to thank Eng. Matteo Corbetta, Eng. Nima Allahverdizadeh and Eng. Sina Amiri, who as a friend were always willing to help and give their best suggestions. It would have been a lonely research office without them.

Finally, I would like to express my deepest gratitude to my parents and my brothers. They were always there, cheering me up and stood by me through the good times and bad.

# Contents

# Abbreviations

| | |
|---|---|
| ANN | Artificial Neural Network |
| Brg | Bragg |
| CI | Confidence Interval |
| CM | Condition Monitoring |
| COMT | Committee |
| DI | Damage Index |
| DP | Damage Prognosis |
| FAA | Federal Aviation Administration |
| FBG | Fiber Bragg Grating |
| FEM | Finite Element Model |
| HUMS | Health and Usage Monitoring |
| MSD | Mahalanobis Squared Distance |
| NDE | Non-Destructive Evaluation |
| OA | Outlier Analysis |
| OPNL | Optimal Noise Level |
| PFA | Probability of False Alarm |
| PME | Probability of Missing Event |
| PWD | Probability of wrong Decision |
| SHM | Structural Health Monitoring |
| SPS | Statistical Process Control |
| STD | Standard Deviation |
| Str | Stringer |
| TN | Training Noise |

# List of Figures

# List of Tables

# Chapter 1 Introduction

In the most general terms damage can be defined as changes to a system that adversely influence its current or future performance. Implicit in this definition is the concept that damage is not meaningful without a comparison between two different states of the system, one of which is assumed to represent the initial, and often undamaged, state. This work is focused on the study of damage identification in an aerospace structure. Therefore, the definition of damage will be limited to changes to the material and/or geometric properties of these systems, including changes to the boundary conditions and system connectivity, which adversely affect the current or future performance of these systems.

In terms of length scales, all damage begins at the material level. Although not necessarily universally accepted terminology, such damage is usually referred to as a defect or flaw and is present to some degree in all materials. Under appropriate loading scenarios the defects or flaws grow and coalesce at various rates to cause component and then system level damage. The term damage does not necessarily imply total loss of system functionality, but rather that the system is no longer operating in its optimal manner. As the damage grows it will reach a point where it affects the system operation to a point that is no longer acceptable to the user. This point is referred to as failure. In terms of time scales, damage can accumulate incrementally over long periods of time such as that associated with fatigue or corrosion damage accumulation. On relatively shorter time scales damage can also result from scheduled discrete events such as aircraft landings and from unscheduled discrete events such as unplanned emergency maneuvers that exceed design limits, enemy fire on a military aircraft.

The process of implementing a damage identification strategy for aerospace structures is referred to as Structural Health Monitoring **(SHM, Farrar and Worden, 2007)**. This process involves the observation of a structure or system over time using periodically spaced measurements, the extraction of damage-sensitive features from these measurements, and the statistical analysis of these features to determine the current state of system health. For long-term SHM, the output of this process is periodically updated information regarding the ability of the structure to continue to perform its intended function in light of the inevitable aging and damage accumulation resulting from its operational environments. Under an extreme event, such as those encountered by military systems, SHM is used for rapid condition screening. This screening is intended to provide, in near real-time, reliable information about system performance during such extreme events and the subsequent integrity of the system. Damage identification is carried out in conjunction with five closely related disciplines that include SHM, Condition Monitoring (CM), Non-Destructive Evaluation (NDE), Statistical Process Control (SPC), and Damage Prognosis (DP). Typically, SHM is associated with on-line, global damage identification in structural systems. CM is analogous to SHM, but addresses damage identification in rotating and reciprocating machinery. NDE is usually

carried out off-line in a local manner with some a priori knowledge of the damage location. SPC is process-based rather than structure-based and uses a variety of sensors to monitor changes in a process, one cause of which can result from structural damage. However, many of the statistical monitoring tools developed for SPC have been adapted to SHM applications. Once damage has been detected, DP is used to predict the remaining useful life of a system. There are no distinct boundaries between these various disciplines and in reality most damage detection methods applied to aerospace structures makes use of some combination of these disciplines.

## 1.1 A historic background of SHM

SHM definition given in the previous section contains two major elements of SHM: loads monitoring as a means to determine the true operational conditions of a structure and damage monitoring as the consequence of the operational conditions and a phenomenon subjected to a statistical nature. It is difficult to say when the origin of loads and damage monitoring has truly been but in aerospace many of the consequences have sadly been the result of serious accidents. The aeronautical world realized the necessity of loads monitoring definitely after the series of fatal crashes that occurred to the de Havilland Comet aircraft between 1952 and 1954. Monitoring systems were built that were based on monitoring the accelerations around the centre of gravity of an aircraft and that would allow exceedances at specific load levels to be monitored.

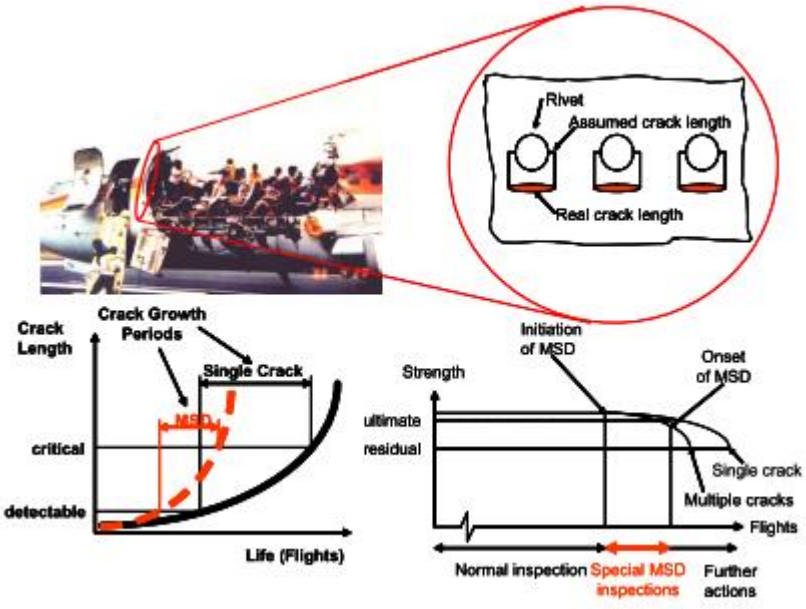



**Figure 1.1 The Aloha Airlines accident and resulting consequences**

The other event which triggered the issue of damage monitoring was the Aloha Airlines accident flight 231 (Figure 1.1), which occurred in April 1988. What had to be learned from

this was that the likelihood of cracking is increasing with an increasing age of an aircraft structure which resulted in the description of the phenomenon called Multi-Site Damage (MSD), which is the effect of a multitude of small cracks to be more sensitive to total fracture than the sum of all these cracks represented as the length of a single crack. This could be represented in general as an increase in crack propagation in a way it is schematically shown in Figure 1.2. The consequence of this has been the regulation set by the airworthiness authorities that ageing aircraft (usually those at the age of 15 years and beyond) have to become subject of an enhanced inspection effort. This is where SHM specifically comes into play.

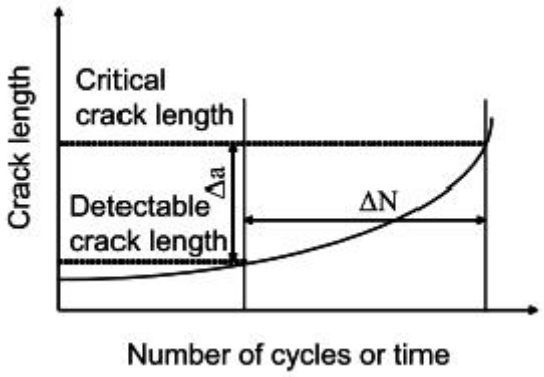



**Figure 1.2 Determination of crack propagation life in damage tolerant design**

## 1.2 Motivation for SHM technology development

There is clearly a rational need for private and government industries to detect damage in their products as well as in their manufacturing infrastructure at the earliest possible time. Such detection requires these industries to perform some form of SHM and is motivated by the potential life-safety and economic impact of this technology. As an example, the US military has long recognized that over their service life the maintenance cost associated with its aircraft far exceeds the purchase price of these systems. Therefore, contracts for new military aircraft often make some mention of SHM with the intent that this technology will help to minimize the lifecycle cost associated with this high-capital expenditure hardware. Aerospace companies along with government agencies are investigating SHM technology for identification of damage to the space shuttle control surfaces hidden by heat shields. Clearly, such damage identification has significant life-safety implications. In addition, many portions of our technical infrastructure including commercial aircraft are approaching or exceeding their initial design life. As a result of economic issues, these structures are being used in spite of aging and the associated damage accumulation. Therefore, the ability to monitor the health of these structures is becoming increasingly important **[1]**.

Most current structural and mechanical system maintenance is done in a time-based mode. As an example, missiles are retired after a set amount of captive-carry hours on the wing of

an aircraft. SHM is the technology that will allow the current time-based maintenance philosophies to evolve into potentially more cost-effective condition-based maintenance philosophies. The concept of condition-based maintenance is that a sensing system on the structure will monitor the system response and notify the operator that damage has been detected. Life-safety and economic benefits associated with such a philosophy will only be realized if the monitoring system provides sufficient warning such that corrective action can be taken before the damage evolves to a failure level. The trade-off associated with implementing such a philosophy is that it requires a more sophisticated monitoring hardware to be deployed on the system and it requires a sophisticated data analysis procedure that can be used to interrogate the measured sensor readings.

Finally, many companies that produce high-capital-expenditure products such as airframes and jet engines are exploring the possibility of moving to a business model where they lease this equipment as opposed to selling the equipment. With these so called "power-by-the-hour" business models the company that manufactures the equipment will take on the responsibilities for maintenance of that equipment. SHM has the potential to extend the maintenance cycles and, hence, keep the equipment out in the field where it can continue to generate revenues for the owner. Also, the equipment owners would like to base their lease fees on the amount of system life used up during the lease time rather than on the current simple time-based lease fee arrangements. Such a business model will not be realized without the ability to monitor the damage initiation and evolution in the leased hardware.

## 1.3 Overview of SHM applied to aerospace structures

Damage identification, as determined by changes in the dynamic response of systems, has been practiced in a qualitative manner, using acoustic techniques (e.g., tap tests on train wheels), since modern man has used tools. More recently, the development of quantifiable SHM approaches has been closely coupled with the evolution, miniaturization and cost reductions of digital computing hardware. In conjunction with these developments SHM has received considerable attention in the technical literature and a brief summary of the developments in this technology over the last 30 years as it relates to the aerospace industry is presented below. To date, the most successful application of SHM technology has been for CM of rotating machinery. The rotating machinery application has taken an almost exclusive non-model based approach to damage identification. The identification process is based on pattern recognition applied to displacement, strain, velocity or acceleration time histories (or spectra) generally measured at a single point on the housing or shafts of the machinery during normal operating conditions and start-up or shut-down transients. Often this pattern recognition is performed only in a qualitative manner based on a visual comparison of the spectra obtained from the system at different times. Databases have been developed that allow specific types of damage to be identified from particular features of the vibration signature. For rotating machinery systems the approximate damage location is generally

known, making a single-channel fast-Fourier transform analyzer sufficient for most periodic monitoring activities. Typical damage that can be identified includes loose or damaged bearings, misaligned shafts, and chipped gear teeth. Today, commercial software integrated with measurement hardware is marketed to help the user systematically apply this technology to the operating equipment. The success of CM is due in part to: (i) Minimal operational and environmental variability associated with this type of monitoring, (ii) Well-defined damage types that occur at known locations, (iii) Large databases that include data from damaged systems, (iv) Well-established correlation between damage and features extracted from the measured data, and (v) Clear and quantifiable economic benefits that this technology can provide. These factors have allowed this application of SHM to have made the transition from a research topic to industry practice several decades ago resulting in comprehensive condition management systems such as the US Navy's Integrated Condition Assessment System. Aircraft engine condition monitoring systems and rotorcraft health and usage monitoring systems have subsequently evolved from these developments in CM and have reached similar levels of maturity in the sense that such systems are now deployed on commercial and military aircraft. The aerospace community began to study the use of damage detection technology during the late 1970's and early 1980's for a variety of civilian and defense applications. Early work focused on loads monitoring where a limited number of sensors are tracked to count load cycles and/or to count the number of time certain threshold response levels are exceeded. Loads monitoring continues to be one of the primary structural health assessment tools used in practice. The development of SHM for aerospace applications has continued and increased considerably in technical sophistication with current applications being investigated for commercial and military aircraft, the National Aeronautics and Space Administration's (NASA) space station and the next generation of reusable launch vehicles. One unique aspect of aerospace SHM applications is that regulatory agencies have been involved with the certification of systems that are deployed on rotorcraft. The application of SHM to aerospace structures has yielded several systems that have made the transition from research to practice. The most notable are rotorcraft health and usage monitoring system (HUMS) and the space shuttle modal inspection system (SMIS) program. Other systems are currently being tested and this industry continues to expend considerable resources on the development of new SHM technology. Perhaps the most refined forms of SHM performed in the aerospace industry are the HUMSs used by the rotorcraft industry. These systems were developed for commercial rotorcraft in response to the significant number of crashes experienced by helicopters servicing North Sea oil platforms. In parallel, these systems were developed for military rotorcrafts. With the introduction of HUMS for main rotor and gearbox components on large rotorcraft, these systems have been shown to reduce "the fatal hull loss within the UK to half what could have otherwise been expected had HUMS not been installed". The essential features of this success are that the rotor speed – although not the torque – is maintained typically within 2% of nominal for all flight regimes and that there is a single load path with no redundancy. These constraints provide a

basis for a stable vibration spectrum from which a change in measured parameter is attributable to component deterioration. As such, the use of vibration data trending for predictive maintenance can be shown to increase rotor component life by 15%. The HUMS systems are used to diagnose faults in helicopter drive trains, engines, oil systems and the rotor system. Most significant is the fact that HUMS have been endorsed by the Federal Aviation Administration (FAA) and Civil Aviation Authority (CAA) as part of an acceptable maintenance strategy with the first certified HUMS systems being flown in the UK in 1991. The Space Shuttle was designed as the first reusable space vehicle and as such requires inspections to assess structural integrity after each flight where it experiences launch, spaceflight, and landing loading environments. In response to this need the SMIS was developed to identify fatigue damage in components such as control surfaces, fuselage panels and lifting surfaces. These areas are covered with a thermal protection system making them inaccessible and, hence, impractical for conventional local non-destructive examination methods. The SMIS has been successful in locating damaged components that are covered by the thermal protection system and all orbiter vehicles have been periodically subjected to SMIS testing since 1987. Early shuttle inspections applied modal test techniques for nondestructive evaluation of the orbiter structure. As an example, testing was performed on the orbiter body flap, which is used to shield the main engines from heat and to provide pitch control during atmospheric re-entry. Single-point random excitation was used to acquire frequency response functions from the flap. Between modal tests, the flap was exposed to an acoustic environment similar to operating conditions. It was observed that the frequencies of the first three modes decreased following the acoustic exposure. Upon disassembly and inspection of the test article, indication of galling in the spherical bearings at the actuator–rib interfaces was discovered. Additionally, shear clips in the interface between the trailing edge wedge and the flap ribs were found to contain significant cracking. It was noted that the conventional visual, X-ray, and ultrasonic inspection techniques had failed to locate this damage. Also, the conventional techniques require the removal of at least some orbiter thermal protective system tiles, whereas the modal inspection technique does not.

The development of SHM systems for reusable launch vehicles continued as NASA began to design a next generation launch vehicle. Strategies were proposed for rapid damage diagnosis and decision making that focused on a distributed sensor system. In addition to the launch vehicle itself, the composite fuel tanks were surfacing as one of the critical items for long-term health monitoring. One such monitoring system for a next-generation launch vehicle was successfully demonstrated during 1996 flight tests at White Sands Missile Range.

Space station applications have primarily driven the development of experimental/analytical methods aimed at identifying damage to truss elements caused by space debris impact. These methods are based on inverse-modeling approaches where analytical models of the undamaged structure are correlated with measured modal properties from both the undamaged and damaged structure. Changes in stiffness indices as assessed from the two

model updates are used to locate and quantify the damage. Since the mid-1990's, studies of damage identification for composite materials have been motivated by the development of a composite fuel tank for a reusable launch vehicle as well as the increasing use of composite materials in all types of commercial and military aircraft. The failure mechanisms, such as delamination caused by debris impacts, debonding of glued joints and corresponding material response for composite fuel tanks are significantly different than those associated with metallic structures. Often such damage is located below the surface of the structure thus increasing the challenges of the damage detection. Also, the composite fuel tank problem presents challenges because the sensing systems must not provide a spark source. This challenge has lead to the development of SHM based on fiber optic sensing systems. Active pulse-echo and pitch-catch wave propagation-based damage detection approaches, acoustic emission passive wave propagation methods and active thermography methods have been developed for damage detection in composite materials.

There are many other examples of more specialized aircraft SHM systems such a monitoring nitrogen pressure in welded chrome-moly fuselage tubing to detect crack and corrosion damage and the development of SHM to monitor rapid satellite assembly and deployment for the US Air Force's Operationally Responsive Space capability where the goal is to assemble and launch a satellite within 1 week of mission definition.

## 1.4 Challenges for SHM applied to aerospace structure

The aerospace application of SHM has challenges that are, in general, similar to the challenges faced by other SHM applications. Aerospace structures experience widely varying environmental and operational conditions that can affect sensor readings. In the case of aerospace structures these changes include widely varying thermal, vibration and acoustic environments, changing gravitational environments, changing mass that results from fuel consumption and changing aerodynamic load caused by varying atmospheric conditions and changing loads that result from how the aircraft is operated. Many portions of the structure are difficult to access when trying to retrofit an existing structure with a sensing system. Some unique challenges faced by the aerospace industry are the restrictions on the weight of the SHM system that can be deployed during flight. Also, the sensing system cannot be a spark source when monitoring fuel tanks, which are a structure of considerable interest for reusable launch vehicle applications. The basic premise of vibration-based SHM feature selection is that damage will significantly alter the stiffness, mass or energy dissipation properties of a system, which, in turn, alter the measured dynamic response of that system. Although the basis for feature selection appears intuitive, its actual application poses many significant technical challenges. The most fundamental challenge is the act that damage is typically a local phenomenon and may not significantly influence the lower-frequency global response of structures that is normally measured during system operation. Stated another way, this fundamental challenge is similar to that in many engineering fields where the

ability to capture the system response on widely varying length and time scales, as is needed to model turbulence or to develop phenomenological models of energy dissipation, has proven difficult. Another fundamental challenge is that in many situations feature selection and damage identification must be performed in an unsupervised learning mode. That is, data from damaged systems are not available. Damage can accumulate over widely-varying time scales, and this poses significant challenges for the SHM sensing system. This challenge is supplemented by many practical issues associated with making accurate and repeatable measurements over long periods of time at a limited number of locations on complex structures often operating in adverse environments. Finally, a significant challenge for SHM is to develop the capability to define the required sensing system properties before field deployment and, if possible, to demonstrate that the sensor system itself will not be damaged when deployed in the field. If the possibility of sensor damage exists, it will be necessary to monitor the sensors themselves. This monitoring can be accomplished either by developing appropriate self-testing and validating sensors (analogous to Built-In Test Equipment or Built-In Self-Test capabilities now routinely deployed in digital avionic systems) or by using the sensors to report on each other's condition. Sensor networks should also be 'fail-safe'. If a sensor fails, the damage identification algorithms must be able to adapt to the new network. This adaptive capability implies that a certain amount of redundancy must be built into the sensor network. In addition to the challenges described above, there are other non-technical issues that must be addressed before SHM technology can make the transition from a research topic to actual practice. These issues include convincing aircraft owners that the SHM technology provides an economic benefit over their current maintenance approaches and convincing regulatory agencies that this technology is reliable and provides a significant life-safety benefit. All these challenges lead to the current state of SHM technology where outside of HUMS applications, SHM in aerospace systems remains a primarily a research topic that is still making the transition to field demonstrations and subsequent field deployment. Therefore, there is a critical need for the development of design principles for aerospace structures utilizing SHM systems (see Design Principles for Aerospace Structures Utilizing SHM) and these design principles will have to be formulated around the fundamental axioms (or some appropriate variant) that have been proposed for SHM. Finally, as SHM technology evolves and matures, it will be integrated into a more comprehensive process referred to as Damage Prognosis. Damage Prognosis (DP) is defined as the estimate of an engineered system's remaining useful life This estimate is based on the output of models that predict material, component and system degradation by coupling information from usage monitoring; SHM; past, current and anticipated future environmental and operational conditions; the original design assumptions regarding loading and operational environments, previous component and system level testing and maintenance, and predictive models for damage initiation and evolution (see Damage Prognosis for Metal and Composite Aerospace Structures). Here the term usage monitoring refers to the process of acquiring operational loading data from a structure or system which preferably includes a measure of

environmental conditions (e.g., temperature, moisture) and operational variables such as mass or speed. Stated another way, DP attempts to forecast system performance by measuring the current state of the system (i.e., SHM), estimating the future loading environments for that system, and predicting through simulation and past experience the remaining useful life of the system. Such predictions will necessarily be probabilistic in nature and will build upon standard maintenance definitions provided under the FAA Maintenance Steering Group-3 process that requires in-service reliability data to justify maintenance as well as safety specific component and system life estimates.

## 1.5 Data acquisition and potential SHM technologies considered

The data acquisition portion of the SHM process involves selecting the excitation methods (when necessary), the sensor type, number and locations, sampling parameter, and the data acquisition/ storage/ transmittal hardware. This process will be application specific and economic considerations will play a major role in making these decisions. The intervals at which data should be collected are another consideration that must be addressed.

Sensing and data acquisition systems for aerospace SHM vary widely depending on the specific application. Many of these systems are commercially available and deployed onboard the structure while it is in flight such as the rotorcraft HUMS. In-flight SHM systems have only become practical with the evolution of microelectronics that allows such systems to be deployed onboard aircraft with minimal weight penalties. Other systems like the shuttle model inspection system (SMIS) used traditional ground-based dynamics data acquisitions systems for tests performed with the shuttle out of service. Generally, large amounts of data can be acquired in-flight necessitating a data management strategy.

Kinematic quantities such as strain and acceleration are the most common response parameters that are monitored using conventional electrical resistance strain gages and piezoelectric accelerometers, respectively. In addition, the aerospace industry has studied the use of fiber optic strain sensors for SHM applications because they are light-weight, do not produce a spark source and because their size makes them somewhat non-intrusive. The small size of optical fibers has motivated several studies where researchers are embedding the fibers into composite materials for more direct measures of their response characteristics. The fibers optic sensors have also been used to measure temperature. Other types of sensors have been deployed for corrosion monitoring as part of a US military demonstration of health monitoring systems for large structural components that investigated different sensing modalities.

Most commonly, when tests are performed in-flight, the aerospace industry measures the response of the structure to operational and environmental loading conditions such as those caused by aerodynamics forces and engine vibration. More recently this industry has started to use active sensing systems, where a small actuator is mounted on the structure to produce

a local excitation signal tailored to enhance the damage detection process. Acoustic emission studies rely on measuring the response of the system that results when strain energy is released during damage initiation or growth. During ground vibration tests performed for damage assessment purposes, electrodynamics shakers are used to excite the structure as is the case with the SMIS.

There is a wide range of sensors and sensor systems being around and developed or under consideration where a selection of those is shown in Figure 1.3 below. The sensors and sensor systems are allocated to a specific physical parameter such as sound, vibration, electromagnetism, temperature, light or possibly others, being further enhanced and more to come such as at the nano scale. Some of the ones being considered most are described below. Data acquisition unit of our case study make use of FBG sensors and in the following an brief introduction of their characteristics is given.

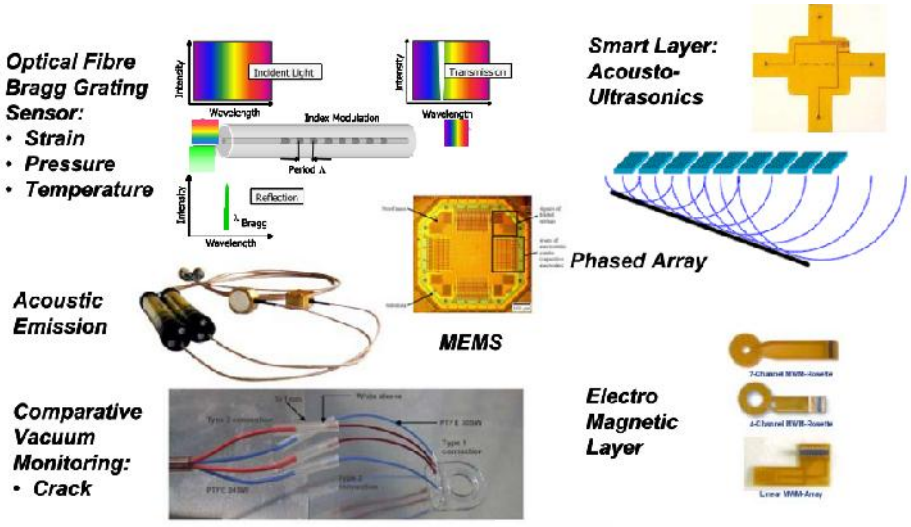



**Figure 1.3 Sensing options for structural health monitoring**

FBG is to be applied to an optical fiber of 100 $\mu m$ in diameter or less that can either be adapted to or integrated into a structure. The latter is specifically popular with composite materials while the former is the method applied to metals. The sensor is able to monitor any kind of strain resulting from either mechanical loads, pressure, temperature, acoustic vibrations or others. A major advantage of the sensor is that a multitude of sensors (possibly hundreds) can be placed along a single fiber, which reduces complexity in wiring significantly. Because of working on an optical basis the system is immune to any electro-magnetic interference. It has shown robustness with regard to in service application

.

## 1.6 Normalization and variability filtering

Because data can be measured under varying conditions, the ability to normalize the data becomes very important to the damage identification process. As it applies to SHM, data normalization is the process of separating changes in sensor reading caused by damage from those caused by varying operational and environmental conditions. One of the most common procedures is to normalize the measured responses by the measured inputs. When environmental or operational variability is an issue, the need can arise to normalize the data in some temporal fashion to facilitate the comparison of data measured at different times of an environmental or operational cycle. Sources of variability in the data acquisition process and with the system being monitored need to be identified and minimized to the extent possible. In general, not all sources of variability can be eliminated. Therefore, it is necessary to make the appropriate measurements such that these sources can be statistically quantified. Variability can arise from changing environmental and test conditions, changes in the data reduction process, and unit-to-unit inconsistencies.

## 1.7 Statistical model development for feature discrimination

The portion of the SHM process that has arguably received the least attention in the technical literature is the development of statistical models for discrimination between features from the undamaged and damaged structures. Statistical model development is concerned with the implementation of the algorithms that operate on the extracted features to quantify the damage state of the structure. The algorithms used in statistical model development usually fall into three categories. When data are available from both the undamaged and damaged structure, the statistical pattern recognition algorithms fall into the general classification referred to as supervised learning. Group classification and regression analysis are categories of supervised learning algorithms. Unsupervised learning refers to algorithms that are applied to data not containing examples from the damaged structure. Outlier or novelty detection is the primary class of algorithms applied in unsupervised learning applications. All of the algorithms analyze statistical distributions of the features derived from the measured data to enhance the damage identification process.

The damage state of a system can be described as a five step process given as the following:

1. Damage detection (existence of a damage)
2. Damage localization
3. Damage type identification (depending on the case study)
4. Damage quantification
5. Damage prognosis

Preceding further in the hierarchy of these steps an increasing knowledge of the damage state is acquired. When applied in an unsupervised learning mode, statistical models are typically

used to answer questions regarding the existence and location of damage. When applied in a supervised learning mode and coupled with analytical models, the statistical procedures can be used to better determine the type of damage, the extent of damage and potential remaining useful life of the structure. The statistical models are also used to minimize false indications of damage. False indications of damage fall into two categories: (i) False-positive damage indication (indication of damage when none is present), and (ii) False-negative damage indication (no indication of damage when damage is present). Errors of the first type are undesirable as they will cause unnecessary downtime and consequent loss of revenue as well as loss of confidence in the monitoring system. More importantly, there are clear safety issues if misclassification of the second type occurs. Many pattern recognition algorithms allow one to weigh one type of error above the other; this weighting may be one of the factors decided at the operational evaluation stage.

# Chapter 2 Theory background

## 2.1 Outlier analysis

To constitute an introductory method for damage detection, here a statistical method, so called ‘outlier analysis’ has been studied. The lowest level of damage identification hierarchy is considered here so that the methods are simply required to signal deviations from normal condition; i.e. the problem is the one of novelty detection. Here, the concept of discordancy from the statistical discipline of outlier analysis is used to indicate deviance from the baseline condition. In the following chapter the methodology has been implemented on our case study.

The problem of damage detection and identification has a natural hierarchical structure. At higher levels, one might require the diagnostic to return information about expected time to failure of a structure, while at the lowest level, the question is simply of whether a fault is present or not. In many ways, the latter is the most fundamental. In response to the need for robust low-level damage detection strategies, the discipline of novelty detection has recently been evolved [7] [8]. The problem is simply to identify, from measured data, if a machine or a structure has deviated from normal condition, i.e., if the data is novel. The idea of novelty detection is not entirely new, in many ways the philosophy is coincident with that of the classical condition monitoring. However, the new terminology is justified by the fact that novelty detection provides a unifying framework for techniques from a wide range of disciplines. Among the large quantity of available approaches to the problem, some are drawn from condition monitoring, others from the field of pattern recognition and yet others from multivariate statistics. The latter field has a very substantial body of theory to support it and is proving to be fruitful source of algorithms for damage detection.

The object of the study is to examine a technique from multivariate statistics and benchmark it on our case study, which has been also examined through other techniques (see chapter 5). The method is that of outlier analysis. This is a well-established field of statistics which has not yet been systematically exploited for damage detection purposes [7]. It will be shown that the method not only allows diagnosis of novelties, but also suggests to which extent the dimension of the data set might be reduced without losing the efficiency of the diagnostic algorithm.

### 2.1.1 Detection of outliers in univariate data

A discordant outlier in a data set is an observation that appears inconsistent with the rest of the data and therefore is believed to be generated by an alternative mechanism to the other data [7] [8]. The discordancy of the candidate outlier is a measure which may be compared against an objective criterion allowing the outlier to be judged to be statistically likely or unlikely to have come from the assumed generating model.

In the case of the univariate data, the detection of outliers is a relatively straightforward process in that the outliers protrude from or other end of the data set. There are numerous discordancy tests but one of the most common, and the one whose extension to multivariate data will be employed later, is based on deviation statistics and is given by,

$$Z_\xi = \frac{|x_\xi - \bar{x}|}{s} \tag{2.1}$$

where $x_\xi$ is the measurement corresponding to the potential outlier and $\bar{x}$ and $s$ the mean and standard deviation of the sample respectively. The two latter values may be calculated with or without potential outlier in the sample depending upon whether inclusive or exclusive measures are preferred. This discordancy value is then compared to some threshold value and the observation declared, or not, to be an outlier.

### 2.1.2 Detection of outliers in multivariate data

A multivariate data set consisting of *n* observations in *p* variables may be represented as *n* points in *p*-dimensional object space. It becomes clear that detection of outliers in multivariate data is more difficult that univariate situation due to the potential outlier having the ability to appear more hidden data mass. That said many of ideas and techniques associated with the detection outlier in multivariate data follow on from those applicable to univariate problems.

The discordancy test which is the multivariate equivalent of equation 2.1 is the Mahalanobis squared distance measure given by,

$$y_i = (\{x_\xi\} - \{\bar{x}\})^T [S]^{-1} (\{x_\xi\} - \{\bar{x}\}) \tag{2.2}$$

where $\{x_\xi\}$ is the potential outlier datum, $\{\bar{x}\}$ is the mean vector of the sample observations and [S] the sample covariance matrix. T indicates transpose.

As with univariate discordancy test, the mean and covariance may be inclusive and exclusive measures. In many practical situations the outlier is not known beforehand and so the test would necessarily be conducted inclusively. The outlier displaying component method which is discussed shortly is also an inclusive method. Finally, the Mahalanobis squared distance of the potential outlier is checked against an appropriate threshold value, as in the univariate case, and the system status is declared as a consequence.

### 2.1.3 Outlier displaying component

In order to display multivariate outliers when dealing with data of greater than two or three dimensions it is necessary to apply special graphical methods. Although there are many graphical techniques for multivariate data, most are not specifically designed to display

outliers. This can be shown [8] to be equivalent to being based on the Mahalanobis squared distance measure with the potential outlier included in the calculation of the sample statistics.

The aim is to find a one-dimensional representation of the sample observation so as to highlight the potential outlier $x_\xi$. The projection vector or one-outlier displaying component, $\beta$, which results in outlier protruding as far as possible from the data mass can be shown [8] to be given by

$$\{\beta\} = [S]^{-1}(\{x_\xi\} - \{\bar{x}\}) \tag{2.3}$$

Where $[S]$ and $\{\bar{x}\}$denote the covariance matrix and the mean of all observation respectively. It is then possible to project the original P-dimensional sample $\{\{x_i\}, i = 1, \dots, n\}$ into one-dimensional sample $\{y_i, i = 1, \dots, n\}$ using the one-outlier displaying component so that each $y_i$ is obtained from equation 2.2.

Note that $y_\xi$ is equal to Mahalanobis squared distance of $x_\xi$ from the mean and the other $y_i$ are the value against which $y_\xi$ can be evaluated.

Gordor stated that one of the useful features of the outlier displaying component is its ability to show which dimensions contribute most to the discordancy of the outlier. It was claimed that the coefficients in the outlier displaying component with the largest absolute values corresponds directly to the variables which have the greatest effect on the discordancy.

### 2.1.4 Calculation of critical values of discordancy

In order to label an observation as an outlier or inlier there needs to be some threshold value against which the discordancy value can be compared. This value is dependent on both the number of observations and the number of dimensions of the problem being studied. The value also depends whether an inclusive or exclusive threshold is required.

A Monte Carlo method was used to arrive at the threshold value and this may be summarized by the following steps.

1) Construct a $(p \times n)$ (number of dimensions×number of observations) matrix with each element being randomly generated from a zero (this “zero” value may change depending on case study. see chapter 4) mean and unit (this “unit” value may change depending on case study. see chapter 4) standard deviation normal distribution.

2) Mahalanobis squared distances calculated for the all observations, using equation 2.1 where $\{\bar{x}\}$ and [S] are either inclusive or exclusive measures (depending on the type of threshold being conducted), and the largest value stores.

3) Process repeated for large number of trails whereupon the array containing all the largest Mahalanobis squared distances then ordered in terms of magnitude. The critical value for 5 and 1% tests of discordancy for a p-dimensional sample of n observations are then given by the Mahalanobis squared distances in the array above which 5 and 1% of the trials occur.

## 2.2 An introduction to artificial neural network

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by examples. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true for ANNs as well.

Neural network simulations appear to be a recent development. However, this field was established before the advent of computers, and has survived at least one major setback and several eras.

Many important advances have been boosted by the use of inexpensive computer emulations. Following an initial period of enthusiasm, the field survived a period of frustration and disrepute. During this period when funding and professional support was minimal, important advances were made by relatively few researchers. Currently, the neural network field enjoys a resurgence of interest and a corresponding increase in funding. The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to obtain good performances.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest are examined and to answer "what if" questions.

Other advantages include:

- Adaptive learning: An ability to learn how to do tasks based on the data given for training or initial experience.
- Self-Organization: An ANN can create its own organization or representation of the information it receives during learning time.

- Real Time Operation: ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
- Fault Tolerance via Redundant Information Coding: Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

### 2.2.1 Human and Artificial Neurons

Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called *dendrites*. The neuron sends out spikes of electrical activity through a long, thin stand known as an *axon*, which splits into thousands of branches. At the end of each branch, a structure called a *synapse* converts the activity from the axon into electrical effects that inhibit or excite activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes.

In a similar way, an artificial neuron (Figure 2.1) is a device with many inputs and one output. The neuron has two modes of operation; the training mode and the using mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the using mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

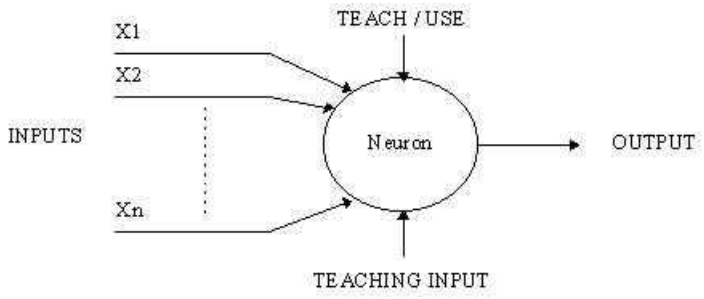



**Figure 2.1 A simple neuron**

### 2.2.2 A more complicated neuron

The previous neuron doesn't do anything that conventional computers don't do already. A more sophisticated neuron (Figure 2.2) is the McCulloch and Pitts model (MCP). The difference from the previous model is that the inputs are ‘weighted’; the effect that each input

has at decision making is dependent on the weight of the particular input. The weight of an input is a number which when multiplied with the input gives the weighted input. These weighted inputs are then added together and if they exceed a pre-set threshold value, the neuron fires. In any other case the neuron does not fire.

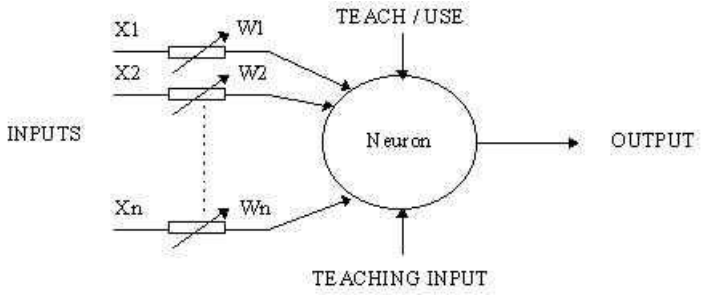



**Figure 2.2 An MCP neuron**

In mathematical terms, the neuron fires if and only if;

$$X1W1 + X2W2 + X3W3 + \ldots > T \tag{2.3}$$

The addition of input weights (W) and of the threshold (T) makes this neuron a very flexible and powerful one. The MCP neuron has the ability to adapt to a particular situation by changing its weights and/or threshold. Various algorithms exist that cause the neuron to 'adapt'; the most used ones are the Delta rule and the back error propagation. The former is used in feed-forward networks and the latter in feed-forward back propagation networks.

Feed-forward back propagation (Feedback) networks (Figure 2.3) can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

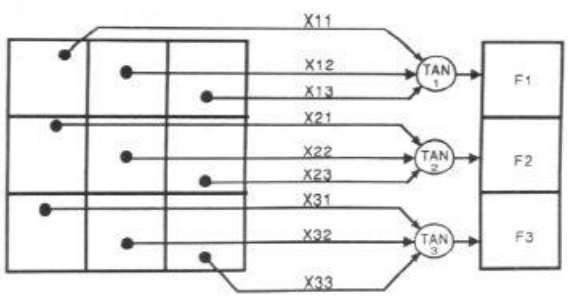



**Figure 2.3 Feedback network**

## 2.2.3 Network layers

The commonest type of artificial neural network consists of three groups, or layers, of units. A layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units (Figure 2.3).

- The activity of the input units represents the raw information that is fed into the network.
- The activity of each hidden unit is determined by the activities of the input units and the weights on the connections between the input and the hidden units.
- The behavior of the output units depends on the activity of the hidden units and the weights between the hidden and output units.

This simple type of network is interesting because the hidden units are free to construct their own representations of the input. The weights between the input and hidden units determine when each hidden unit is active, and so by modifying these weights, a hidden unit can choose what it represents.

Single-layer and multi-layer architectures have to be distinguished. The single-layer organization, in which all units are connected to one another, constitutes the most general case and is of more potential computational power than hierarchically structured multi-layer organizations. In multi-layer networks, units are often numbered by layer, instead of following a global numbering.

## 2.2.4 The Learning Process

The memorization of patterns and the subsequent response of the network can be categorized into two general paradigms:

1. *Associative mapping* in which the network learns to produce a particular pattern on the set of input units whenever another particular pattern is applied on the set of input units. The associative mapping can generally be broken down into two mechanisms:

    - *Auto-association*: an input pattern is associated with itself and the states of input and output units coincide. This is used to provide pattern competition, i.e. to produce a pattern whenever a portion of it or a distorted pattern is presented. In the second case, the network actually stores pairs of patterns building an association between two sets of patterns.
    - *hetero-association*: is related to two recall mechanisms:

        a. *nearest-neighbor* recall, where the output pattern produced corresponds to the input pattern stored, which is closest to the pattern presented, and
        b. *Interpolative* recall, where the output pattern is a similarity dependent interpolation of the patterns stored corresponding to the pattern presented. Yet another paradigm, which is a variant associative mapping, is classification, i.e. when there is a fixed set of categories into which the input patterns are to be classified.

2. *Regularity detection* in which units learn to respond to particular properties of the input patterns. Whereas in associative mapping the network stores the relationships among patterns, in regularity detection the response of each unit has a particular 'meaning'. This type of learning mechanism is essential for feature discovery and knowledge representation.

Every neural network possesses knowledge which is contained in the values of the connections weights. Modifying the knowledge stored in the network as a function of experience implies a learning rule for changing the values of the weights. Information is stored in the weight matrix W of a neural network. Learning is the determination of the weights. Following the way learning is performed, two major categories of neural networks can be distinguished:

- *Fixed networks* in which the weights cannot be changed, i.e. $\mathrm{dW}/\mathrm{dt} = 0$. In such networks, the weights are fixed a priori according to the problem to solve.
- *Adaptive networks* which are able to change their weights, i.e. $\mathrm{dW}/\mathrm{dt} \neq 0$.

All learning methods used for adaptive neural networks can be classified into two major categories:

- *Supervised learning* which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning

process global information may be required. Paradigms of supervised learning include error-correction learning, reinforcement learning and stochastic learning. Important issue concerning supervised learning is the problem of error convergence, i.e. the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. One well-known method, which is common to many learning paradigms, is the **least mean square (LMS) convergence**.

- *Unsupervised learning* uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. Paradigms of unsupervised learning are Hebbian learning and competitive learning.

### 2.2.5 Transfer Function

The behavior of an ANN (Artificial Neural Network) depends on both the weights and the input-output function (transfer function) that is specified for the units. This function typically falls into one of three categories:

- linear (or ramp)
- threshold
- sigmoid

For linear units, the output activity is proportional to the total weighted output. For threshold units, the output are set at one of two levels, depending on whether the total input is greater than or less than some threshold value. For sigmoid units, the output varies continuously but not linearly as the input changes. Sigmoid units bear a greater resemblance to real neurons than do linear or threshold units, but all three must be considered rough approximations.

To make a neural network that performs some specific task, we must choose how the units are connected to one another, and we must set the weights on the connections appropriately. The connections determine whether it is possible for one unit to influence another. The weights specify the strength of the influence.

### 2.2.6 The Back propagation Algorithm

The back propagation algorithm [**bishop**] is used in layered feed-forward ANNs. This means that the artificial neurons are organized in layers, and send their signals “forward”, and then the errors are propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. There may be one or more intermediate hidden layers. The back propagation algorithm uses supervised

learning, where the error (difference between actual and expected results) is calculated. The idea of the back propagation algorithm is to reduce this error, until the ANN learns the training data. The training begins with random weights, and the goal is to adjust them so that the error will be minimal.

The activation function of the artificial neurons in ANNs implementing the back propagation algorithm is a weighted sum (the sum of the inputs $x_i$ multiplied by their respective weights $w_{ji}$):

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^{n} x_i w_{ji} \tag{2.4}$$

We can see that the activation depends only on the inputs and the weights. If the output function would be the identity (output=activation), then the neuron would be called linear. But these have severe limitations. The most common output function is the sigmoid function:

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1+e^{A_j(\bar{x},\bar{w})}} \tag{2.5}$$

The sigmoid function is very close to one for large positive numbers, 0.5 at zero, and very close to zero for large negative numbers. This allows a smooth transition between the low and high output of the neuron (close to zero or close to one). We can see that the output depends only in the activation, which in turn depends on the values of the inputs and their respective weights.

Now, the goal of the training process is to obtain a desired output when certain inputs are given. Since the error is the difference between the actual and the desired output, the error depends on the weights, and we need to adjust the weights in order to minimize the error. We can define the error function for the output of each neuron:

$$E_j(\bar{x}, \bar{w}, d) = \left(O_j(\bar{x}, \bar{w}) - d_j\right)^2 \tag{2.6}$$

We take the square of the difference between the output and the desired target because it will be always positive, and because it will be greater if the difference is big, and lesser if the difference is small. The error of the network will simply be the sum of the errors of all the neurons in the output layer:

$$E_j\left(\bar{x}, \bar{w}, \bar{d}\right) = \sum_j \left(O_j(\bar{x}, \bar{w}) - d_j\right)^2 \tag{2.7}$$

The back propagation algorithm now calculates how the error depends on the output, inputs, and weights. After we find this, we can adjust the weights using the method of *gradient descendent*:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \tag{2.8}$$

This formula can be interpreted in the following way: the adjustment of each weight( $\Delta w_{ji}$) will be the negative of a constant eta ($\eta$) multiplied by the dependence of the previous weight on the error of the network, which is the derivative of $E$ in respect to $w_i$. The size of the adjustment will depend on $\eta$, and on the contribution of the weight to the error of the function. This is, if the weight contributes a lot to the error, the adjustment will be greater than if it contributes in a smaller amount. The Equation 2.8 is used until we find appropriate weights (the error is minimal).

Therefore, we "only" need to find the derivative of $E$ in respect to $w$. This is the goal of the back propagation algorithm, since we need to achieve this backwards. First, we need to calculate how much the error depends on the output, which is the derivative of $E$ in respect to $O_j$ (Eq. 2.6).

$$\frac{\partial E}{\partial O_j} = 2(O_j - d_j) \tag{2.9}$$

And then, how much the output depends on the activation, which in turn depends on the weights (from Equations 2.4 and 2.5):

$$\frac{\partial O_j}{\partial w_{ji}} = \frac{\partial O_j}{\partial A_j}\frac{\partial E}{\partial O_j} = O_j(1 - d_j)x_j \tag{2.10}$$

And we can see that (from Equations 2.9 and 2.10):

$$\frac{\partial E_j}{\partial w_{ji}} = \frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial w_{ji}} = 2(O_j - d_j)O_j(1 - d_j)x_j \tag{2.11}$$

And so, the adjustment to each weight will be:

$$\Delta w_{ji} = -2\eta(O_j - d_j)O_j(1 - d_j)x_j \tag{2.12}$$

We can use Equation 2.12 as it is for training an ANN with two layers. Now, for training the network with one more layer we need to make some considerations. If we want to adjust the weights (let's call them $v_{ik}$) of a previous layer, we need first to calculate how the error depends not on the weight, but in the input from the previous layer. This is easy, we would just need to change $x_i$ with $w_{ji}$ in Equations 2.10, 2.11, and 2.12. But we also need to see how the error of the network depends on the adjustment of $v_{ik}$. Therefore:

$$\Delta v_{ik} = -\eta \frac{\partial E}{\partial v_{ik}} = -\eta \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial v_{ik}} \tag{2.13}$$

Where:

$$\frac{\partial E_j}{\partial w_{ji}} = 2(O_j - d_j)O_j(1 - O_j)x_j w_{ji} \tag{2.14}$$

And assuming that there are inputs $u_k$ into the neuron with $v_{ik}$ (from Eq. 2.10):

$$\frac{\partial x_i}{\partial v_{ik}} = x_i(1 - x_i)v_{ik} \tag{2.15}$$

If we want to add yet another layer, we can do the same, calculating how the error depends on the inputs and weights of the first layer. We should just be careful with the indexes, since each layer can have a different number of neurons, and we should not confuse them.

For practical reasons, ANNs implementing the back propagation algorithm do not have too many layers, since the time for training the networks grows exponentially. Also, there are refinements to the back propagation algorithm which allow a faster learning.

### 2.2.7 Generalization and early stopping

One of the major advantages of artificial neural networks is their ability to generalize. This means that a trained ANN could classify data from the same class as the learning data that it has never seen before. In real world applications developers normally have only a small part of all possible patterns for the generation of an artificial neural network. The default method for improving generalization is called early stopping. This technique is automatically provided for all of the supervised network creation functions, including the back propagation network creation functions in software such as Matlab.

To reach the best generalization, the dataset should be split into three parts:

- Training data set: The training set is used to train an ANN. The error of this dataset is minimized during training.
- Validation data set: The validation set is used to determine the performance of an ANN on patterns that are not trained during learning.
- Test data set: A test set is used for final checking of the overall performance of an ANN.

The learning should be stopped in the minimum of the validation set error. At this point the ANN generalizes best. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. When learning is not stopped, overtraining

occurs and the performance of the ANN on the whole data decreases, despite the fact that the error on the training data still gets smaller.

The test set error is not used during training, but it is used to compare different models. It is also useful to plot the test set error during the training process. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this might indicate a poor division of the data set.

### 2.2.8 Committee of networks

It is a common practice in the application of the neural networks to train many different candidate networks and then to select the best, on the basis of performance on an independent validation set for instance, and keep only this network and discard the rest [13]. There are two disadvantages with such an approach. First, all the effort in training the remaining networks is wasted. Second the generalization performance on the validation set has a random component due to the noise on the data, and so the network which had best performance on the validation set might not be the one with the best performance on the new test data.

These drawbacks can be overcome by combining the networks together to form a committee. The approach of such an approach is that it can lead to significant improvements in prediction of new data, while evolving little additional computational effort. In fact the performance of a committee can be better than the performance of the best single network used in isolation. For notational convenience we consider networks with a single output, although the generalization to several outputs is straightforward. Suppose we have a set of $L$ trained network models $y_i(x)$ where $i = 1, \dots, L$. This set might contain networks having different number of units, or networks with the same architecture but trained to different local minima of the error function. It might even include different kinds of network models or a mixture of network and conventional model. We denote the true regression function which we are seeking to approximate by h(x). Then we can write the mapping function of each network as a desired function plus an error.

$$y_i(x) = h(x) + \epsilon_i(x) \tag{2.16}$$

The average sum-of-squares for model $y_i(x)$ can be written as

$$E_i = \mathcal{E}[\{y_i(x) - h(x)\}^2] = \mathcal{E}[{\epsilon_i}^2] \tag{2.17}$$

Where $\mathcal{E}[.]$ denotes the expectation, and corresponds to an integration over $x$ weighted (averaged) by unconditional density of $x$, which $x$ in our literature is the crack length variable, so that

$$\mathcal{E}[\epsilon_i^2] = \int \epsilon_i^2(x)p(x)\,dx \tag{2.18}$$

From (2.17) the average error made by networks acting individually is given

$$E_{ave} = \frac{1}{L}\sum_{i=1}^{L} E_i = \frac{1}{L}\sum_{i=1}^{L} \mathcal{E}[\epsilon_i^2] \tag{2.19}$$

We now introduce a simple form of committee. This involves taking the output of the committee to be the average of the outputs of $L$ network which comprise the committee. Thus, we write the committee prediction in the form

$$y_{COM}(x) = \frac{1}{L}\sum_{i=1}^{L} y_i(x) \tag{2.20}$$

The error due to the committee can then be written as

$$E_{COMT} = \mathcal{E}\left[\left(\frac{1}{L}\sum_{i=1}^{L} y_i(x) - \mathrm{h(x)}\right)^2\right] = \mathcal{E}\left[\left(\frac{1}{L}\sum_{i=1}^{L} \epsilon_i\right)^2\right] \tag{2.21}$$

If we now make the assumption that the errors $\epsilon_i(x)$ have zero mean and are uncorrelated, so that

$$\mathcal{E}[\epsilon_i] = 0, \;\; \mathcal{E}\,\epsilon_i\epsilon_j] = 0 \;\; if\; j \neq i \tag{2.22}$$

Then, using Equation 2.19 we can relate the committee error (Eq. 2.21) to the average error of the networks acting separately as follows:

$$E_{COMT} = \frac{1}{L^2}\sum_{i=1}^{L} \mathcal{E}[\epsilon_i^2] = \frac{1}{L}E_{AV} \tag{2.23}$$

This represents the apparently rather dramatic result that the sum-of-squares error can be reduced by a factor of $L$, simply by averaging the predictions of $L$ networks. In practice, the reduction of error is generally much smaller than this, because the errors $\epsilon_i(x)$ of different models are typically highly correlated, and so the Assumption 2.22 does not hold. However we can easily show that the committee averaging process cannot produce an increase in expected error by making use of Cauchy's inequality in the form

$$\left(\sum_{i=1}^{L} \epsilon_i\right)^2 \leq L\sum_{i=1}^{L} \epsilon_i^2 \tag{2.24}$$

which gives the result

$$E_{COMT} \leq E_{AV} \tag{2.25}$$

Typically, some useful reduction in error is generally obtained, and the method has the advantage of being trivial to implement. There is a significant reduction in processing speed for new data, but in many in application this will be irrelevant. The reduction in error can be viewed as arising from reduced variance due to the averaging over many solutions. This suggests that the members of the committee should not individually be chosen to have optimal trade-off between bias and variance, but should have relatively smaller bias, since the extra variance can be removed by averaging.

The simple committee discussed so far involves averaging the predictions of the individual networks. However, we might expect that some members of the committee will typically make better prediction than other members. We would expect to be able to reduce the error still further if we give greater weight to some committee members than to others. Thus, we consider a generalized committee prediction given by a weighted combination of the predictions of the members of the form

$$y_{GEN}(x) = \sum_{i=1}^{L} \alpha_i y_i(x) = \mathrm{h}(x) + \sum_{i=1}^{L} \alpha_i \epsilon_i(x) \tag{2.26}$$

where the parameter $\alpha_i$ will be determined shortly. We now introduce the error correlation matrix **C** with elements given by

$$C_{ij} = \mathcal{E}\left[\epsilon_i(x)\epsilon_j(x)\right] \tag{2.27}$$

This allows the error due to the generalized committee to be written as

$$E_{GEN} = \mathcal{E}[\{y_{GEN}(x) - \mathrm{h}(x)\}^2] \tag{2.28}$$

$$= \mathcal{E}\left[\left(\sum_{i=1}^{L} \alpha_i \epsilon_i(x)\right)\left(\sum_{j=1}^{L} \alpha_i \epsilon_j(x)\right)\right]$$

$$= \sum_{i=1}^{L}\sum_{j=1}^{L} \alpha_i \alpha_j C_{ij}$$

We can now determine optimal values for $\alpha_i$ by the minimization of $E_{GEN}$. The way to calculate optimal values for $\alpha_i$ is not discussed here as it involves mostly mathematical concepts that are not of our interest here. The interested readers can refer to [13] [14].

# Chapter 3 Experimental setup and damage index identification

This chapter starts with a brief explanation about the experimental setup procedure of our case study. Further into the chapter the data mining and feature extraction will be performed on the raw data acquired from the experimental tests (see Section 3.2); this part introduces the same method of normalization that has to be performed on the FEM database used in chapter 5. The eventual purpose of the chapter is to end up with experimental datasets which will be of an essential use in the chapter 4 where outlier analysis will be conducted on these data, and in chapter 5 where the Finite Element model, used in diagnostic algorithm, must be compared and then verified with these data sets; Furthermore, all algorithms in the hierarchy of diagnostics must be tested with the provided experimental datasets. Part of the information provided by this chapter has not been carried out as a part of this thesis, and has been extracted from the preceding works done on this case study. Interested readers can refer to [4] for detailed information concerning this chapter. Finally the effect of random load components can be investigated on the defined damage indices. In section 3.4 a different method of data mining from the raw database, against the firstly introduced method in section 3.3, has been taken in order to investigate the above mentioned issue regarding the effect of random loads components on the defined damage index.

## Sensor network

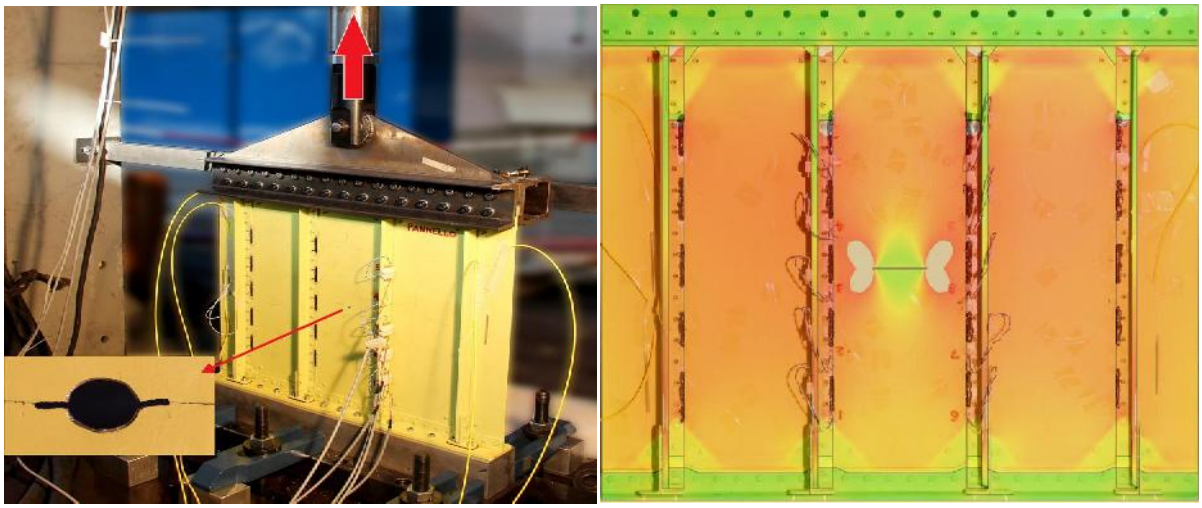


**Figure 3.1 Experimental test setup with lower and upper gripper illustrated (Left). Sensor network alignment based on FBG technology, installed on the test panel (Right).**

The sensor network has been configured as depicted in Figure 3.1. The regions near the upper and lower gripping systems were not considered. The resulting network consists of 20 FBG sensors positioned on stringers (5 sensors per stringer). Stringers are numbered in

sequence from 1 to 4 (left to right), and 5 FBG sensors on each stringer are numbered from 1 to 5 (up to down).

## 3.1 Damage index definition

The helicopter fuselage is subjected to a time varying load that is hard to predict due to the environmental influences. The effect of the boundary load has thus to be filtered out, while taking into consideration that the strain variation due to the load, has orders of magnitude higher than the damage effect over the strain field where e.g. the strain shows a value about 700 $\mu\varepsilon$ in an undamaged case and this value rises to a maximum about 1400 $\mu\varepsilon$ at the largest crack length considered for a sensitive sensor (Figure 3.2 none normalized DI). One could decide to leave the load condition as an output parameter of the diagnostic algorithm; nevertheless this will further complicate the inference problem as a parameterization of the load condition would be required. Thought this might be a further improvement of the work, for the purposes of the present thesis it has been decided to filter the load influence on the mentioned structure. The approach described here consists in normalizing each sensor of a confined region (the entire panel might be seen as part of global fuselage) with respect to the average value measured by all sensors within the same region [4]. As shown in Figure 3.2 the damage influences a sufficient percentage of the sensors and the average strain value measured by all the sensors is linearly dependent on loads and is largely unaffected by the damage. Therefore, one can consider each sensor strain output as a multiplication of load by a constant coefficient for a certain crack length i.e. $\varepsilon_i^{cl} = K_i^{cl} \times F$, where $F$ is the applied sinusoidal load by actuator, $K_i^{cl}$ is the constant coefficient relevant to a certain crack length of the sensor $i$ ($K_i^{cl}$ obviously increases for the sensors close to the crack, as the crack grows longer) and $\varepsilon_i^{cl}$ is the corresponding collected strain value for that sensor.

Thus, by using the average strain as normalization factor for each single sensor, the effect of the load can be filtered out (Eq. 3.1). The method remains valid only for linear material behavior and under the assumption that the damage is localized and affects the minority of the sensors; otherwise the damage information would neutralize the normalization factor. In addition, it is important that the sensors contributing to the average value, all are measuring homogeneous quantities (all measuring strain in the same direction as the normalized sensor and simultaneously). Finally assuming that the N strain sensors measure exactly in the same direction, the expression of the normalized output $\varepsilon_k^{norm}$ for the $k^{th}$ sensor is given by,

$$\varepsilon_k^{norm} = \frac{\varepsilon_k}{\sum_{i=1}^{N} \varepsilon_i / N} \tag{3.1}$$

$$\varepsilon_k^{norm} = \frac{K_k^{cl} \times F}{(\sum_{i=1}^{N} K_i^{cl} \times F)/N} = \frac{K_k^{cl}}{\sum_{i=1}^{N} K_i^{cl}/N} = cts \tag{3.2}$$

The real strain signal, without normalization, induced by the sinusoidal fatigue load with constant amplitude adopted during the experimental fatigue crack propagation test for panel 1

has been depicted in Figure 3.1 in the left plot. The strains are sampled when the peak of the sinusoidal load applies (35kN). For each crack length, a number of sampling of peak strain (peak mode sampling) are performed and then averaged, resulting to a more robust evaluation of the peak strain value. In Figure 3.1 the Str3-FBG3 sensor is close to the crack position as it shows a ascending trend (sensitive sensor), meanwhile the strain value of the Str1-FBGg3 sensor remains rather constant with the crack propagation, as it is far from the crack position (insensitive sensor). The normalization effect on load filtering can be confirmed by comparing both plots of Figure 3.2; e.g. the absolute strain values of all 20 sensors show a discrepancy at crack length of 63mm with respect the normalized strain values, where a consistency with strain values of neighborhood crack lengths is present. This effect is crystal-clear referring to the Figure 3.3 in which strain sampling has been performed randomly in time as the sinusoidal load varies.

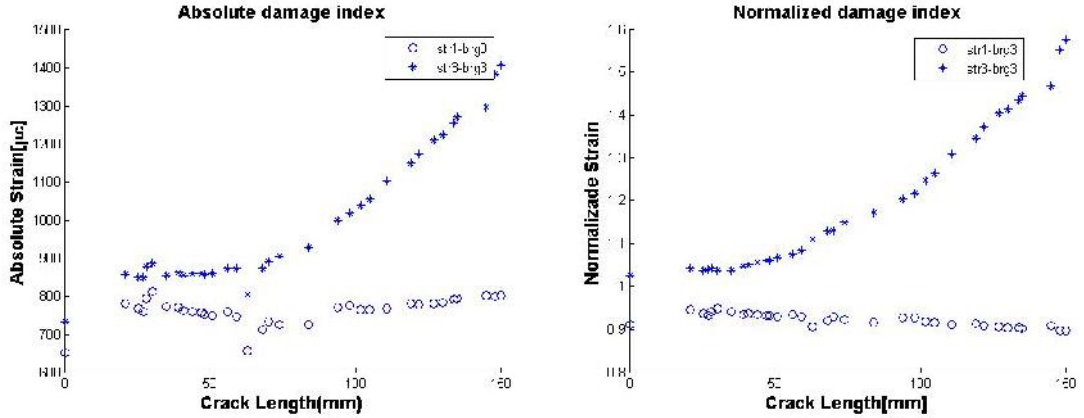



**Figure 3.2 Effect of normalization in load filtering when the strains are sampled at load peak value. Absolute strain values (Left). Normalized strain values (Right)**

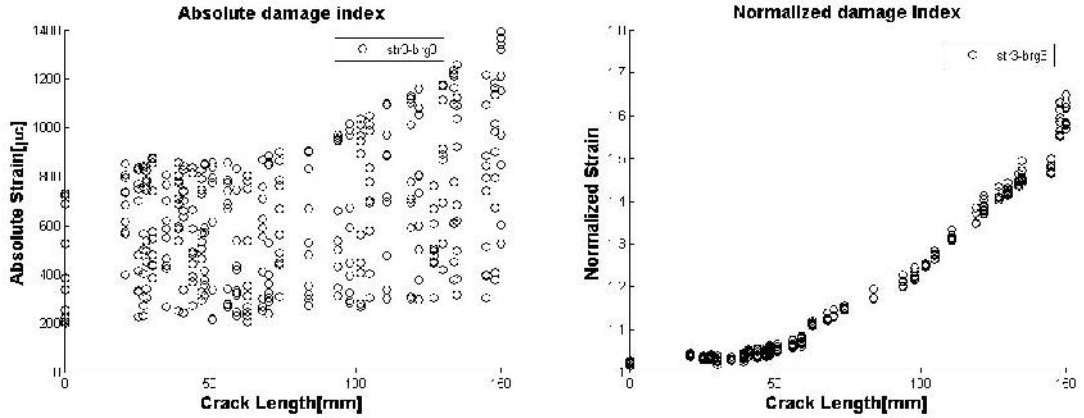



**Figure 3.3 Effect of normalization in load filtering when the strains are sampled at randomly in time. Absolute strain values (Left). Normalized strain values (Right)**

## 3.2 Real data acquisition simulation

In real operation condition, there is no information *a priori* about the quality and quantity of the load exerted to the system, therefore a method has to be introduced in order to generalize the lab-case results to the real case. To simulate a real case with an unknown random load, the time instant for the simultaneous acquisition of the sensor is randomly sampled 50 times for each crack length level (peaks are not considered in this case), thus simulating a load which falls in the [3.5kN-35kN] range. For a non normalized strain, this sampling method corresponds to a wide variation in the sampled strain value (Fig. 3.3 left). Nevertheless, upon the implementation of normalization, the load influence is filtered out from the strain values (Fig. 3.3 right). To compare this sampling method with the one of peak selection, the mode strain value has been calculated for these 50 strain values at each crack length. The mode value can give a good estimate of the strains' true value. In the next step, the mode value is compared with the strain value obtained from peak sampling method (Fig. 3.4). The good compatibility between two lines in Figure 3.4 verifies the efficiency of methodology in random sampling (random mode sampling) which was adopted to generalize data acquisition to a real case.

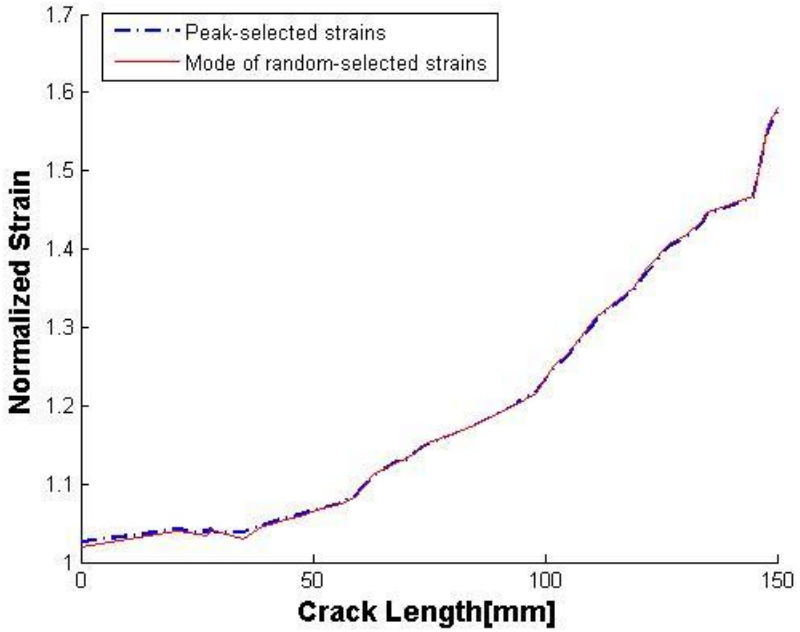



**Figure 3.4 Comparison of modes of normalized randomly sampled strains with normalized peak sampled strains.**

## 3.3 Effect of locally exciting random loads

In some cases, the monitored component will undergo some random local excitation e.g. due to regional temperature variation within the panel, local stress distribution resulting from localized external forces and the regional changes in boundary condition, etc. The variation

of strain value, measured by all the sensors in average, is no longer linearly dependent on the main load (e.g. the sinusoidal load which is applied in the boundary conditions and is sensed by all sensors). This is due to the fact that not all sensors feel this local load function. The sensors sensing this local load function (other than a load with the same frequency-pattern of the main load or a load of constant or zero amplitude), are now producing strain values which include an extra component which does not vary linearly together with other sensors. For this reason the local load component cannot be filtered out of the sensors affected by it, and this appears as a wider dispersion in the randomly sampled strain values for a certain crack length. This influences the strain value dispersion similar to that of an uncorrelated noise present in data acquisition from each sensor. Basically, in diagnostic framework, this might induce misclassifications, intended as false alarms and missed damage detections.

Knowing this fact it is possible to investigate the effect of the locally exciting random loads of different magnitudes by addition of Gaussian noise of different levels using equation 5.2. The added noise is a function of the amplitude of the sampled strain. STDs of order of 2.5% and 5% of the strain magnitude have been selected for the added noise. This can highlight the effect of localized loads of different levels on the sampled strains (see Figure 3.5). Considering the 95% of confidence interval (CI), the mentioned magnitudes for the additional noise can be a representative of unknown localized loads of 5% and 10% of the main load (the load which is sensed by all sensors).

At each crack length, the divergence from the ideal damage indices can be appreciated comparing the mode value of the normalized randomly sampled strains (biased with Gaussian noise) with the corresponding normalized peak strain indicates. We may call the peak sampled strains as *ideal damage indices* as they are supposedly obtained in an ideal testing condition from a test panel (this ideal damage index is of extensive use when verifying the Finite Element model in chapter 5). The divergence of each mode line in Figure 3.6 from the ideal damage index line can be perceived as the uncertainty that can be present in acquired strain values from fuselage when the helicopter operates in real conditions of different severity. As it is expected for random sampling with no noise the mode line lies on the ideal damage index line (Figures 3.4 and 3.6)

This uncertainty has not been counted as a parameter when defining the anomaly detection algorithm in the outlier analysis context (chapter 4), or in Finite Element model verification carried out in section 5.3. Nevertheless this uncertainty (introduction of additional noise) becomes part of the considerations when testing the novelty detection system with real crack propagation dataset of test panel 1 (see Section 4.9), and also when testing the diagnostic algorithms, deigned based on MLP neural networks, with simulated crack propagations (see Section 5.10).

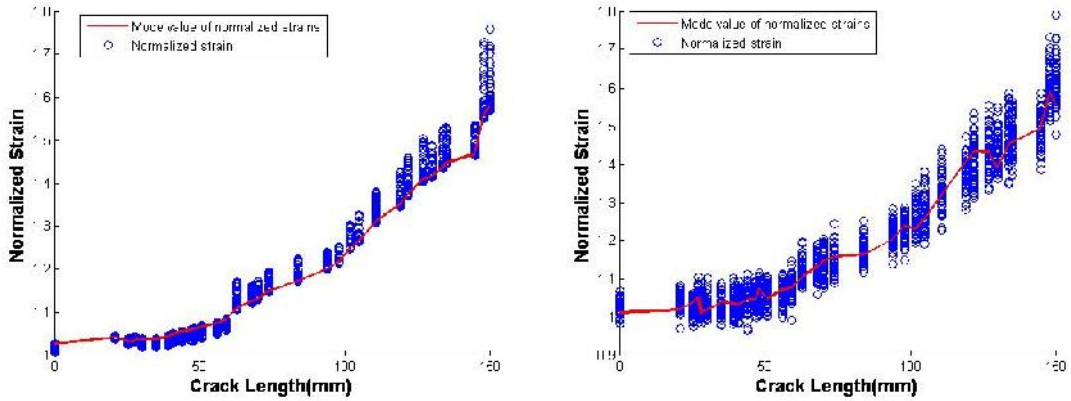



**Figure 3.5 Randomly sampled strain values and relevant mode value. Strains biased with 5% additional noise (Left). Strains biased with 10% additional noise (Right)**

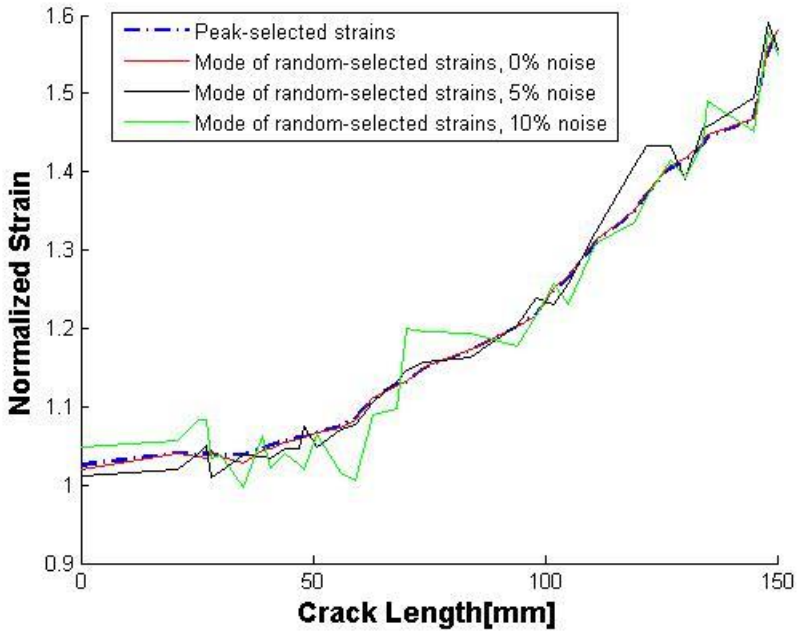



**Figure 3.6 Mode comparison with ideal damage index line. The mode values have been calculated for normalized randomly sampled strains where they are biased with noise of different levels of 0%, 5% and 10%.**

# Chapter 4 Outlier analysis for novelty detection in multivariate empirical data sets

Outlier analysis is a statistical novelty detection method. The application of this method is reliant upon possessing a sufficient sampled observation of the system in the normal state (healthy condition). In the current case study this corresponds to sample enough strain patterns from a healthy panel which is being tested under simulated operating load. The database collected upon this condition is referred as *baseline* data set (undamaged database). The availability of the baseline data set is of the essential need to define the diagnostic algorithm based on outlier analysis. In the other words, as the outlier analysis is considered as an *unsupervised* algorithm, it is necessary and enough to only have the baseline database. When designing outlier-analysis-based diagnostic unit, the most attention is given to the definition of a threshold value against which the discordancy value of sensor readings can be compared. This is necessary to label a sampled strain reading (observation) as an outlier or inlier. This value is dependent on both the number of observations and the number of dimensions of the problem being studied (Section 2.1). Confirming that the baseline strain values, for each sensor, is distributed normally (see Section 2.1), a threshold value can be obtained with the use of Monte Carlo method.

Detection of an outlier indicates that an abnormality is present in the behavior of the system. This informs about a possible alternative mechanism inside the panel generating the abnormality [7] [8]. After definition of a proper threshold value, any sampled observation (strain readings) of the loaded panel can be checked against this threshold, thus, it allows determining whether a damage is present in the panel or not. Working with a multivariate data set in our case study, the Mahalanobis squared distance (Eq. 2.2) has been applied to each sensor reading (potential outlier), which generates a scalar value that is checked against an appropriate threshold value; this check provides a simple binary yes/no answer. The system status is declared as a consequence.
In this chapter at first, data preparation framework is shortly discussed, where the necessary data bases for threshold determination and algorithm testing are obtained. Then, the threshold determination methods are discussed in details. MATLAB 2010a has been implemented for the analysis of this chapter.

## 4.1 Procedures to data preparation

Data sets which are utilized in the context of outlier analysis are acquired from four experimental fatigue crack propagation tests, namely test panel l, 2, 3 and 4. Within the empirical tests all panels are subjected to a harmonic load which varies in range of 3.5KN to 35KN. Strain measures, for both damaged and undamaged scenarios, are sampled from 20 installed FBG sensors with a frequency of 1000Hz. Note that this is the *raw* database which is not used directly in the procedure of the design and testing of the algorithm. We shall take

the same exact approach in data preparation and feature extraction which was taken in chapter 3; therefore, again using both peak strain sampling and random strain sampling modes.

After designing the diagnostic algorithm, the testing is performed where the both baseline data set and damaged data set are used in testing. It is expected that these two testing data sets will be labeled as normal data and damaged data respectively.

## 4.2 Threshold determination using Monte Carlo approach

The Monte Carlo approach (Section 2.4) in threshold evaluation makes uses of the discordancy values calculated over sensor readings (sampled observations) of the baseline condition. The scalar discordancy values can be evaluated by the use of the Mahalanobis square distance (Eq. 2.2). Therefore, before going through the Monte Carlo approach, we shall obtain the parameters "standard deviation ($\bar{\sigma}$)" and "mean ($\bar{x}$)" which are present in the Equation 2.2. As the working data sets are multivariate, these tow parameters are given in the forms of vectors.

As we will discuss later in Section 5.3, a calibration is performed on the experimental data in which the mean value of the baseline strain measures (of each sensor-dimension) is set to "one". Therefore, all the elements of the mean vector are equal to one. The $\bar{\sigma}$ is calculated over the distribution of the baseline strain measures (of each sensor-dimension), which are sampled in the random mode (see Section 3.3). For example, 100 strain measures (observations) are sampled randomly in time when the panel is loaded in its baseline condition, thus providing a $20 \times 100$ matrix (20 FBGs or dimensions) of baseline data set. Then, the STD of each dimension is calculated to give the elements of vector $\bar{\sigma}$.

Upon the availability of these two parameters, the indicated procedure in Section 2.14 is taken, in order to define the threshold. Note that some modification has to be done to make the methodology applicable to our case study. Here, for the sake of convenience we shall restate the steps (the necessary modifications have been executed):

1. Construct a $(p \times n)$ (number of dimensions×number of sampled observations) matrix with each element being randomly generated from a *one* mean and calculated standard deviation over the baseline strain measures, with a normal distribution.
2. Mahalanobis squared distances calculated for the all observations, using equation 2.1 where $\{\bar{x}\}$ and [S] are either inclusive or exclusive measures (depending on the type of threshold being conducted), and the largest value stores.
   The vector $\bar{x}$ is a unit vector, and [S] is calculated over the baseline data set.
3. Process repeated for large number of trails (e.g. 200 times) whereupon the array containing all the largest Mahalanobis squared distances then ordered in terms of magnitude. The critical value for 5 and 1% tests of discordancy for a p-dimensional

sample of n observations are then given by the Mahalanobis squared distances in the array above which 5 and 1% of the trials occur.

4. Having the vector of largest Mahalanobis squared distances, in the last step, the mean value $\mu$ and the STD $\sigma'$ of this vector is calculated. Then, the threshold of 95 and 99% confidence intervals (CI) (95 and 99% of the discordancy values fall in that interval) are calculated as following:

$$CI_{95\%} = \mu + 1.96 \times \sigma'$$

and

$$CI_{95\%} = \mu + 2.58 \times \sigma'$$

respectively. This is adopted as 5 and 1% tests of discordancy for a p-dimensional sample of n observations above which 5 and 1% of the trials occur [11].

### 4.2.1 Specifications on threshold determination

As it will be mentioned in Section 5.3, there are a variety of factors that contribute to the presence of an uncertainty in sensor readings for both baseline and damaged strain patterns. This uncertainty actually explains the reason why the strain reading of each sensor shows a distribution, as they are influenced with this uncertainty. Furthermore, it has been mentioned in chapter 3 that the strain readings for small cracks do not show sensitivity to the damage presence. Therefore, these strain readings adopt values similar to the baseline strain measures. Knowing these facts, we can include the strain measures of small cracks inside the baseline database, which is used for the determination of the threshold (Fig. 4.1).

This approach suggests a number of benefits. First, we are provided with a larger baseline database, which results to a more robust evaluation of the parameters of Equation 2.2. Second, we end up with a baseline database that its distribution nature is more normal (normal distribution of the data is a priory condition to use Monte Carlo Method). Third, we insert the uncertainties in the final value of threshold; this provides a more robust threshold as it decreases the probability of receiving a fake alarm due to the variation of the strain readings which is caused by uncertainties.

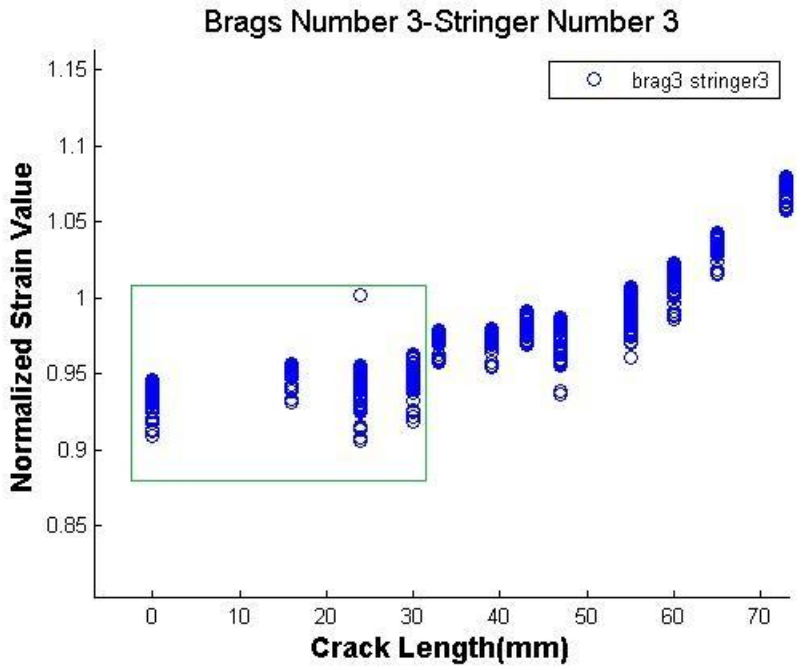



**Figure 4.1 First four crack lengths of str3-brgg3 are taken to produce 13th dimension of "total testpattern 1"**

## 4.3 Evaluation of the algorithm with experimental crack propagation

After calculating the threshold value, in the next step, it is possible to check the performance of the algorithm with a real crack propagation which is located in the centre of the panel. The data which are provided for testing phase are sampled in the random mode (see Section 3.3). In this method 50 normalized strain measures (observations), for each crack length, have been sampled. In the next step the, each single strain measure is fed to the equation 2.2, in order to assess the corresponding novelty value. Then, this novelty value which is an index of the discordancy of the observation is compared against the threshold. Any novelty value which falls above the threshold can be assumed as the presence of damage inside the panel.

In the next step, we shall to define a proper representation for the algorithm, wherein the performance of the algorithm can be evaluated by an experimental strain pattern (see Fig. 4.5). In the Figure 4.5, Mahalanobis squared distance (novelty value) of all observations is given as an implicit function of crack length. In the other words, all 50 novelty values (corresponding to 50 observation) for each crack length is included in between of two vertical lines. Note that Mahalanobis squared distances are given in log scale.

The reason lying behind this behavior is the fact that starting from a small crack length, passing from one small crack length to the next one (e.g. crack length 16mm to 24mm) every dimension experiences a variation in both mean value and standard deviation (data points dispersion). In fact this variation has nothing to do with the sensitivity to crack presence but

intrinsic noise of the system (highly possibly coming non-linear load effects and temperature). Thereby there is an accumulation of the contribution of each dimension in equation 2.2 which results in a larger novelty value as output using equation 2.2 comparing to only when the sensitive sensors are considered. However, at larger cracks (e.g. above 45mm), where only monotonically increasing trend can be observed, the less sensible sensors introduce a negligible contribution and the sensors 8 and 13 which are the closest to the damage location project a dominant quantity in the output of equation .

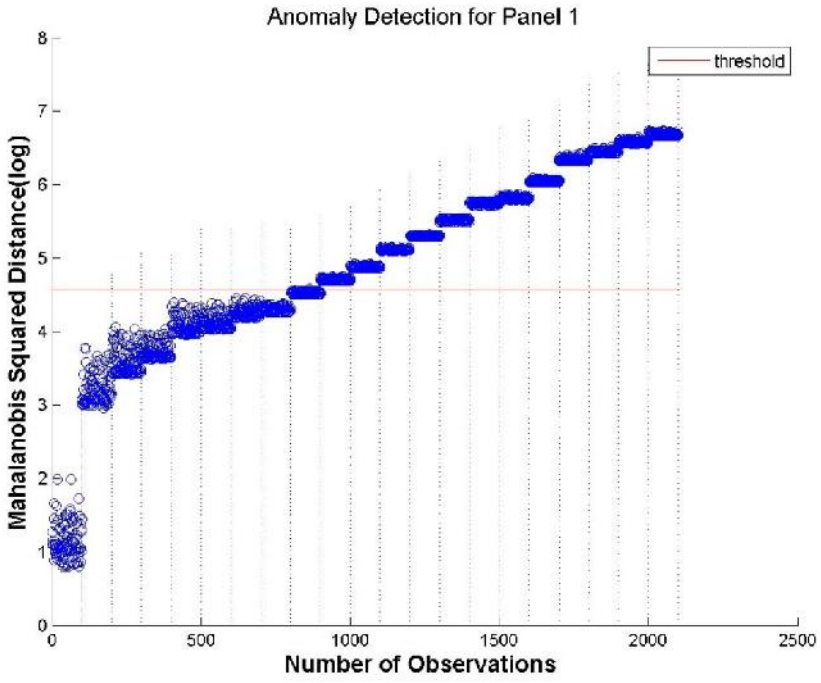



**Figure 4.2 Novelty values calculated for different crack lengths by using Mahalanobis squared distance (log scale). The horizontal red line is the threshold.**

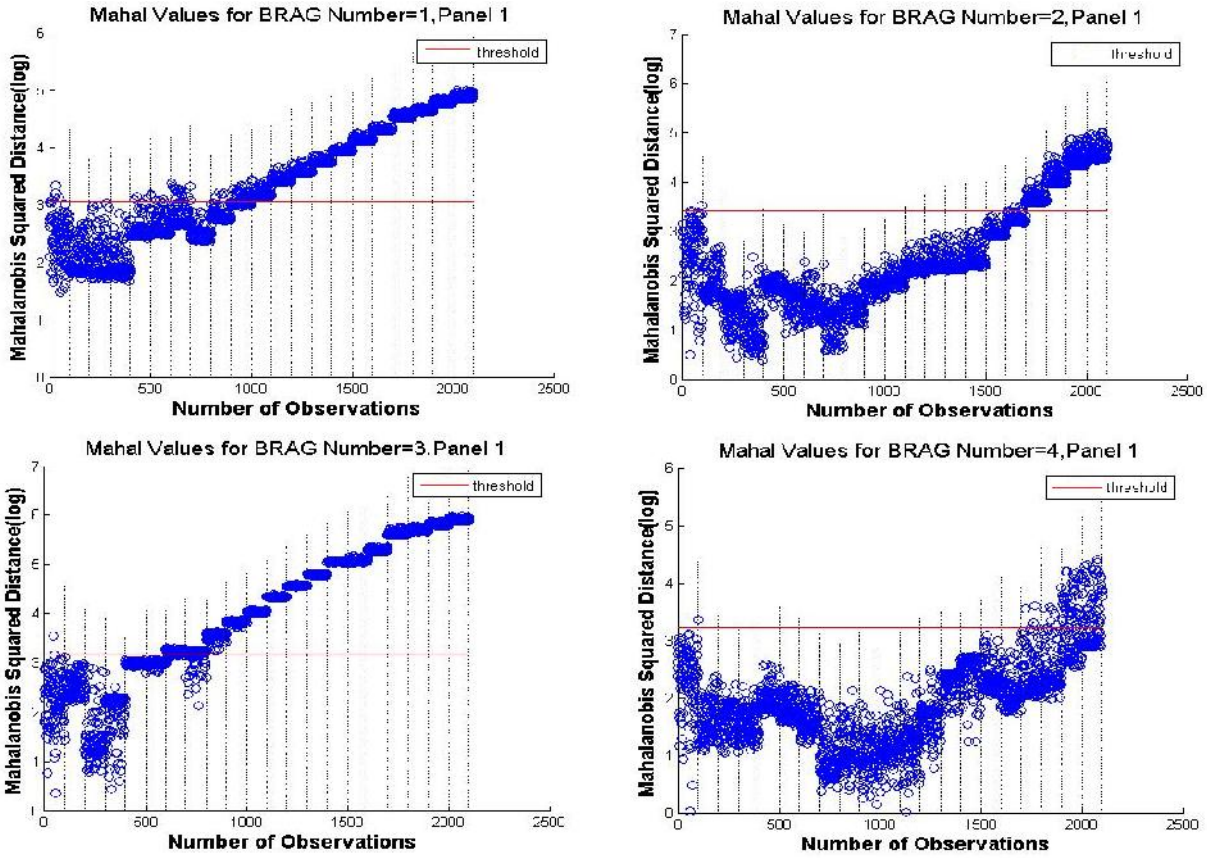
## 4.4 Outlier analysis and damage localization

To go further it is possible to implement outlier analysis for damage location detection. This analysis is not suggested for a real application, since the maximum precision in estimated location using this methodology is incomparable with offered precision of methodology which is conducted in chapter 5 (the application of the artificial neural networks). The method applied here, proposes a very straightforward procedure to localize the crack.

To do so, a dimension reduction of the both baseline and damaged databases has been considered. To be specific, the databases are divided into subsets, where each subset contains only some of the sensors reading. These subsets once include the strain readings from FBGs located on each stringer, and once the strain readings from FBGs located on each Bragg.

In the next step the novelty values are calculated and represented in a similar manner of that done for entire panel. When the novelty values of each subset are compared against the corresponding threshold, it can be seen that at a certain crack length the threshold has been surpassed (see Figure 4.6). However, this crack length is not equal for all subsets, and this can be used to understand which is the closest stringer or Bragg to the damage site; e.g. by looking to the Figure 4.6, the novelty values relevant to the third Bragg subset surpasses the threshold at an earlier crack length. This can indicate that damage site is close to the central Bragg.

Adopting the same path for the data subsets relevant to the FBGs located on stringers, we obtain four corresponding plots of calculated novelty values (see Fig. 4.7). Considering these plots, it is simple to decide that stringers 2 and 3 are the closest to the damage location; since on average they violate the threshold from smaller crack lengths. Finally, when considering Figures 4.6 and 4.7, one can infer that damage site is located somewhere close to the centre.



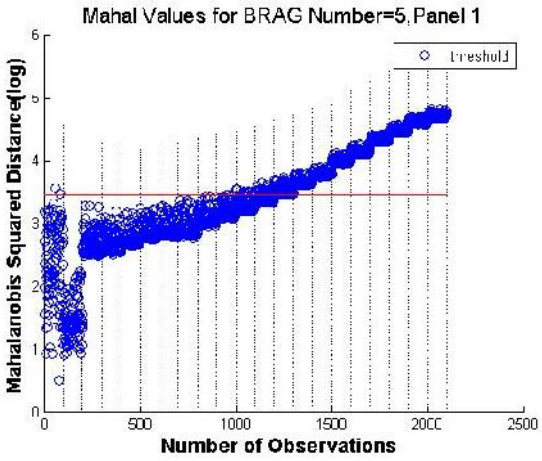



**Figure 4.3 Damage localization - Comparison of novelty indices provided from 5 different datasets. Each dataset is the strain measures of FBG sensor located on one Bragg.**

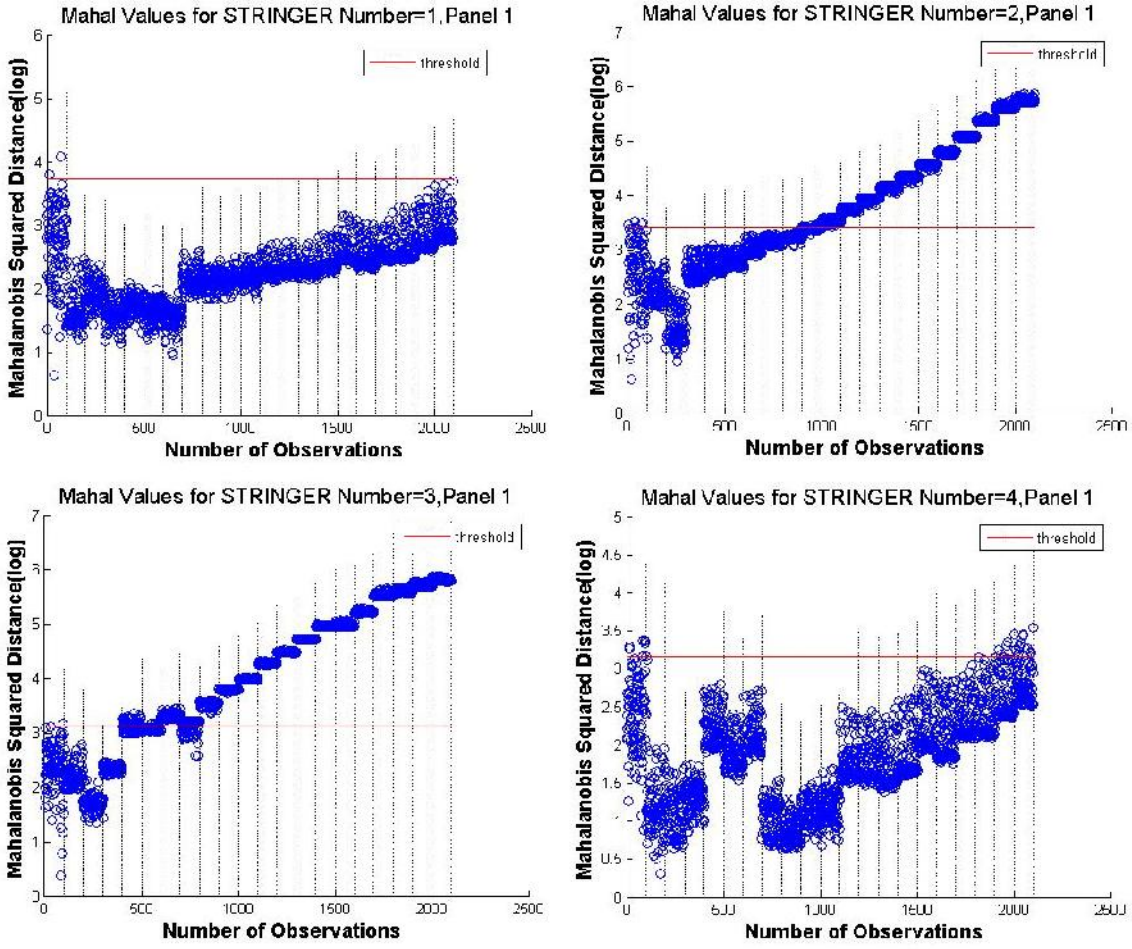



**Figure 4.4 Damage localization - Comparison of novelty indices provided from 4 different datasets. Each dataset is the strain measures of FBG sensor located on one stringer.**

## 4.5 Threshold determination for the panel with the broken stringer

A panel with damage initiated from a stringer is another case of interest. When a stringer fails, the other stringers have to tolerate the extra load, and this extra load is sensed by other sensors which results in a large strain values to be sensed by FBG sensors on other stringers. As a consequence an abrupt increment occurs in damage indices of sensors positioned on remaining stringers. Due to this large change it is expected that novelty indices surpass the threshold immediately after the presence of the crack of smallest measure (Fig. 4.10).

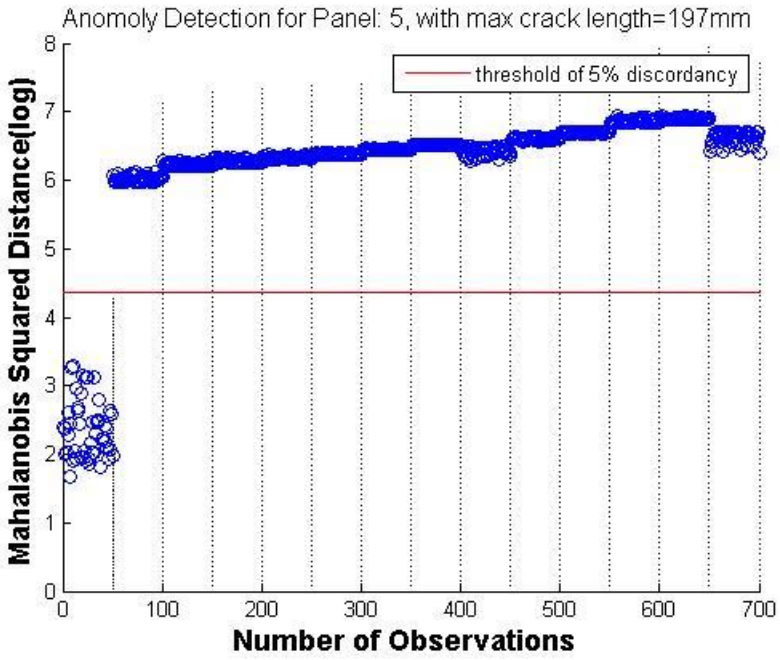



**Figure 4.5- Novelty detection for panel with broken stringer**

# Chapter 5 The machine learning approach (MLP feed-forward)

Artificial Neural Networks (ANN) have been selected as the most appropriate tool for the damage diagnosis, mainly because of their ability to reason on the basis of the experience created during the training phase. This basic knowledge is provided hereafter through finite element simulations, thus reducing the cost of the design phase. The structure of the adopted algorithm is described in **Section 5.7**, while Sections **5.7, 5.8 and 5.9** present the methods adopted in order to optimize the learning process of the artificial neural networks, aimed to increase the robustness of the structural diagnosis. The possibility to predict the SHM performance in both simulated environment and real experimental data acquired during fatigue crack propagation are described in Section **5.10**. The ANN toolbox provided by MATLAB 7.10.0 R2010a has been used hereafter.

## 5.1 Hierarchy of diagnosis algorithm

As it is mentioned in chapter 1, a hierarchy of procedure toward diagnostic and prognostic together can be defined. Anomaly detection (classification), damage localization, damage type identification, assessment of damage extent and prediction consist the different steps in this hierarchy. However, the current case study only concerns the development of a diagnostic algorithm, thereby no study about prediction or prognosis unit has been carried out; furthermore within the diagnostic algorithm damage type identification (which refers to identifying damage whether it is located on a skin or stringer of the panel) is excluded from our study. A hierarchical model based on Neural Network is investigated in order to detect the presence of the damage, to give a measure of intensity of damage and to locate the damage. Two latter steps in the hierarchy will initiate in case the output of anomaly detection confirms a damage presence.

The first step is an ANN employed to identify the presence of an anomaly and, in case of damaged structure; the second step of the diagnostic hierarchy is assigned to characterize the intensity of damage. As it is declared we do not regard the damage type identification in literature of the current case study, therefore we only focus on skin crack scenario, and the stringer failure case is not conducted. However the stringer failure assessment holds an identical path to the one of skin crack. Interested readers can refer to [4].

The Diagnostic Unit can be divided into two macro-steps:

• Anomaly detection level is intended to indicate whether or not there is damage in the structure, and this will be a problem of pattern recognition.

• Following the case of damage detection, the corresponding levels of localization and quantification specify in detail the position and the dimension of the damage. Both levels represent a non-linear regression problem.

## 5.2 Database creation for diagnostic unit

Building a diagnostic unit using MLP neural networks, we need to establish a numerical model, which is used to generate the simulated experience, necessary to train the diagnostic unit. Two databases are available from [4], one for skin crack damage, the other for stringer failure. The latter one is not of concern of the analysis of this chapter.

Concerning skin crack, a total number of 1700 cases have been simulated, increased to 3400 if the symmetry with respect to the vertical axis is considered. In particular, 17 crack lengths have been simulated, ranging from 20mm to 100mm, with 5mm step. 100 crack centre position have been considered at each crack length level, randomly selecting the [x, y] position of the crack centre, within the validation region which is a rectangular region covering the centre of the panel. The borders of the region have $x_1 = 75mm, x_2 = 525mm$, $y_1 = 70mm$, $y_2 = 370mm$ coordinates. It is important to notice that the positions of the simulated crack centers are different at each crack length level, thus allowing more efficient scan of damage domain. The damage index map (normalized strains) to be used for the diagnostic has been calculated for each damage case. Apart from the required computational effort there is no practical restriction of the number of damage examples that one can simulate [4]. However, the minimum crack length is limited to 20mm because of the sensor damage configuration adopted during experiment. In fact, given the fact that the skin crack will be initiated in the centre of the bay the strain sensitivity for a sensor located on stringers will be insignificant for cracks smaller than 20mm. this is clearly shown in **Figure 5.1**, where numerical and experimental damage indices are compared as a function of crack length for the identical sensor-damage configuration of the previous chapters. The efficiency of the FE model to describe the damage effect is evident. Due to the fact that the sensors are located on stringers, the early crack detection is not possible for damage located in the centre of the bay, with a relatively damage index sensitivity practically equal to null when the crack is shorter than 20 mm. This is true for all the strain readings from the sensors closest to the crack site.

## 5.3 Evaluation of two scaling methods and their relevant uncertainties

Assuming that the FE simulation perfectly models the real panel while a particular loading condition applies, it is expected that the strain measures of the experimental tests and simulated environment will be the same in both undamaged and damaged (a crack of certain position and length) scenarios. This of course does not happen in reality as the FE model shows a level of uncertainty in matching real experimental model. As the algorithms of

diagnostic unit are trained with the simulated model in the final scope of inferring the state of a real model, investigating and understanding the extent of this uncertainty is important since it surely influences the inference. To simplify the uncertainty analysis, the sensor readings for both the real experimental panel and the FE model can be grouped into baseline condition measure and the measures of damaged cases.

The uncertainty present in sensor reading relevant to the damaged scenario is the result of various interfering factors as they are discussed in the following. If the FE simulation and the virtual sensor network do not perfectly model the real panel, e.g. incorrect positioning of the virtual sensors, having a geometry property different from the real panel and etc., a nonlinear divergence (as crack evolves) in some sensor can be expected. Another reason for this discrepancy between the FE and the real model comes from the testing condition i.e. presence of some local stress distribution (due to localized loads and local temperature variation) within the panel during crack propagation which influence only few sensors. This testing condition variation only occurs in the real experiment. On the contrary, the condition of simulation environment is totally under control. Another cause to this uncertainty can be referred to the fact that in the real experiment the crack initiation site is somehow controlled but its evolution cannot be controlled; therefore a controlled propagation from the crack initiation spot cannot be guaranteed. This will obviously affect the sensor readings differently from the FEM, where both the crack initiation and the evolution are imposed by the designer.

Concerning the inevitable presence of uncertainty in the sensor readings also for the baseline condition, fewer factors contribute to the uncertainty level. Since strain measures are acquired only once (compared to the damaged case that has to be repeated for different crack lengths) the testing condition variation can be supposed as controlled. Therefore the only causes for discrepancy of the FE model baseline measures from the real one are referred to be first having a FE model which is different from reality, second, bad positioning of the virtual sensors and, third, any environmental mismatch with respect to the modeled baseline condition. The discrepancy present in baseline strain measures can be represented as a difference vector.

At this step it is convenient to perform a calibration in a way that can at least exclude the uncertainty in the baseline (undamaged) strain measures between FE and real model. The baseline signal for both real and simulated model can be simply evaluated by testing them. Therefore the path to calibrate the FE model on the real model at baseline condition is straightforward. This calibration makes use of scaling vector. Two methods for calibration have been suggested and their corresponding performances have been evaluated for both methods. This evaluation is based on their uncertainties and their influence on the performance of diagnostic algorithms. The uncertainty is considered as a discrepancy of real experimental sensor readings from the corresponding readings of the FE model, calculated at any crack length. It was stated before that the only available experimental database is for centre crack propagation. Therefore, the calculated uncertainties regarding each virtual

sensor measures in the FE model, at each crack length, belong to a central damage evolution. Due to the fact that the crack lengths at which the strains have been acquired are different for experimental model and the FE model, interpolations of strain values has to be performed.

It is said before in chapter 3 that sensors are not sensitive to defects of small size. Therefore the normalized strain values, for these small cracks, are expected to be constant (assuming that strain values vary linearly with load). Thereby, after performing the calibration, concerning small crack lengths the only present uncertainty can have its origin in the local stress distribution that might occur in experimental test. Notice that if the meshing property of the FE model has been selected properly it must show a fairly flat trend at these insensible crack lengths **(see Fig. 5.1**).

Considering insensible crack lengths, it was stated that factors “imperfect FE model” and improper virtual sensor network” do not contribute to the uncertainty. Therefore, to take into account all contributing factors, the uncertainty is calculated for sensor readings for cracks greater than 40mm (sensible crack lengths). In this manner all the discrepancy causes are counted in the uncertainty measure and no trivial effort of uncertainty calculations is done.

Before concerning our analysis with uncertainties we shall define a procedure to calibration; in the other words data scaling. In the literature of the previous works done on this case study, a different approach from the current work was taken in defining a scaling method. We shall discuss both methods to illustrate the extent to which the final performance of diagnostic algorithms can be influenced by a proper scaling definition.

After applying both scaling factors to the baseline strains of FE model, obviously the scaling method which provides less uncertainty in the damaged strain pattern corresponds to a better matching of the FEM model to the experimental models. This results, as the ANNs are trained with a more similar FEM model to experimental ones, to a better mapping of trained ANNs and therefore better prediction when dealing with real observations. In the first scaling approach the ratio of baseline readings of each real sensor to the corresponding virtual sensor is calculated to compile a “ratio vector”. In the next step all the strain measures of both baseline and damaged conditions are multiplied by this ratio vector. Therefore, baseline strain values for centre crack propagation of real and simulated models initiate from a same value for a sensor **(Fig. 5.1)**. The elements of the ratio vector are calculated using **Equation 5.1,** where $i$ represents the number of the sensor. Letters $r$ and $s$ stand for real model and simulated model respectively.

$$R_i = \frac{\varepsilon_i^r}{\varepsilon_i^s} \tag{5.1}$$

When applying the second scaling method two scaling vectors are calculated. One of the vectors holds the differential values of real baseline strains from 1, and the other vector has the same property but for the FEM baseline strains. Subtracting the scaling vectors from the

corresponding FEM and real databases, all the undamaged (baseline condition) and damaged strain measures are shifted of a same quantity. In this manner all baseline sensor measures of both FE and real model adopt the value 1 (**Fig 5.2**).

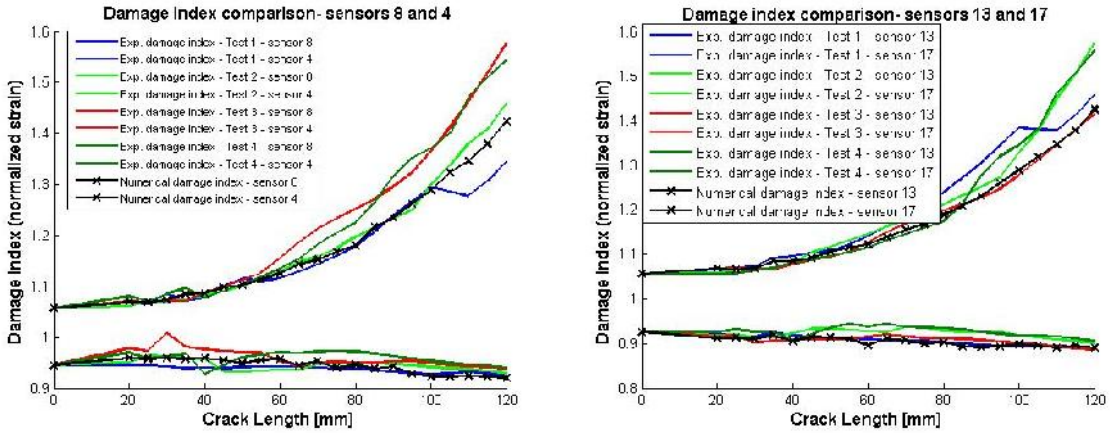



**Figure 5.1 Fem verification by damage indices comparison between the Fem baseline and 4 experimental baselines for sensors 8 and 4 (Left), and sensors 13 and 17 (Right) using first scaling factor**

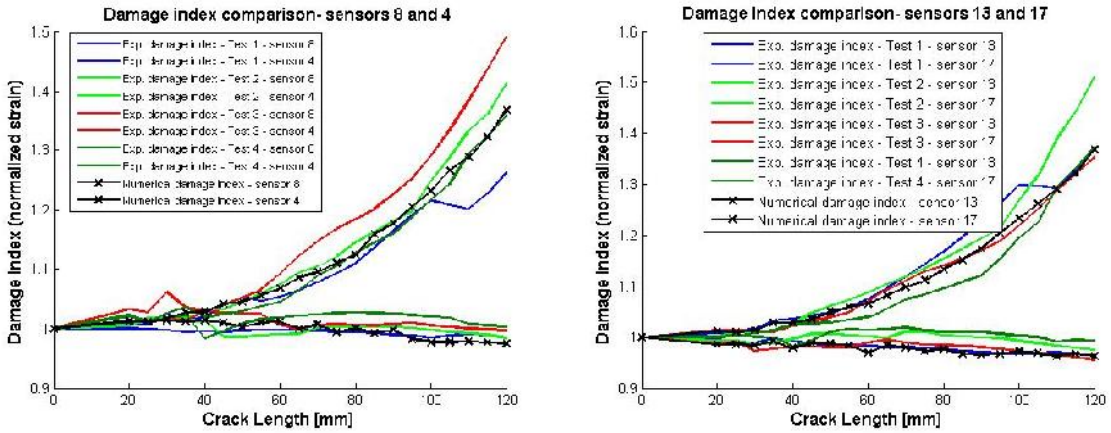



**Figure 5.2 Fem verification by damage indices comparison between the Fem baseline and 4 experimental baselines for sensors 8 and 4 (Left), and sensors 13 and 17 (Right) using second scaling factor**

To obtain the uncertainties, the differences of real model damage indices (strain values) from the FEM damage indices for crack lengths greater than 40mm are calculated obtaining a matrix of uncertainty values. Then, the average values of uncertainties are calculated for each sensor separately. The value of the sensor with the maximum averaged uncertainty percentage (represented as the percentage of nominal strain value) is selected and shown in **table 5.1** for 4 available experimental tests**.** Most probably this maximum uncertainty belongs to the nearest sensors to the damage site. This is because these sensor's outputs contain one more uncertainty (as mentioned before) which is due to the fact that center crack evolution in the real panel does not happen as same as the crack which is modeled in FEM. Thus, the crack presence influences the sensitive sensors of different extent. The last row in the **table 5.1** gives the uncertainty magnitude of any sensor in average.

| Test number | Test 1 | Test 2 | Test 3 | Test 4 |
|---|---|---|---|---|
| Maximum uncertainty-1st scaling factor | 4.49% | 4.70% | 5.00% | 4.37% |
| Maximum uncertainty-2nd scaling factor | 4.60% | 5.83% | 3.34% | 4.1% |
| Average uncertainty-2nd scaling factor | 1.78% | 2.13% | 2.58% | 2.43% |

**Table 5.1 Uncertainty values for two different scaling factors**

As the uncertainties for both scaling methods show similar values **(table 5.1),** this can end up with the decision that there is not a better choice between the two scaling methods. However, when considering the anomaly detection algorithm of the diagnostic unit, the undamaged database is part of the training process. Thus, any change in the baseline strain map can have a notable influence in the performance of the algorithm which is irrelevant to the influence of uncertainty in the training data. As it will be discussed later in the **section 5.4**, the baseline strain values (undamaged database) in the training database are associated to 0 in the output vector during training process. As a matter of fact, when the anomaly detection algorithm is trained with the database which is scaled by implementing the second scaling method, it shows a better performance than algorithm trained with the database provided by first scaling method. No mathematical explanation is proposed here, but **Figure 5.4** might help to understand that visually it is easier to associate the straight line (a vector with elements all equal to 1) to an undamaged case (0 in the output) rather than the zigzag line. In specific, during the training process the MLP neural network adjusts its connection (synapses) weights in a way that it can generalize better during testing.

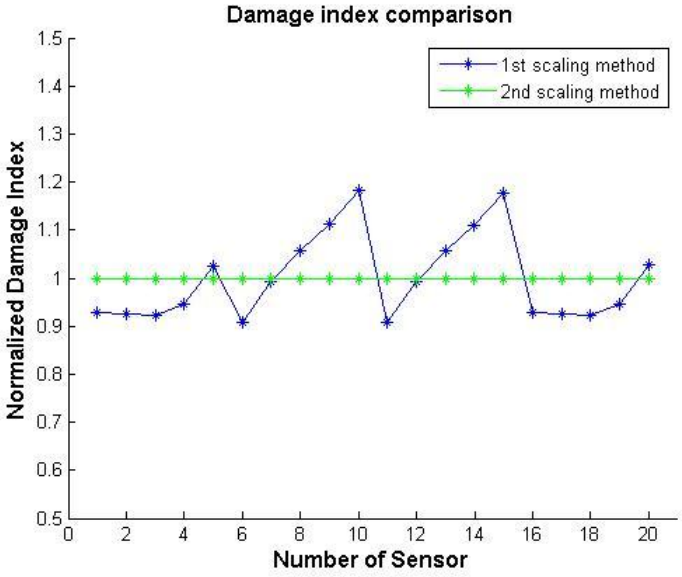



**Figure 5.3 Comparison between undamaged and damage indices provided by two scaling methods. The blue line is obtained when applying 1st scaling method and the green line when applying 2nd one**

To compare the performance of the ANNs trained with databases provided by each scaling method, two classifying plots are represented in **Figure 5.4**. The **Figure 5.4** shows classification algorithm performance where the additive noise **(See Section 5.8.2)** to the training database is 4% in both undamaged and damaged data sets. At this level of additive noise it is evident that the second scaling method provides a better performance for the algorithm to allocate anomaly indices (green line).

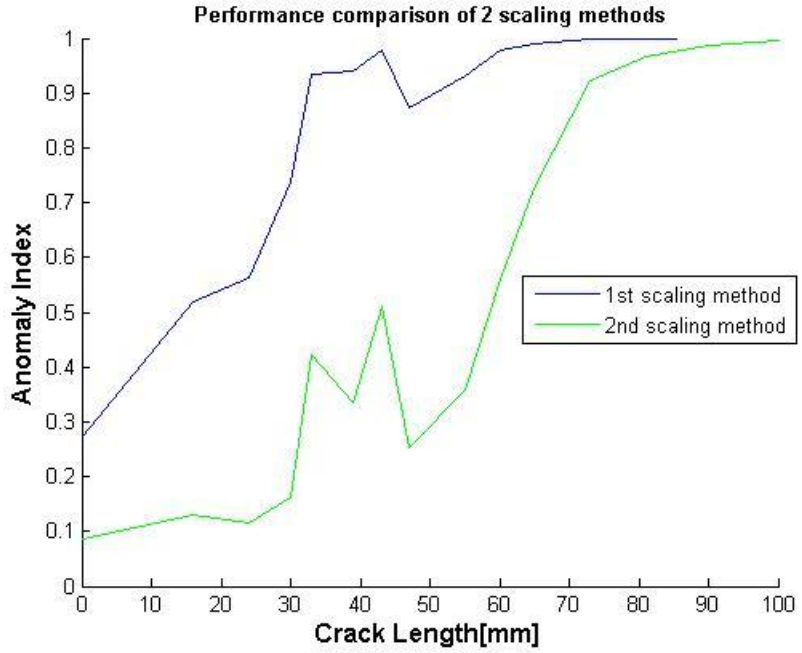



**Figure 5.4 Left: Performance Comparison of two scaling methods when additive noise is 4% in the data- Right: Performance Comparison of two scaling methods when additive noise is 8% in the data**

We do not discuss the performance comparison between quantification and localization algorithms when applying the two different scaling methods. It is enough to mention that even in the assessment context of these two algorithms, it is also proved that second scaling method stands for more reliable predictions, especially when inferring the position and intensity of the cracks of smaller sizes.

## 5.4 Artificial Neural Network definition

To reduce the dimension of the ANN and thus the number of parameters to be optimized during the training process, the inference relative to different damage parameters (position and length) has been divided into separate ANNs, thus allowing for a separate optimization of each diagnostic level. Three standard (**refer to section 5.7**) feed-forward ANNs have been trained to perform the anomaly detection, the localization and the damage quantification respectively. Three types of ANN algorithms have thus been trained and are summarized in **Table 5.2**. One supervised-learning algorithm has been trained for anomaly detection, for

skin crack. The algorithm takes the information on both the undamaged and damaged scenarios during training as input and tries to correctly classify new inputs during testing.

| ID | Diagnostic level | Input data | Output data |
|---|---|---|---|
| Type 1 | Anomaly detection | Normalized strain database for undamaged and skin crack cases | 0 = no alarm, for undamaged patterns; 1= alarm, for skin crack pattern |
| Type 2 | Quantification | Normalized strain database for skin crack damages | Crack length |
| Type 3 | Localization | Normalized strain database for skin crack damages | Crack centre position |

**Table 5.2 General information about input and output of ANNs of different types.**

Anomaly detection algorithm is entitled to distinguish damage index patterns relative to real undamaged condition. The input format is a vector of 20 elements corresponding to the virtual damage index vector. The output format is a [0, 1] class index, being "1" and "0" associated to the alarm and non-alarm cases respectively. In order to provide a more balanced training database (only one healthy case model is available, versus 3400 skin crack damaged cases), the undamaged scenario has been replicated by sampling each virtual sensor measure from a Gaussian distribution, centered on the nominal numerical value of each measure. The characteristic of this Gaussian distribution is interconnected with one of the considered algorithm optimization parameters **(see Section 5.8**) through its standard deviation. Although the FE model has been calibrated on baseline signal (**Section 5.3**), this does not take into account any possible discrepancy between FEM and reality, as the scaling factors practically refer to a particular load, temperature, etc. condition. The characteristic of this Gaussian distribution is thus indicative of the uncertainty due to environmental influences (temperature, loading condition, etc.) and has been based on data from previous studies (**Section 5.3**).

As we decided to focus on algorithms for skin crack characterization, two functions fitting ANNs have been designed to estimate the crack centre position (Type 2) and crack length (Type 3)(see **table 5.2**). The input format for training is again a 20-element vector containing the simulated damage index pattern. Each algorithm receives the simulated experience on both the damage position and dimension variables during training as the input. On the other hand, the output is the simulated crack centre position and crack length respectively. The two algorithms are thus trained with the same database (the entire database for skin crack damage, **Section 5.2)** and are asked to extract information on crack centre position and crack length respectively. Only the data relative to damaged cases have been used during the learning process for damage characterization.

A summary for the input-output format for Type 1 to 3 ANN is reported in **Table 5.3**. In practice, also the damaged scenarios in some cases have been replicated through the addition of a random Gaussian noise, as will be explained in detail in **Section 5.8**.

| ID | Undamaged DB replication | Skin crack DB replication | Input matrix format | Output matrix format | Output |
|---|---|---|---|---|---|
| Type 1 | 2620 | 1 | $20 \times 5240$ | $1 \times 5240$ | Index in [0,1] range |
| Type 2 | 0 | 1 | $20 \times 2620$ | $1 \times 2620$ | Crack length |
| Type 3 | 0 | 1 | $20 \times 2620$ | $2 \times 2620$ | [x ,y] coordinates |

**Table 5.3 Specification on input and output of ANNs of different types**

The adopted training technique was common to all the ANNs and a summary of the parameters defining the ANN structure and the training procedure is reported in **Table 5.4**, while the theory background for ANN is presented in **Section 2.2**. Early stopping criterion is also adopted to guarantee generalization **(Section 2.2.7).**

To end up with more robustly performing diagnostic algorithms both hidden layer and neuron numbers have been optimized (**Section 5.7**). In particular, the input-output dataset has been randomly split into three subsets, namely training, validation and testing sets, containing the 70%, 15% and 15% of the data respectively. The optimal hidden neuron and layer numbers are the one that provides the least Mean Square Error on the validation set (**Section 5.7.1**).

| ID | Input layer nodes | Hidden layer number | Hidden layers nodes | Output layer nodes | Training strategy | Activation function for output node | Activation function for hidden node |
|---|---|---|---|---|---|---|---|
| Type 1 | 20 | 2 | 10 | 1 | Levenberg-Marquardt back propagation | Hyperbolic tangent | Sigmoid |
| Type 2 | 20 | 2 | 10 | 1 | Levenberg-Marquardt back propagation | Hyperbolic tangent | Linear |
| Type 3 | 20 | 2 | 10 | 2 | Levenberg-Marquardt back propagation | Hyperbolic tangent | Linear |

**Table 5.4- Specification on structures, training functions and activation functions for ANNs of different types**

## 5.5 Application of Committee of ANNs for damage diagnosis

**Figure 5.5** shows the effect of ANN training uncertainty for ANN types 1, 2 and 3, relative to anomaly detection, quantification and localization of a skin crack. The results reported in **Figure 5.5** have been obtained by simulating centre propagating crack with the validated FEM (similar to the one adopted during experimental test). The sensor network configuration is identical to the one of the experimental panel (20 sensors have been used to monitor the whole panel). A random Gaussian noise has been added to the simulated strain pattern. The 10% range with respect to the FEM nominal value corresponds to the 95% confidence interval for classification. This percentage value is in compliance with the optimal value of the parameter “additive noise” which is determined later in **section 5.8.1**. The “additive noise level” to the training data is also called as “training noise (TN)”. Considering the

quantification and localization algorithms, this “additive noise” parameter has been set to null which is consistent with the results obtained in **section 5.8.2.** A random Gaussian noise has also been added to the strain pattern which is used to test the algorithms. This strain pattern is sampled from a simulated crack propagation. The 6% range with respect to the FEM nominal value corresponds to the 95% confidence interval for all three types of ANNs. This percentage value is in compliance with the average uncertainty calculated for $2^{nd}$ scaling factor and given in table 1.1.

**Figure 5.5** (upper left plot) demonstrates the relevance of the additional uncertainty introduced by random sampling from a training database. The mean of the 20 ANNs is a suitable estimation of the true committee output (**Section 5.4**) and provides a reasonable guarantee for the repeatability for the ANN output. The alarm threshold has to be set to 0.5 to ensure an unbiased detection resulting in a minimum detectable crack length of around 60mm. Considering the original database unbiased with noise, the minimum detectable crack length is however strictly dependent on the selected geometry, sensor network and damage configuration [4].

**Figure 5.5** (upper right plot) focuses on damage quantification. In crack length estimation, attention is drawn to how the simulated crack propagation is reflected in the quantification committee output. An upper limit of 100mm for crack length has been chosen. This value is based on the availability of the skin crack damage database presented by FEM simulation which is limited by the geometry of the panel. The algorithm performance decreases for smaller crack lengths (simulated cracks from 20mm to 100mm have been included in the training database); as a matter of fact, this is reflected in a close to null sensitivity of the damage index (**Figs. 5.1 and 5.2**) for a 20mm crack located in the centre of panel. However, the quantification algorithm should be run after the generation of the alarm obtained from the anomaly detection level. When the pattern recognition algorithm classifies the structure correctly (detection index equal to or above 0.5), the quantification inference becomes accurate. The uncertainty associated to one single ANN is significant, while the expected values of the plotted distributions (the committee output) accurately predict the target crack length.

Localization has been performed once to estimate normalized X crack position and once to estimate normalized Y crack position for the same centre crack propagation (notice that target position does not exist for a 0mm long crack). Similar to the quantification algorithm, the localization performance manifests a general improvement while considering more sever damages (longer cracks). **Figure 5.5** (lower left and lower right plots) show the uncertainty in the estimated crack centre position related to one single ANN.

In conclusion, it is important to focus on the algorithm structure design. For the purposes of the current diagnostic system, algorithms for damage quantification and localization always give an estimate, independent of the existence of cracks. They should therefore be used in

series with the anomaly detection layer which provides an index indicating the reliability of the assumptions made by the other ANNs [4].

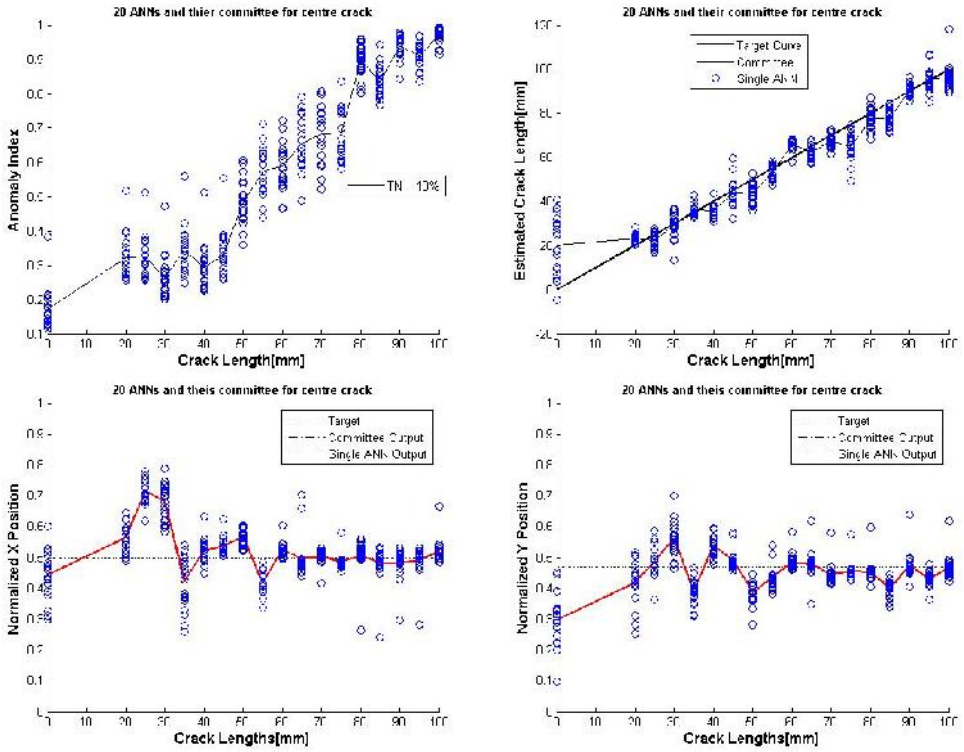



**Figure 5.5 Effect of uncertainty in ANN training for the (upper left Figure) anomaly detection, (upper right Figure) damage quantification, (lower left Figure) damage localization of X coordinate and (lower right Figure) damage localization of Y coordinate. A FEM simulated centre crack is used for testing the algorithms.**

## 5.6 Performance optimization of an artificial neural network algorithm

There exist a variety of interfering factors where a proper choice of each, leads to an optimal performance of a typical feed-forward back-propagation ANN. We shall parameterize these factors and consider theme as variables. In the literature of the feed-forward artificial neural networks, the performance is referred as a better targeting of the algorithm when the algorithm is fed with a new experience. To seize this aim a comprehensive study has to be done, in order to firstly understand what are the influential parameters in ANN performance and secondly to optimize these parameters.

Whatever purpose that an ANN is designed for, one of the crucial factors in the performance of a neural network is its structure, apart from the sufficiency and efficiency of the available training database. Thereby, the very first parameters to be optimized firstly are the number of hidden layers and hidden neurons (in general hidden units) of an ANN structure, and this is discussed in details in **section 5.7**.

Another influential parameter that is treated as an influential parameter in the performance of an ANN is supposed to be an artificially added Gaussian noise to the training database, and we will investigate its impact on ANNs of all three types. As it is discussed before, the training database is provided by simulated experiments in FE model. Therefore, it is assumed that the data are not biased with noise a prior, which is good for the sake of availability of an unbiased database. However, presence of noise of a certain level in the training data sets of ANNs turns to be of a great benefit for classification purpose, as it allows the ANNs to generalize their output better to an objective function (**Section 5.8.1**).

Another factor that we are willing to treat it as an optimization parameter is the training data size used in training phase of ANNs of any type. As it is stated before we are not confined in generating simulated damaged cases of various positions and lengths throughout the panel. Obviously, a greater generated database will correspond to larger time spent in training phase of ANNs of any type. Usually ANNs are provided with unnecessarily large training database in order to keep the safe side in efficiency of training phase. Therefore, for two reasons, it becomes of essential use to have an idea about the sufficient volume of this data generation which grants an acceptable performance of the trained ANNs of any type. First reason is that for the similar ANN training process, that might be required in later analysis, a noticeable amount of time will be saved; second reason is that when simulating other portions of the fuselage (of the similar geometry of the current panel) a great time can be saved in generating adequate number of damage simulations when applying a similar sensor network to that portion. This justifies the necessity of conducting an investigation to realize the optimal data size in training phase as it is carried out in **section 5.9**.

## 5.7 Optimal architecture of the feed-forward ANN

To decide on the optimal architecture of an ANN, it is possible to construct a number of ANNs which differ in the number of their hidden unit and check the performance of each one against a target, where this target corresponds to a minimal root mean square error of ANNs. The RMS error is an indication of the performance of an ANN that is easily evaluated in training phase over the training, validation and test data sets (**See section 2.2.7**) while implementing the neural network toolbox of Matlab. Among these three subsets of the original data set, training data set is adopted for the training phase (**See section 2.2.7**); therefore it cannot be considered as an independent data set that can permit a fair judgment on the performance of the ANNs. The subset which will base the performance analysis of an ANN is the validation data set. Keep in mind that performance index (RMS error), which is obtained from the validation set, however is only of use in this section. In the other optimization arguments we shall take other approaches to verify the best performing ANNs.

Since training of a neural network is a time consuming process, a minimal but sufficient number of structures are considered to prove the robustness of the methodology (to acquire

enough knowledge about the performances of the ANNs as their complexity increases) on one hand, and to make it time-justifying in training stage on the other hand. Considering a maximum complexity of an ANN to have 3 layers and 20 neurons, a total number of 60 different structures for each of the classifying, quantifying and localizing artificial neural networks are designed.

To realize a trend, RMS error versus structure complexity of ANN is presented in the **Figure 5.6** for quantification and localization ANNs**.** Three different curves in each plot represent ANNs with 1, 2 and 3 layers where each varies in its number of neurons from 1 to 20. Considering a maximum number of 3 layers and 20 neurons for the structure of an ANN, a total number of 60 different structures for each of the classifying, quantifying and localizing artificial neural networks are designed.

The performance is dependent on both number of hidden layers and number of neurons. For the ANNs of 1, 2 and 3 layers, generally, the performance is enhanced as the number of neurons increases in hidden layers. Based on this one can decide on more complex structure to be assigned to the further analysis, but the fact is that, there is not much of the performance difference between ANN having 2 hidden layers and 10 neurons and an ANN having 3 hidden layers and 20 neurons in each. So, there is a need to compromise between the complexity of the chosen optimal structure and it's inevitably increasing training time in the afterward analysis. To simplify our analysis, in our case study, one same optimal structure for pattern recognition and function fitting has been selected which consists of 2 hidden layers and 10 hidden nodes in each.

It is worth to mention that the reason why the curves do not show a smooth trend, as the number of hidden neurons increases, is because the split of the original data set (data set acquired from the simulated environment to train ANNs) into training, validation and test sets is carried out randomly, and also the initial synapses weights are adjusted randomly. This random selection of validation dataset results to different RMS error even for two ANNs of the same structure. In **Figure 5.6** to provide a smoother curve one can train 10 ANNs of each structure and make an average of RMS error of it, so the effect of the randomness of data set split and weight adjustment are attenuated. This approach has been neglected since this will be a time consuming approach to train an overall 1800 ANNs.

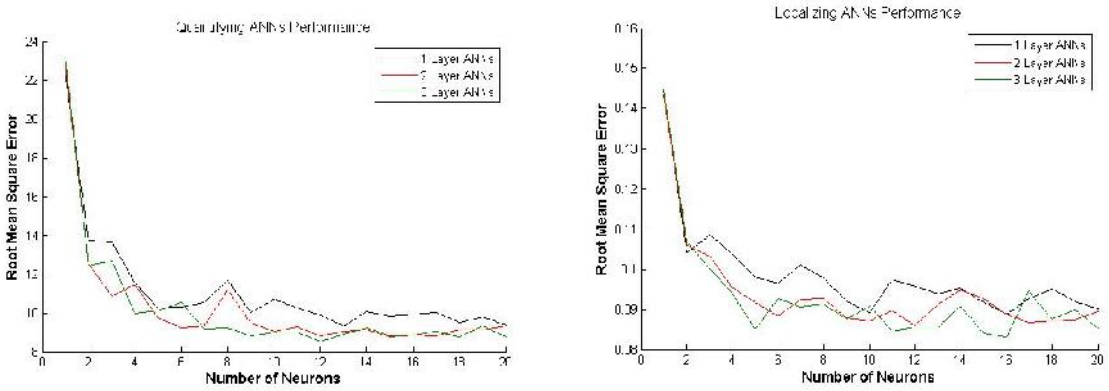



**Figure 5.6 RMS error evaluation for ANNs of different structures calculated for quantifying algorithms (Left) and localizing algorithm (Right)**

### 5.7.1 Statistical evaluation of RMS error of best performing ANNs

The artificial neural network can be employed for three major application fields; function fitting, pattern recognition and data clustering. In the current case study the neural networks are used for two first above-mentioned purposes. In the literature of this work pattern recognition ANNs are referred as "classification" ANNs, and function fitting ANNs are used for damage "localization" and "quantification" purposes. The classification ANNs are employed to assign an anomaly index to each new input where this index varies between 0 (no crack exists) and 1 (presence of crack), and this allows for understanding the integrity state of the structure. The quantification and localization ANNs are used in series with the anomaly detection layer and aim to quantify and localize the existing crack respectively.

For classification networks the RMS error of best classifying ANN is 0.16 **(see Table 5.5)**, which correspond to 16% deviation (in average) in the ANN predictions from the true index of the class (0 for undamaged and 1 for damaged). This RMS error for a baseline signal and for large crack lengths tends to null and for the small cracks this error can adopt values much greater than 16%. However, this indeed is a very good precision of prediction for classification algorithm. This actually is not the case when the simulated data used in training has to be biased with a level of noise (**Section 5.8**) to satisfy a defined objective (**see Section 5.8.1**) in the context of classification. Furthermore, the ANNs are going to be tested with simulated crack propagations which are biased with a level of noise which obviously increases the RMS error in predictions. This noise percentage is in compliance with the uncertainty measured on real test with respect to the FEM in presence of damage and after model calibration (See **Table 5.4**).

For quantification, the RMS error in predicting the length of the crack inside the panel inferred by the best performing ANN is 8.52 mm. So, during the evolution of the crack from 20 to 100 mm the deviation of about 8.5 mm is expected. This is different from reality where we again have to count for uncertainties as we did for classification.

| ID | Optimal number of hidden layers | Optimal number of number of neurons | RMS error over validation set |
|---|---|---|---|
| 1 | 2 | 19 | 0.16 |
| 2 | 3 | 16 | $\cong 0.08 \times 600$ =48 mm |
| 3 | 3 | 12 | 8.52 mm |

**Table 5.5 The best performing ANNs based on their structures for Classification, Localization and Quantification algorithms and their relevant RMSE error which has been calculated over the validation set of the training database**

For localization, since both X and Y values of the crack position are normalized by the side lengths of the panel, the dimensionless RMS error is multiplied by the length of the panel to have the final error in millimeters. This is not a precise approach but an approximation, because the error value belongs to both x and y predictions, and we only use the length of the panel to convert it to millimeters. Finally, in position estimate an error of about 48mm can be expected from the best performing ANN, where the uncertainties are excluded from this evaluation.

## 5.8 Algorithm performance optimization by Gaussian noise addition

As discussed in **section 5.2** a number of factors contribute to the presence of an uncertainty in FE model simulation to fit the real model. One of these factors was identified as local stress distribution present within panel which comes from environmental influences. In the real operational condition, each sensor installed on the fuselage can undergo a level a local stress distribution that manifests as noise in strain measures which is often uncorrelated among different sensors, thus generating "unexpected" strain maps. This makes the interpretation requirement for the ANN a challenge. Moreover, many training patterns are needed to correctly represent complex functions with ANNs and the addition of noise can be regarded as a methodology for regularization, allowing the ANN to generalize beyond the training set. In this section we treat the artificially added noise to FEM database as a variable that can optimize the performance of classification, quantification and localization algorithms (later we will see that noise addition does not introduce any contribution to the performance of the second and third level algorithms of diagnostics). This approach has been taken because the function describing the relationship between the strain map and damage parameters is difficult to approximate. This additional noise can also count for uncertainties which are the result of other factors; i.e. unavailability of a perfect FE model and imprecise configuration of the virtual sensor network. Thus, noise is another parameter that needs to be defined (Eq. 5.2) and optimized [4].

$$\varepsilon_{noised} = \varepsilon_{nom}\left(1 + \frac{\%noise}{2\times100}.r\right) \tag{5.2}$$

Equation 5.2 relates artificially noised strain $\varepsilon_{noised}$ to the FEM nominal value $\varepsilon_{nom}$. $\%noise$ is the percentage noise with respect to FEM nominal strain and r is a random number sampled from a Gaussian distribution with zero mean and unitary standard deviation. The division by 2 is related to the association of a 95% confidence to the specified percentage $\%noise$.

### 5.8.1 Objective definition for classification algorithm

As it is discussed before the strain measures are not sensitive to the defect presence if the crack is short in length. It corresponds to strain overlaps between the strain readings of baseline condition and small cracks. However, the quantity of this "small crack" is strongly dependent on crack configuration and sensor network as it was mentioned in **section 5.5**. When training a MLP neural network with such data in the scope of pattern recognition, due to overlap in two classes (undamaged and damaged), misclassification in the later testing of the ANN is inevitable (**see Fig. 5.4**). However, it is possible to achieve perfect separation of the training data by having a highly flexible model (complex model) as decision boundary, but this does not guarantee a good separation of the testing data. In fact a highly complex model can cause a poorer generalization performance than a model with intermediate complexity [13].

To overcome the problem we shall redefine what is considered as damaged state. To do so, the panel is considered faulty for cracks larger than 60mm and under this value the panel is considered as healthy. With this assumption, a perfect classifying algorithm is expected to release an alarm (in ANN testing phase) when the crack length has a value equal or greater than 60mm. In the other words, ANN has to adjust an anomaly index above the threshold (**0.5, see section 5.5**) when the strain reading belongs to a crack larger than 60mm. Actually by setting the alarming threshold to 0.5 and minimum detectible crack lengths to 60mm, we have indicated a objective against which the generalization performance of the classifying algorithm can be evaluated in the next section.

### 5.8.2 Procedure to Classification algorithm optimization using additional noise

Before discussing performance optimization of the classifying ANNs by any parameters of section 5.6, we briefly explain an initiative that can enhance the reliability of the final algorithm. It is important to verify the performance of diagnostic algorithms of any type, for all possible crack propagation scenarios (cracks of different sizes and positions). In optimizing the ANN performance with "additive noise" parameter, once that the parameter selects a new value, the training process is repeated. The trained ANN, then, is tested with crack propagation of certain position including crack length variability from 0 to 100mm. This testing is done to provide another parameter that can be evaluated against the objective defined in **section 5.8.1**. However, it is possible to elevate the reliability of the methodology

by introducing a variability to the position of the candidate crack propagation, which are used for testing. To make this happen 9 simulated crack propagations of different positions **(see Fig. 5.8)** are generated through the FE model. Again a Gaussian noise is added to the data sets of crack propagations. The 6% range with respect to the nominal strain value corresponds to 95% confidence interval. This percentage value is in compliance with the uncertainty percentage present in FEM model and given in **table 5.1.** As a result of noise addition, a completely new strain pattern is generated. Therefore, the noise addition can be adopted as a strategy to generate more strain pattern samples from a same crack position. In this manner, more crack propagations are available that can be used in algorithm testing. This allows for a more robust evaluation of the performance.

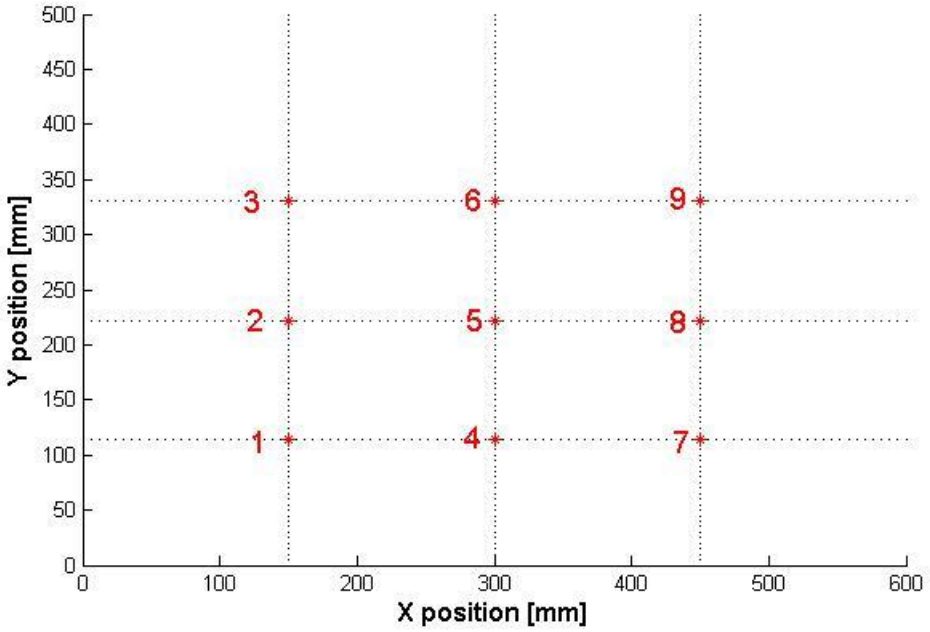



**Figure 5.7 Positions of crack propagation chosen for testing diagnostic algorithm for classification, quantification and localization; 9 crack propagations simulated in FEM**

The anomaly detection for single ANN and committee is a supervised-learning classifier. Its classification is essentially based on damage index sensitivity, which is clearly visible in **Figures 5.1 and 5.2** and small cracks can therefore easily be confused with the undamaged case (missed detection) and vice versa (false alarm). The addition of noise during the training phase masks small damages, due to the fact that the effect of a crack on the strain field has the same order of magnitude as the noise itself. Consequently, only pronounced damage will contribute significantly to the synapses weight optimization during training and will produce a robust output curve, at least for the anomaly detection inference.

Here, a methodology is adopted that checks the performance of classifying ANNs by producing three probability parameters. Probability of false alarm (PFA), probability of

missing event (PME) and the probability of wrong decision (PWD) are the parameters upon which we judge the performance. As it is mentioned in **section 5.4,** a classifying **(**anomaly detection**)** algorithm allocates an anomaly index between 0 and 1 to each sensor reading.

After designing a classifying network, it is tested with a crack propagation e.g. crack number 5 (**Fig. 5.7**). Each crack propagation data set has 20 dimensions corresponding to 20 virtual sensors. Each virtual strain pattern relevant to a simulated crack propagation has been replicated 5 times by sampling each virtual sensor measure from a Gaussian distribution. The classification algorithm is then tested with all 5 replicated strain patterns, producing 5 anomaly indices for each crack length. Based on the defined objective of the classification, it is an ideal result to have all anomaly indices below threshold for cracks smaller than 60mm and above threshold for cracks larger than 60mm. However, some misclassifications will obviously occur for the classifying ANN.

PFA is the probability of having outputs above the threshold for the cracks smaller than 60mm. In a similar way one can define PME which counts for probability of having outputs below the threshold for the cracks larger than 60mm. Having PFA and PME, it is straightforward to calculate the PWD, which is the average of PFA and PME **(see Fig. 5.9)**. Once the optimization parameter “additive noise” has been set to a new value (addition of Gaussian noise to the training database), the training process is repeated. PFA, PME and PWD are also calculated again, after testing the ANN with all 5 replicated strain patterns. Having calculated PWD for different levels of “additive noise” in training data, it is simple to decide the optimal value of the parameter, which is coincident to the minimal value of PWD. **Figure 5.8** shows an optimal additive noise level of 10% at which the minimal value of PWD is obtained. This means that the optimal additive Gaussian noise, with 95% confidence interval, ranges 10% of the nominal strain value. Accordingly, the presence of this level of normally distributed noise, contributes the classification ANN to generalize better to the objective defined in **section 5.8.1.**

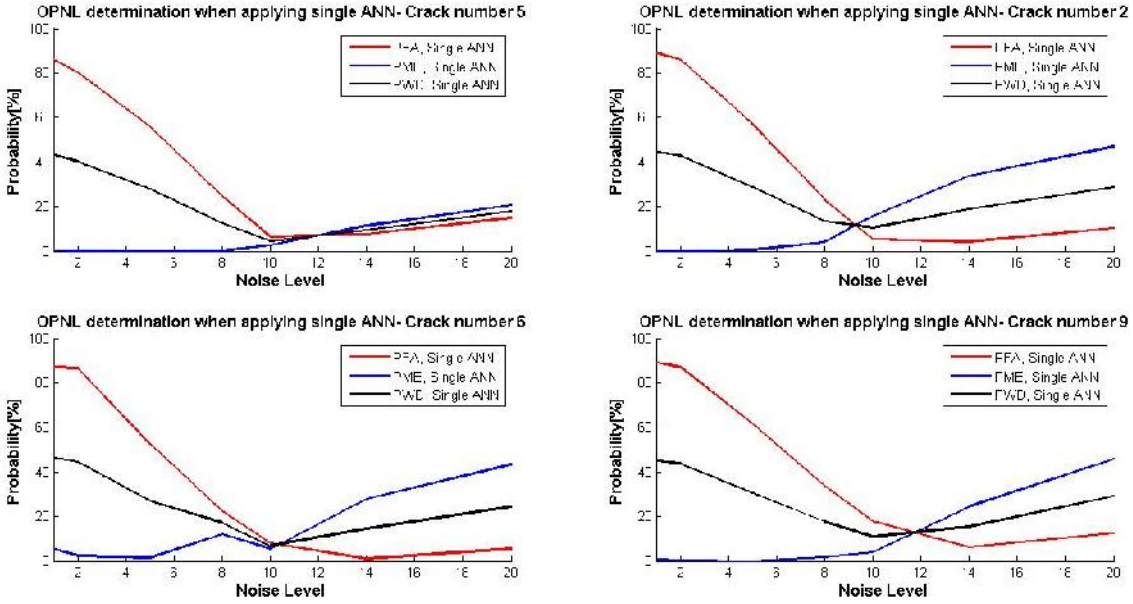



**Figure 5.8 Optimal noise level (OPNL) determination. PFA, PME and PWD calculated for anomaly detection (classification) algorithm when applying a single neural network. Algorithm performance calculated for crack propagation positions number 2, 5, 6 and 9**

The performance of a single ANN was studied in the beginning of this section. Concerning the application of committee of MLP neural networks, an average of 20 network predictions is used as the committee inference. Thus, the parameters PFA, PME and PWD can be calculated in an identical manner to the one of the single network. The PWD is calculated for the committee of 20 ANNs and it again shows a minimum when the additive noise level is about 10% **(Figure 5.9).**

To have a better understanding of the generalization performance of the algorithm at all possible crack positions, the calculated PWD is averaged over 9 crack positions. These average PWD values, for both single ANN and committee of ANNs, are given in **table 5.6** for different levels of parameter "additive noise". In this average mode of PWD it is clear that on average an additive noise level of 10% in training data provides the best performance for classification ANNs to generalize better to the objective defined in **section 5.8.1.**

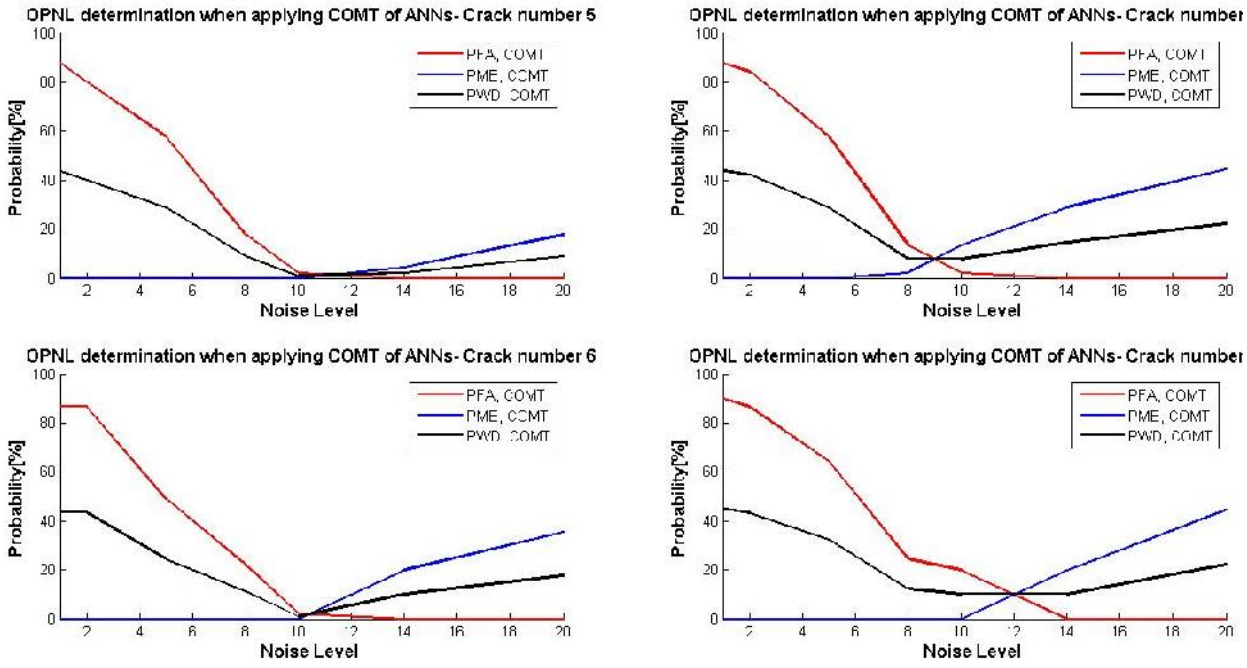



**Figure 5.9 PFA, PME and PWD calculated for anomaly detection (classification) algorithm when applying committee of 20 networks. Algorithm performance calculated for crack propagation positions number 2, 5, 6 and 9**

| Noise level [%] | 0 | 2 | 5 | 8 | 10 | 14 | 20 |
|---|---|---|---|---|---|---|---|
| Average PWD of single ANN [%] | 49.60 | 44.15 | 27.09 | 17.27 | 10.87 | 21.35 | 32.92 |
| Average PWD of COMT [%] | 49.63 | 44.94 | 26.67 | 11.36 | 7.53 | 17.90 | 29.01 |

**Table 5.6 The average probability of wrong decision (PWD) for single network and a committee of networks is calculated and represented versus the additional noise level**

The graphical representation of the two averaged PWD parameters versus noise level in the training data is given in **Figure 5.10.** As these two parameters are indices of generalization performance of classifying ANNs, one can easily decide on 10% of additional noise as it coincides with a minimum PWD when applying both of single ANN and committee of ANNs.

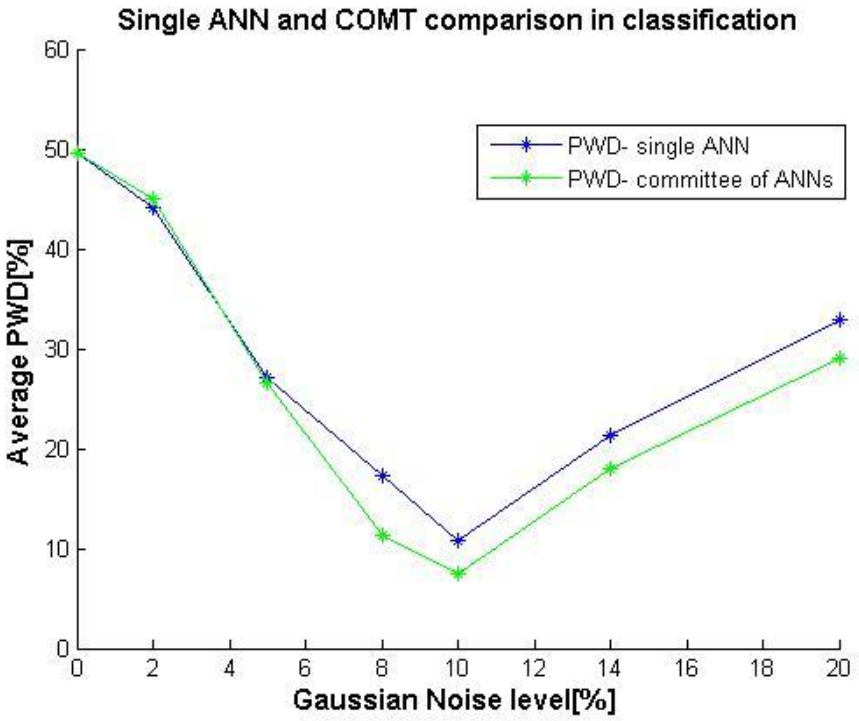



**Figure 5.10 Average probabilities of wrong decision (PWD) which is calculated by averaging the PWDs of both single ANN and committee of networks of 9 crack positions**

### 5.8.3 Procedure to Quantification and Localization algorithm optimization using additional noise

In classification we decided to define an objective against which the generalization performance of the classifying algorithm was evaluated. However, in working with quantification and localization algorithms the procedure is routine, where we simply set the objective to minimization of RMS error of the designed ANNs.

Two parameters have been used for quantification and localization performance analysis in a simulated environment. First parameter offers an index of single neural network performance. To do so, 100 strain readings are simulated in randomly selected positions. This is repeated for each crack length. Then the root mean square error of 20 ANNs is calculated separately, for each crack length, over the 100 strain readings. Finally, the average of 20 RMSE values for each crack length is taken to have a more robust assessment of the performance of a single ANN. This is accomplished by using **Eq. 5.2**

$$RMSE_t = \frac{1}{N_{ann}} \sum_{i=1}^{N_{ann}} \sqrt{\frac{\sum_{j=1:}^{N_{obs}} \left[ (x_{ji} - x_{jt})^2 + (y_{ji} - y_{jt})^2 \right]}{N_{obs}}}$$

for localization where x and y are the coordinates of the estimated crack centre position, $X_{jt}$ and $Y_{jt}$ are the coordinates of the target crack position, $N_{obs}$ referrers to the total number of strain readings selected for each crack length and $N_{ann}$ is the total number of ANNs belonging to the damage localization and quantification ANNs. The parameter

$$RMSE_t = \frac{1}{N_{ann}} \sum_{i=1}^{N_{ann}} \sqrt{\frac{\sum_{j=1:}^{N_{obs}} \left[ (cl_{ji} - cl_t)^2 \right]}{N_{obs}}}$$

is used to check the performance of the quantifying ANNs where $cl$ and $cl_t$ are estimated crack length and the target crack length respectively. Results are reported for both quantification and localization in **Figure 5.11** (left plot) and **Figure 5.11** (right plot) respectively**.**

The second parameter concerns the performance of committee of ANNs (**see Section 5.4)** and is expressed through **Eq. 5.4** and **Eq. 5.5.** It is an error in the committee output indicating the distance of the committee estimated crack centre from the target position for localization and the difference of the committee estimated crack length and the target crack length for quantification. Results are reported for both localization and quantification **Figure 5.12** (left plot) and **Figure 5.12** (right plot) respectively.

$$\Theta_t = \sqrt{\frac{\sum_{j=1}^{N_{obs}} \left[ \left( \left( \frac{\sum_{i=1}^{N_{ann}} x_{ij}}{N_{ann}} \right) - x_{tj} \right)^2 + \left( \left( \frac{\sum_{i=1}^{N_{ann}} y_{ij}}{N_{ann}} \right) - y_{tj} \right)^2 \right]}{N_{obs}}}$$

$$\Theta_t = \sqrt{\frac{\sum_{j=1}^{N_{obs}} \left( \left( \frac{\sum_{i=1}^{N_{ann}} cl_{ij}}{N_{ann}} \right) - cl_t \right)^2}{N_{obs}}}$$

Where

$$\left( \frac{\sum_{i=1}^{N_{ann}} cl_{ij}}{N_{ann}} \right)$$

represents the output of committee of ANNs.

Regardless of the parameter “additive noise”, a small increase in the precision of the ANN is evident for both parameters as the crack length increases; nevertheless this tendency is not stable over the entire crack length range. To produce non-trivial results, two parameters have not been calculated for the undamaged case for localization (crack length equal to 0mm).

Nevertheless in **Figure 5.11** (left plot) and **Figure 5.12** (left plot) these two parameters are calculated even for 0 crack length for quantification to give an idea of algorithm prediction in an undamaged scenario. Here, it is enough to mention that selected optimal additive noise level (OPNL) for quantification and localization is 0%. The RMSE for localization ANNs has been calculated over normalized X and Y coordinates, therefore, it is dimensionless.

Quantification and localization algorithms essentially perform a multi-dimensional regression on the available simulated data, and the available database contains enough information for the optimization of the algorithm functions.

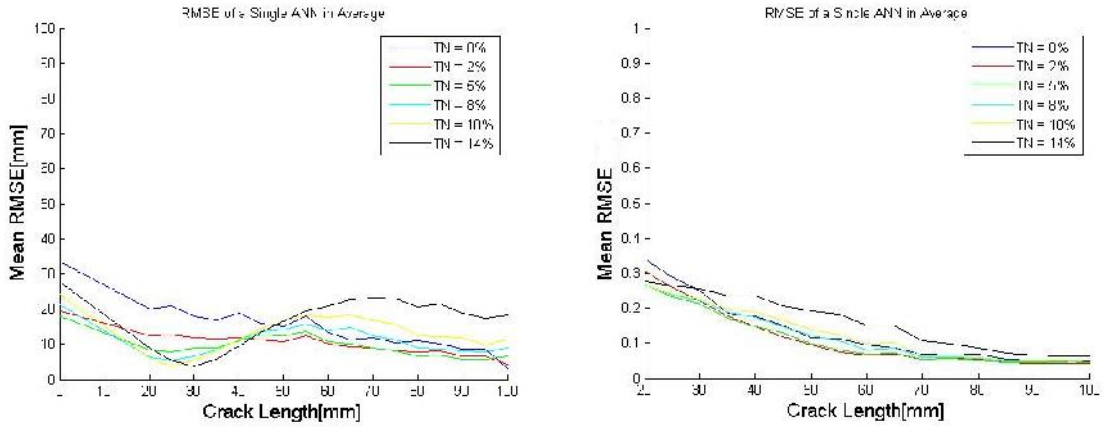



**Figure 5.11 RMS error is calculated as an index of performance of a single ANN as a function of the noise level available in training dataset. This has been calculated for quantification algorithm (Left) and localization algorithm (Right). RMS error calculated over observations of 100 randomly selected crack positions for each certain crack length**

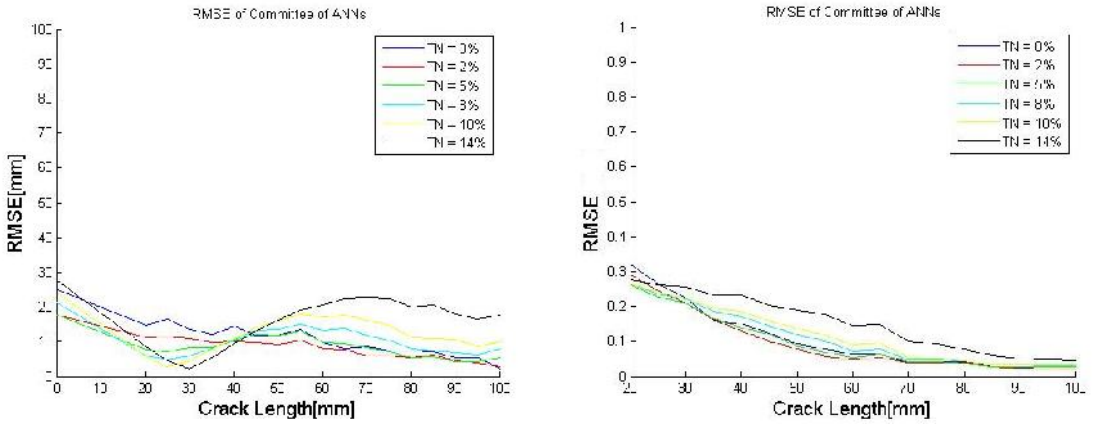



**Figure 5.12 RMS error is calculated as an index of performance of committee of ANNs as a function of the noise level available in training dataset. This has been calculated for quantification algorithm (Left) and localization algorithm (Right). RMS error calculated over observations of 100 randomly selected crack positions for each certain crack length**

## 5.9 Optimizing training database size used in ANNs training

For ANNs of any type, the training database has to contain a good variability, in the sense that it has to be a good representative of all possible damaged configurations that can occur

on the panel. Furthermore, for ANNs of type 1, as said before, in order to provide a more balanced training database the undamaged scenario has been replicated by sampling each virtual sensor measure from a Gaussian distribution. This replication is specifically done for the first level ANNs, since the undamaged scenario is excluded from training phase of type 2 and 3 ANNs.

One can claim that volume of the available training database (**see Section 5.2 and Table 5.3**) is unnecessarily large that costs an extra training time, or in the opposite an inadequate training database resulting to an unreliable diagnostic system. There was no preliminary available methodology for the current case study to determine the ideal training database size. Therefore, it is found convenient to parameterize the concept of the optimal training database size, in a same approach which was conducted to define parameters “OPNL” and “optimal architecture”. The initial assumption is that the primary training data size is unnecessarily large which costs an extra training time. Therefore, the evaluation of the parameter “database size” automatically concerns the reduction of the primary selected database size.

In the next step, a size reduction vector is defined where each element of the vector corresponds to the volume percentage of the primary database that must be randomly eliminated. Starting with 0%, this volume reduction is conducted up to 95%. The 95% data elimination corresponds to 260 sampled strain readings for classification and 130 sampled strain readings for quantification and localization each, which are used as new training databases. As the optimizing parameter adopts a new value, the training phase is repeated over the resized database. Then, the generalization performance of the trained ANN is tested against the objective parameter which is PWD for classification ANN and RMS error for quantification and localization ANNs. In the training process of ANNs of any type, the variability of the parameter “optimal data size” is considered together with the variability of the parameter “OPNL” in training database. In this manner it is possible to count for the changes in the selected optimal noise value that can happen as a consequence of variability of the parameter “optimal database size, DPE”.

### 5.9.1 Optimal training database size for classification networks

To check the generalization performance of the ANN type 1 against the objective parameter defined in **section 5.8.1,** the trained ANN is tested with one of the 9 simulated crack propagations (**Fig. 5.7**), e.g. crack number 5. Testing is conducted for ANNs trained with different additive noise and with 0, 50, 90 and 95% DPE in their training database. The probability parameters PFA, PME and PWD are then calculated. To judge the effect of DPE on the classification performance, the value “PWD” at the optimal noise level is calculated, and this repeated for each DPE level. If a notable increase in the PWD happens at a certain selected DPE, one can conclude that the DPE percentage has been great enough to leave a deteriorating influence in the performance of classifying ANNs.

As it is shown in **Figure 5.13** there is not much of the difference in PWD values for DPE=0% and DPE= 50%, but the difference is evident when comparing PWD of DPE= 0% with those of DPE= 90% and 95%. This simply indicates that even working with half of the available data the algorithm performance does not change, but training time cost can be reduced. In the case of reducing the size of the training database up to 90% the performance definitely declines as the PWD almost doubles.

The same analysis of algorithm performance applies when working in the context of the committee of networks. The only difference is the less vulnerability of the algorithm performance due to the employment of 20 ANN predictions instead of one single ANN prediction (see **Fig. 5.14**). The numerical values of optimal noise at each level of DPE and the corresponding PWD of single classifying ANN and the committee of ANNs are given in **Table 5.7**.

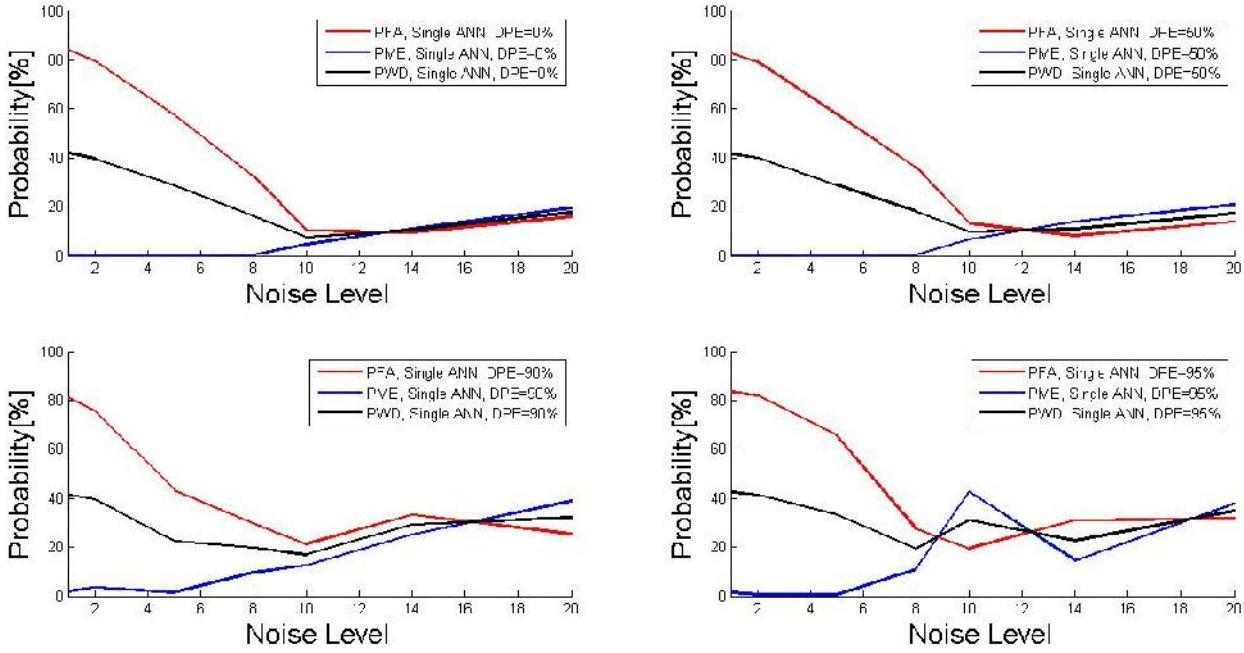



**Figure 5.13 Effect of DPE on the value of PWD for single ANNs**

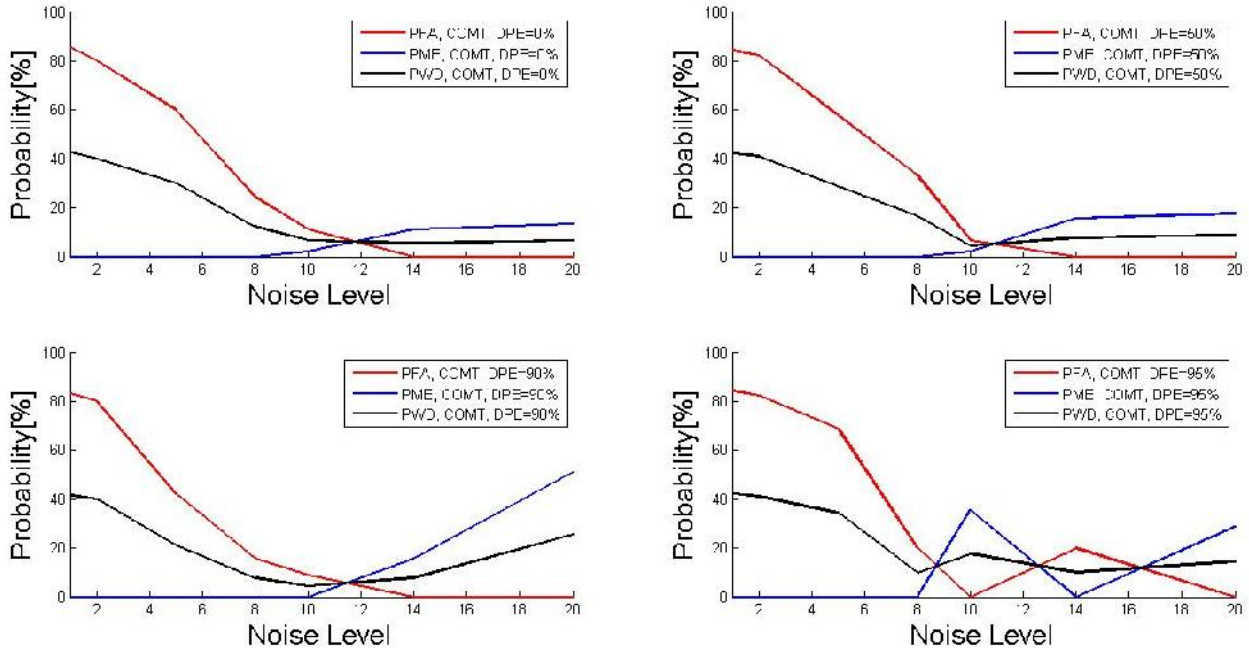



**Figure 5.14 Effect of DPE on the value of PWD for committee ANNs**

| DPE [%] | 0.00 | 0.10 | 0.30 | 0.50 | 0.70 | 0.90 | 0.95 |
|---|---|---|---|---|---|---|---|
| OPNL of both single ANN and COMT[%] | 10 | 10 | 10 | 10 | 10 | 10 | 8 |
| PWD of single ANN [%] | 7.44 | 8.55 | 7.50 | 9.89 | 13.06 | 16.89 | 19.17 |
| PWD of COMT [%] | 3.67 | 2.22 | 4.44 | 4.44 | 5.56 | 6.58 | 10 |

**Table 5.7 Presenting optimal noise level relevant to each DPE and the corresponding PWD committed by single ANNs and committee of ANNs**

### 5.9.2 Optimal database size for training Quantification and Localization networks

In **section 5.8.3,** generalization performances of the ANNs of type 1 and 2 were analyzed, where the minimization of the RMS error was the objective. In **section 5.8.3** the parameter "OPNL" was considered as the only optimization parameter. In this section the influence of the parameter "DPE" on the generalization performance has been investigated (together with the parameter "OPNL"). To this aim, at each level of additive noise (0, 2, 5, 8, 10 and 14%), ANN training process is repeated for different values of DPE (0, 50, 90 and 95%). Then, testing is conducted over the trained ANNs and the corresponding RMS errors are collected. Having the RMSE values as a function of the parameters OPNL and DPE, one can judge the influence of both parameters on the generalization performance simultaneously.

A simple assessment of RMSE versus the parameter OPNL, at any levels of DPE, confirms a null additive noise level as best level of noise addition to the training data (**Fig. 5.15**). This is also true when checking the performance of both quantification and localization ANNs. Thus, the variability of noise parameter can be excluded from our analysis by fixing it to 0%. Thereby, we can confine our attention to the influence of training data size on the performance.

For ANNs trained with different training data sizes, the RMS error can be evaluated at each crack length, by adopting the same approach explained in **section 5.8.3**. Comparing the RMS error values that are obtained at each level of DPE, it is possible to decide upon the maximum data elimination that can be done without compromising the performance. By referring to the **Figures 5.16 and 5.17,** it is easy to agree that the best performance is attained when entire simulated database is used in training phase. This simply corresponds to DPE=0% as best.

Remember that the execution of the quantification and localization algorithms is bounded to existence of alarm in anomaly detection algorithm. This implicitly implies that, in the concern of our performance analysis, RMSE values become important when crack length is larger then 60mm.

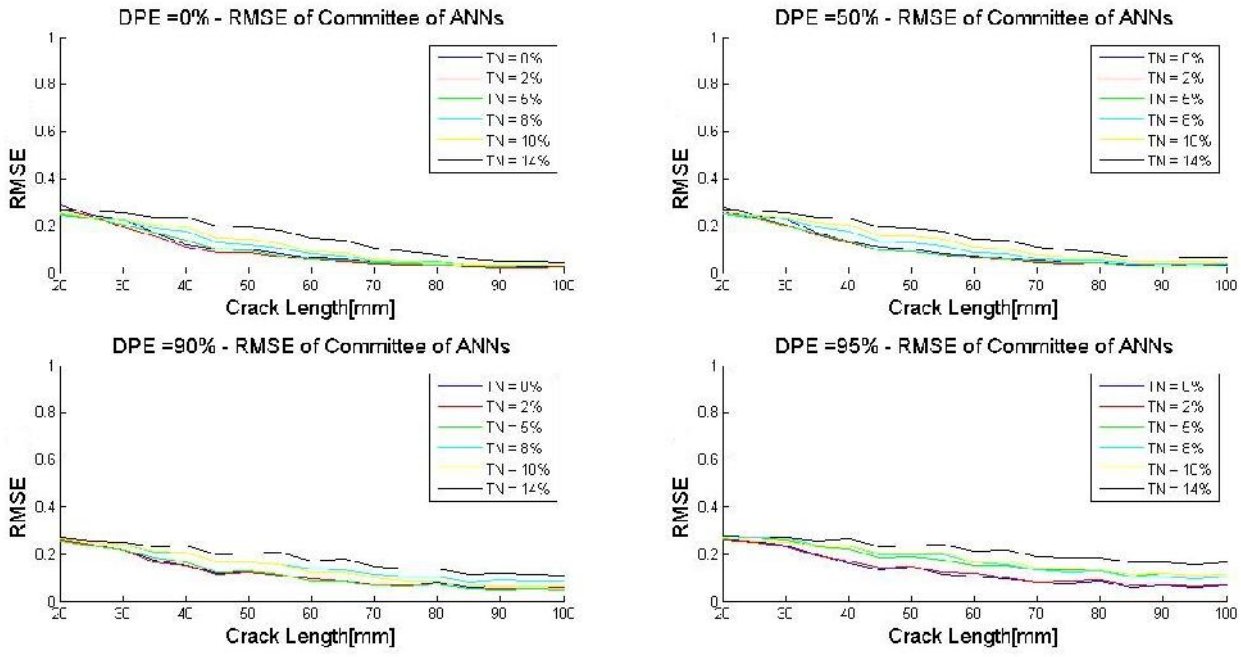



**Figure 5.15 RMS error is calculated as an index of performance of COMT of ANNs as a function of the noise level available in training dataset and the imposed DPE to the original training database size. This has been calculated for localization algorithm. RMS error calculated over observations of 100 randomly selected crack positions for each certain crack length**

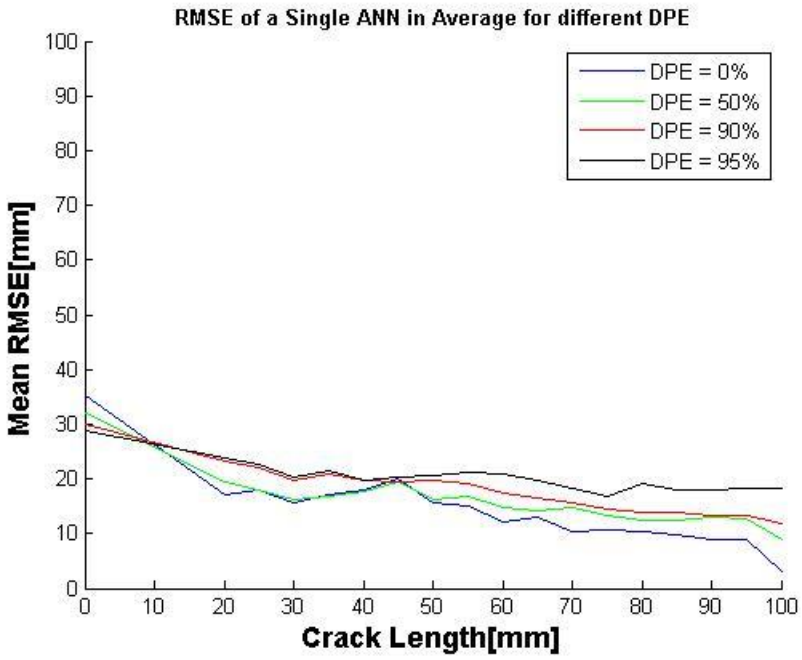



**Figure 5.16 RMS error is calculated as an index of performance of committee of ANNs as a function of imposed DPE to the original training database size. The available noise percentage in training dataset has been set to 0%. This has been calculated for quantification algorithm. RMS error calculated over observations of 100 randomly selected crack positions for each certain crack length.**

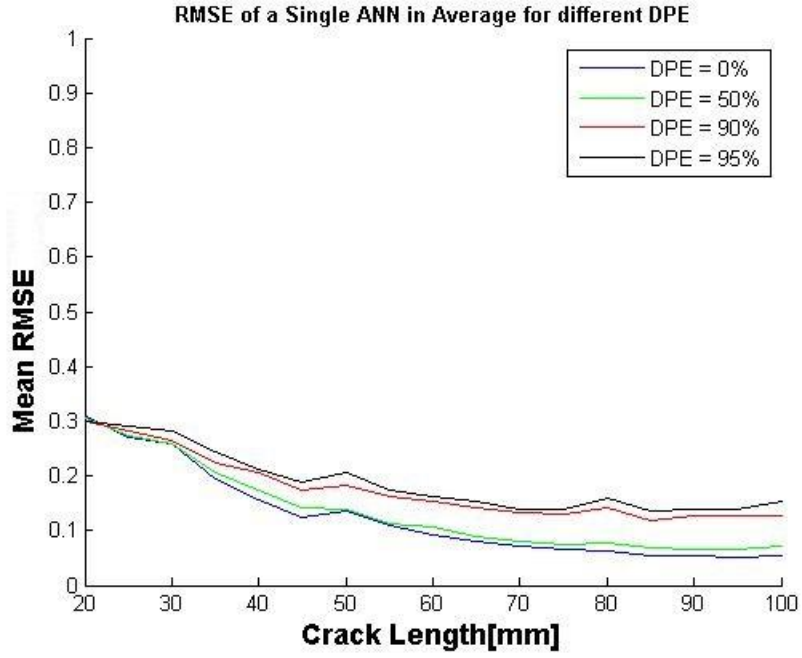



**Figure 5.17 RMS error is calculated as an index of performance of committee of ANNs as a function of imposed DPE to the original training database size. The available noise percentage in training dataset has been set to 0%. This has been calculated for localization algorithm. RMS error calculated over observations of 100 randomly selected crack positions for each certain crack length.**

## 5.10 Representation and evaluation of classification, quantification and localization algorithms in simulated and real experimental environments

In chapters 5.7, 5.8 and 5.9 we focused on the optimization of three types of algorithms which are implemented in diagnostic unit. This was done by considering three optimization parameters: optimal structure, optimal additive noise and optimal size of training database. The final adopted value of each parameter for each network type is reported in **table 5.8**. Having these optimal values, one can construct a number of ANNs to be assigned to the tasks of diagnostic hierarchy (ANNs of type 1, 2 and 3). Up to this section, the mission of designing a diagnostic unit is accomplished. However, it appears to be beneficial to provide the user with some graphical interface of the algorithms inference. It is easier to visually investigate the influence of optimization parameters over the inference performance. Therefore the variations in the two optimization parameters OPNL (also shown by TN) and DPE are included in the graphical representations. To be able to visually judge the inference performance of the algorithms, a target has been set in each plot (e.g. a line, a point, etc.) against which the inference deviation from the target can be seen. All the possible consideration has been taken to provide an easy but comprehensive path to the overall evaluation of the algorithms.

| ID | Hidden layers number | Hidden neurons number | Optimal additive noise | Original training data size | Minimum valid data size- single ANN | Minimum valid data size- COMT |
|---|---|---|---|---|---|---|
| Type 1 | 2 | 10 | 10% | $20 \times 5240$ | $20 \times 2620$ | $20 \times 1572$ |
| Type 2 | 2 | 10 | 0% | $20 \times 2620$ | $20 \times 1310$ | $20 \times 786$ |
| Type 3 | 2 | 10 | 0% | $20 \times 2620$ | $20 \times 1310$ | $20 \times 786$ |

**Table 5.8 Final agreed parameters of designed ANNs.**

The data used in evaluation phase, are the crack propagations from both real and simulated models. Remember it is more important to verify the algorithms with real crack propagation, since the final purpose of the algorithm is to infer in real situations. Therefore, after going through the graphical evaluation of the algorithm with one of the 9 simulated crack propagations (Fig. 5.7), the algorithms are checked with the real centre crack propagation. Note that again a Gaussian noise is added to strain patterns of simulated crack propagations. The 6% range with respect to the nominal strain value corresponds to 95% confidence interval. This percentage value is in compliance with the uncertainty percentage present in FEM model and reported in **table 5.1.**

## 5.10.1 Anomaly detection algorithm testing with a simulated database

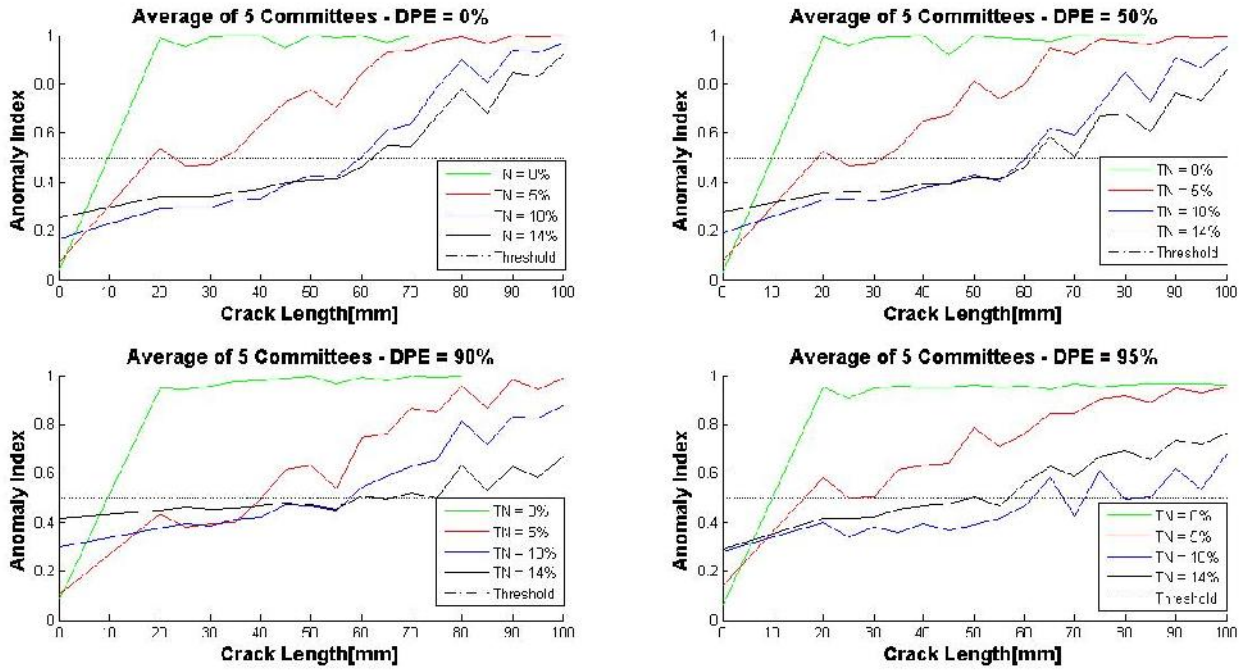



**Figure 5.18 Classification testing of ANNs trained with different additive noise in training data and DPE=0% (upper left), DPE=50% (upper right), DPE=90% (lower left) and DPE=95% (lower right) and tested with simulated centre crack propagation. The predictions are obtained by averaging 5 outputs of COMT.**

## 5.10.2 Anomaly detection algorithm testing with real experimental database

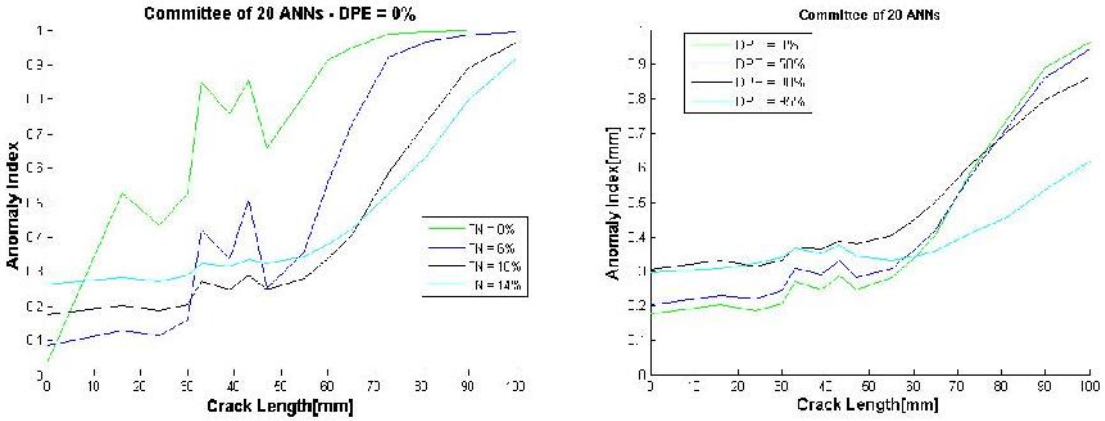



**Figure 5.19 Classification testing of COMT of ANNs trained with different additive noise in training data and tested with real centre crack propagation (left). Classification testing of COMT of ANNs trained with the optimal additive noise (10%) in training data which is checked for DPE in database of 0, 50, 90 and 95% (right).**

## 5.10.3 Quantification algorithm testing with a simulated database

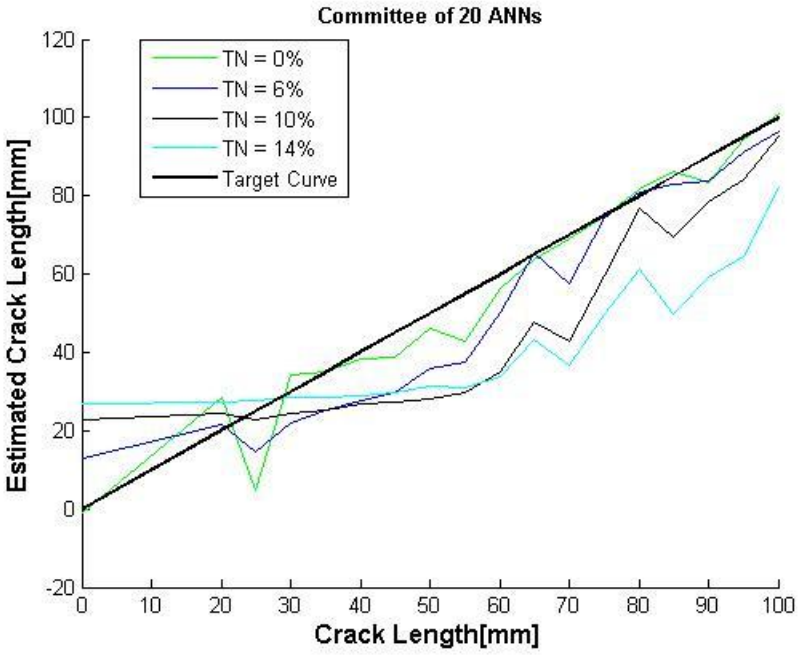



**Figure 5.20 Quantification testing of COMT of ANNs trained with different additive noise in training data and tested with real simulated crack propagation. The straight line is the target.**

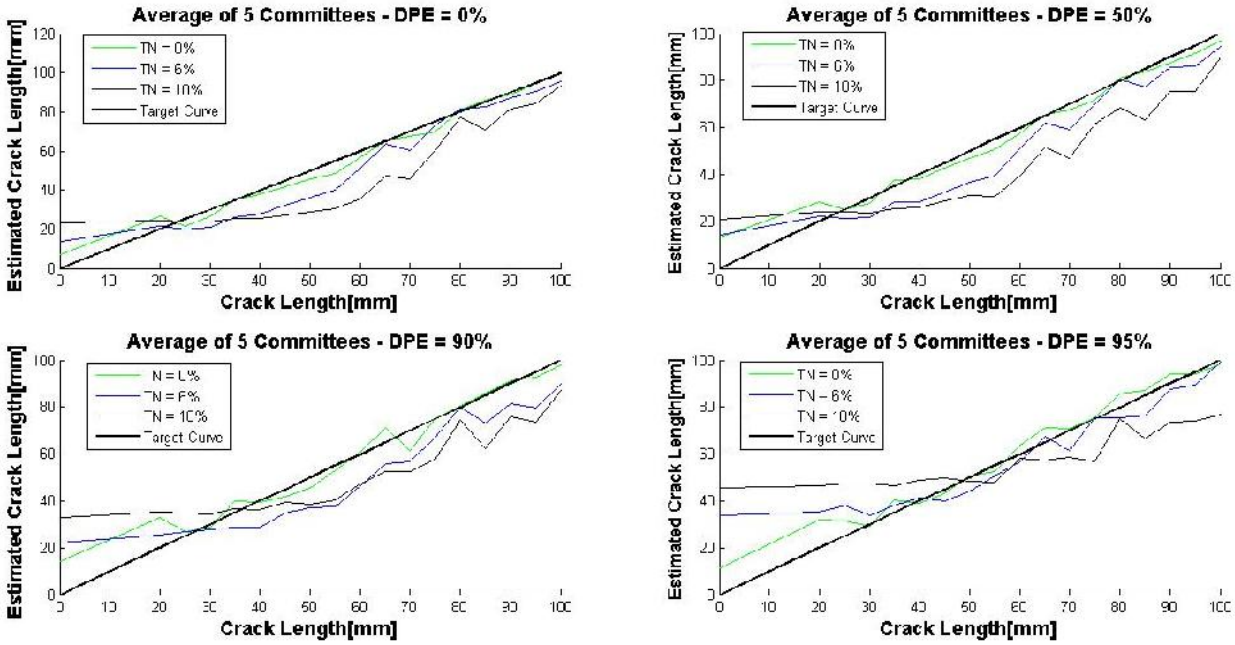



**Figure 5.21 Quantification testing of ANNs trained with different additive noise in training data and DPE=0% (upper left), DPE=50% (upper right), DPE=90% (lower left) and DPE=95% (lower right) and tested with simulated centre crack propagation. The predictions are obtained by averaging 5 outputs of COMT. The straight line is the target.**

## 5.10.4 Quantification algorithm testing with real experimental a database

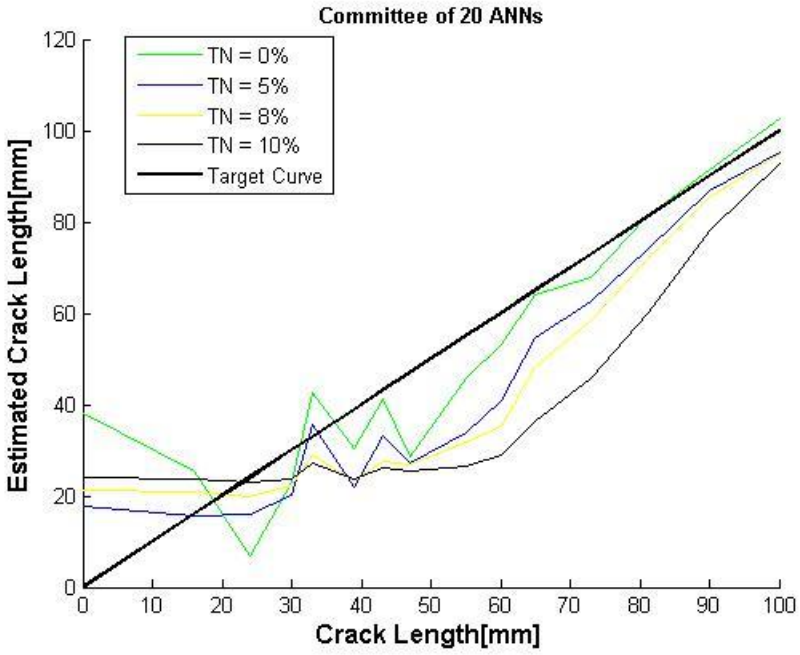



**Figure 5.22 Quantification testing of COMT of ANNs trained with different additive noise in training data and tested with real centre crack propagation (left). DPE=0%.**

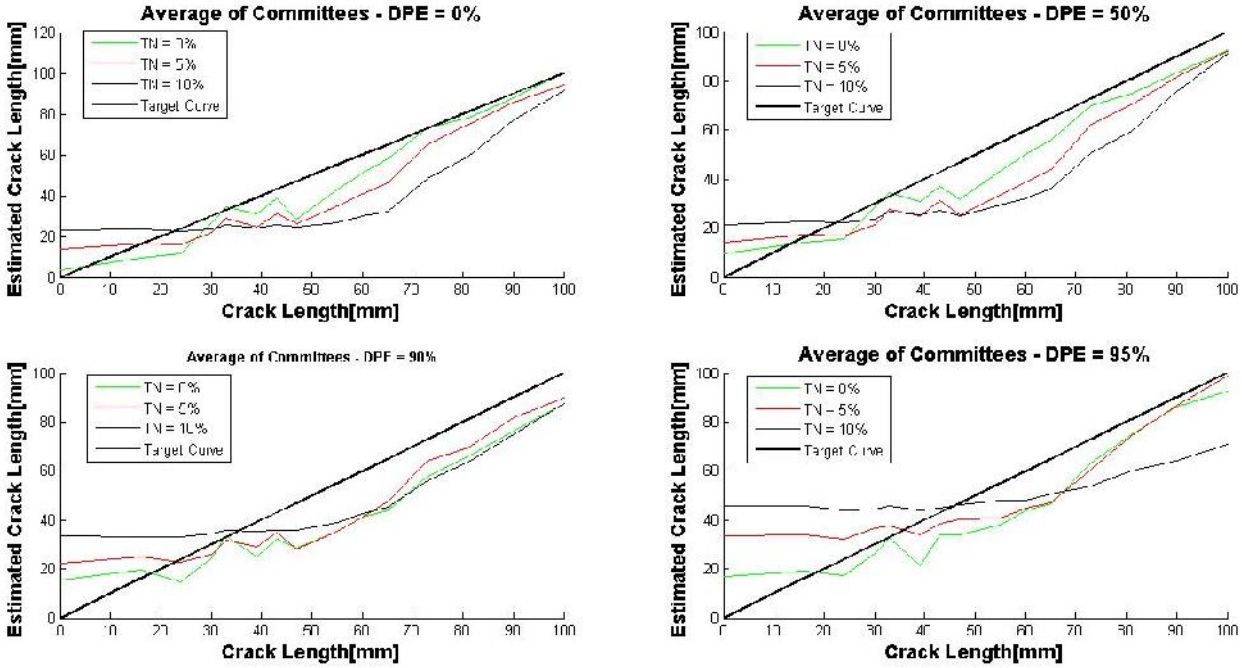



**Figure 5.23 Classification testing of ANNs trained with different additive noise in training data and DPE=0% (upper left), DPE=50% (upper right), DPE=90% (lower left) and DPE=95% (lower right) and tested with real centre crack propagation. The predictions are obtained by averaging 5 outputs of COMT.**

## 5.10.5 Localization algorithm testing with a simulated database

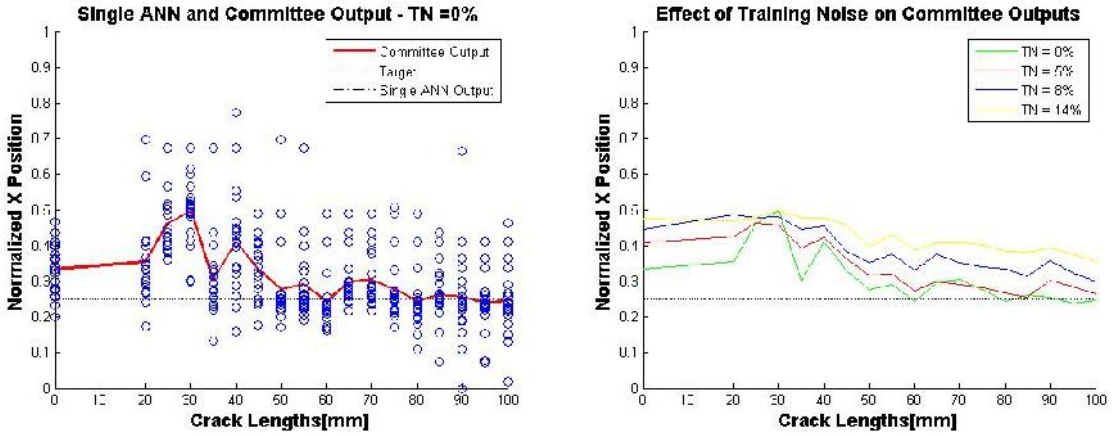



**Figure 5.24 Normalized X position prediction of single ANNs and their COMT with TN=0% and tested with the simulated crack number 3 (left). Normalized X position prediction of COMT of ANNs with different additive noise in training data and tested with simulated crack number 1 (right).**

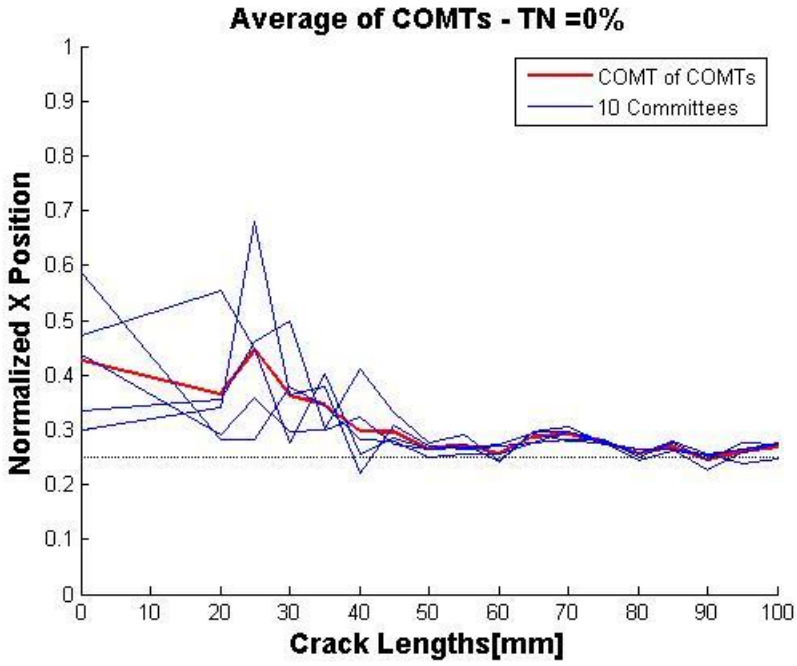



**Figure 5.25 Uncertainty effect in COMT predictions. Normalized X position prediction averaged over 10 COMT outputs with TN=0% and tested with the simulated crack number 1.**

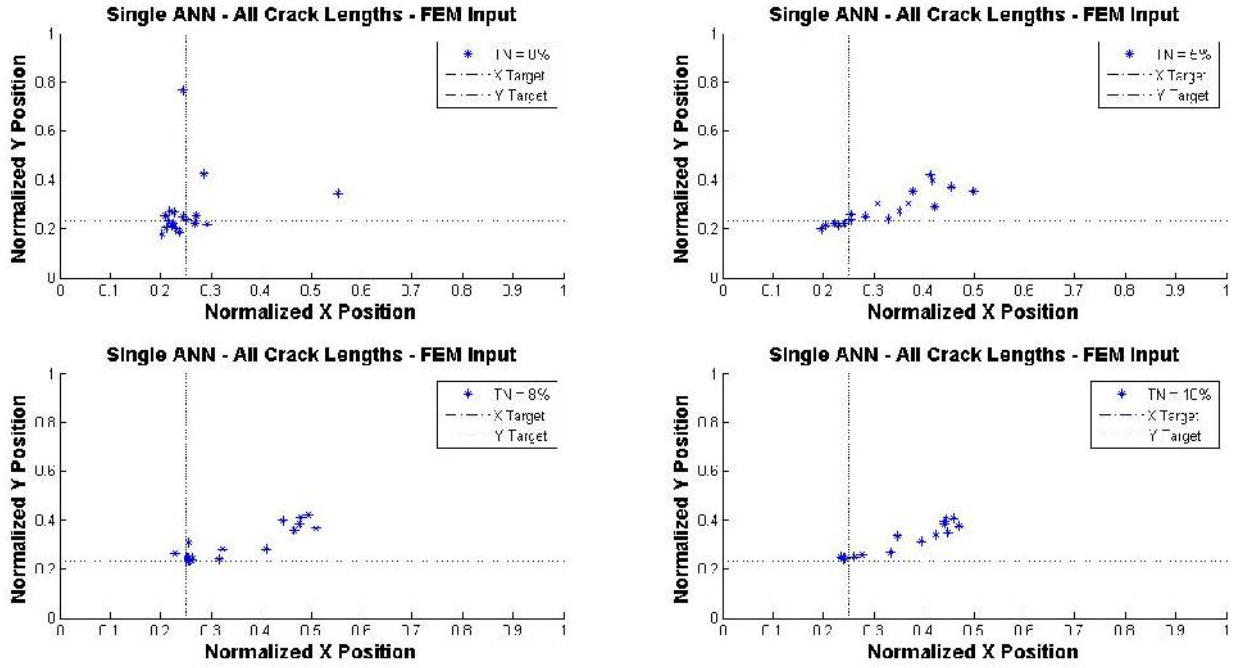



**Figure 5.26 Localization testing with crack positioning (crack number 1) on panel using single ANNs with TN=0% (upper left), TN=5% (upper right), TN=8% (lower left) and TN=10% (lower right)**

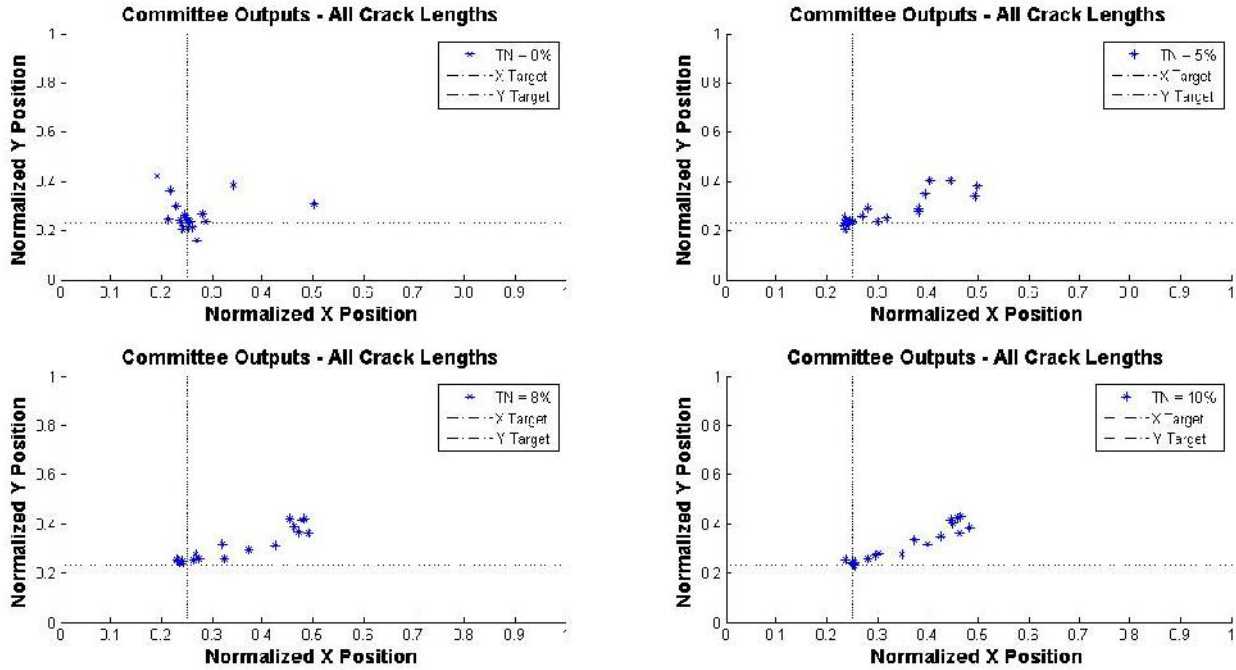



**Figure 5.27 Localization testing with crack positioning (crack number 1) on panel using COMT of ANNs with TN=0% (upper left), TN=5% (upper right), TN=8% (lower left) and TN=10% (lower right)**

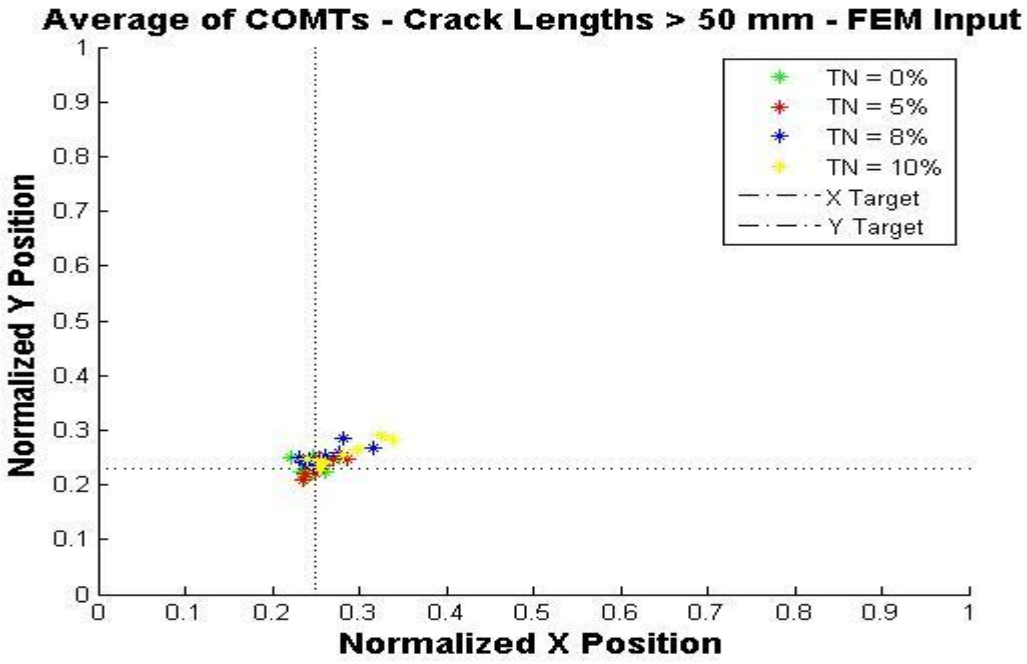



**Figure 5.28 A comprehensive representation of localization testing for COMT prediction which is averaged over 5 COMT outputs. In accordance with the anomaly detection algorithm strategy, only the cracks greater than 50mm are considered.**

To be more informative in the graphical representation when testing the localization algorithm, TN and DPE parameters are treated as variable simultaneously in a same plot; also both position coordinates of damage site are indicated as well. This is done for simulated cracks propagations number 1, 3, 5, 7 and 9 simultaneously (**see Figures 5.29, 5.30, 5.31 and 5.32**).

Again, the strain measures relevant to cracks smaller than 60mm are discarded from the input of algorithm. Increasing the DPE from 0% to 50% does not reduce the inference performance, but when DPE is set to 90%, the negative effect of insufficient training database size becomes evident. This worsening influence is more apparent for greater values of TN (additive training noise) i.e. TN=8%. Increasing the DPE even further up to 95%, only algorithms with TN=0% can locate the crack large measure to some extent.

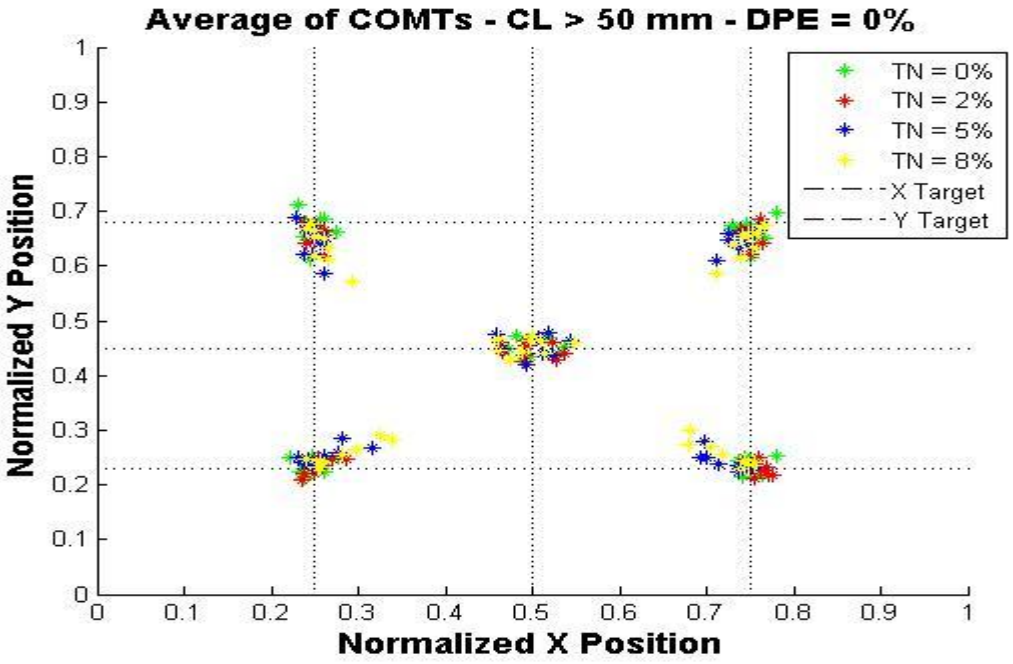



**Figure 5.29 Localization testing with crack positioning (crack number 1, 3, 5, 7 and 9) on panel using COMT ANNs averaged over 5 inputs with TN=0, 5, 8 and 10% . DPE = 0%.**

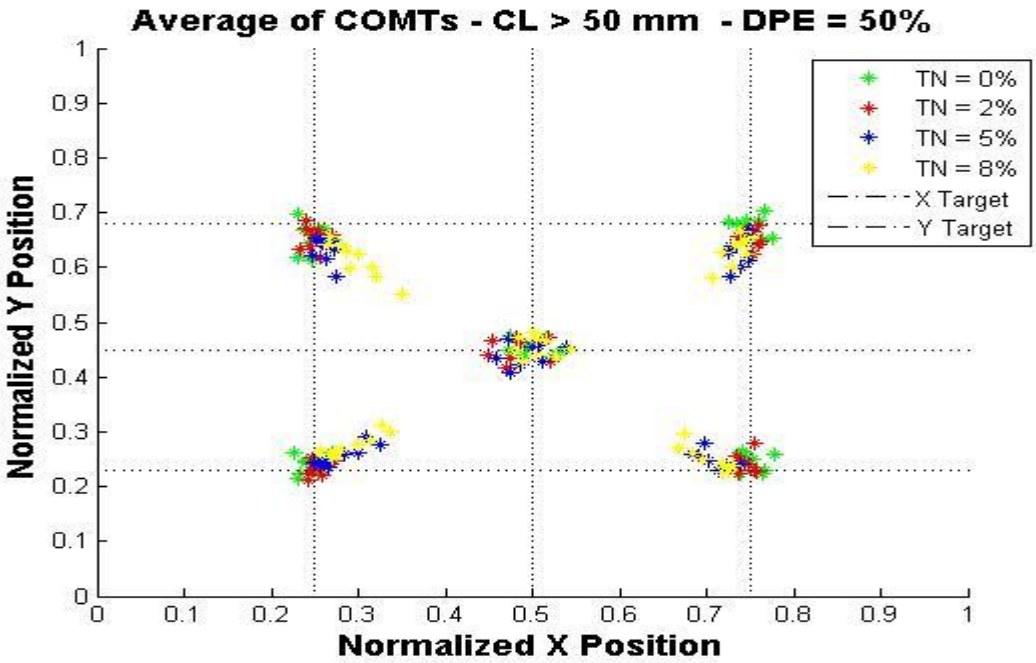



**Figure 5.30 Localization testing with crack positioning (crack number 1, 3, 5, 7 and 9) on panel using COMT ANNs averaged over 5 inputs with TN=0, 5, 8 and 10% . DPE = 50%.**

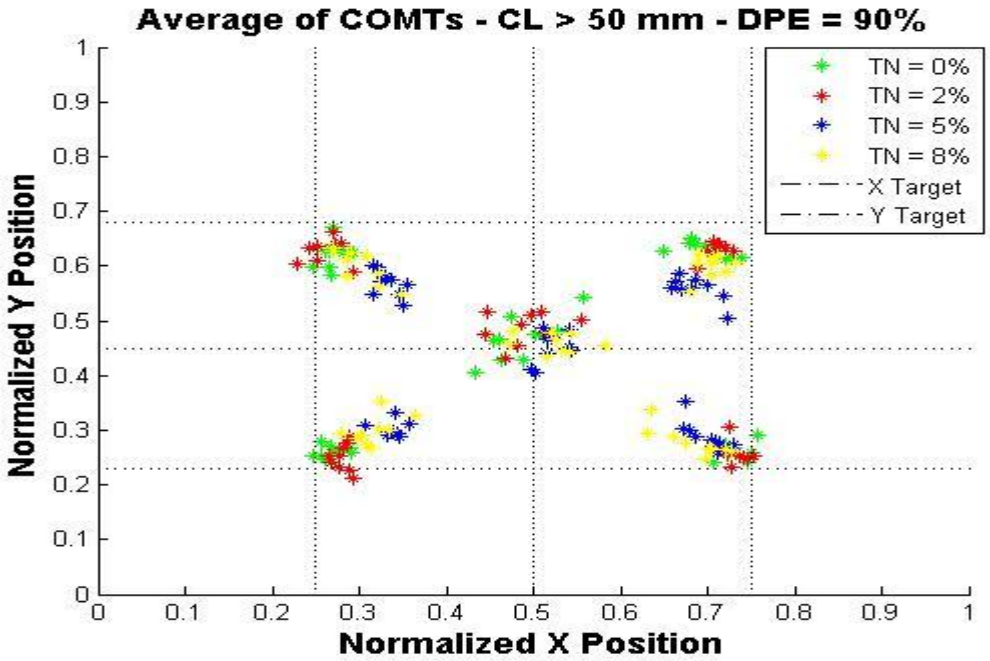



**Figure 5.31- Localization testing with crack positioning (crack number 1, 3, 5, 7 and 9) on panel using COMT ANNs averaged over 5 inputs with TN=0, 5, 8 and 10% . DPE = 90%.**

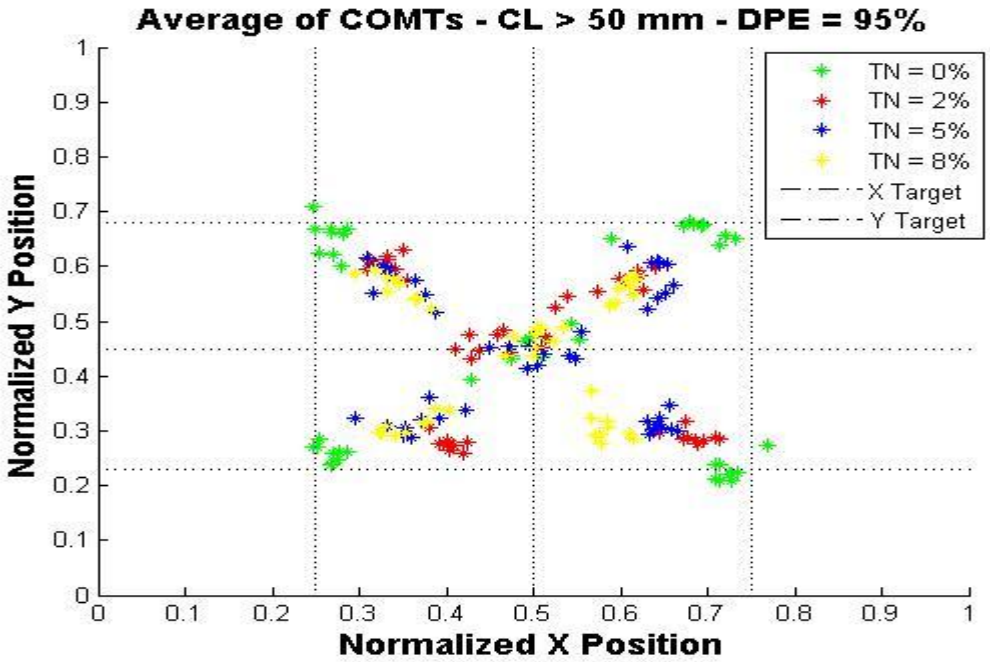



**Figure 5.32 Localization testing with crack positioning (crack number 1, 3, 5, 7 and 9) on panel using COMT ANNs averaged over 5 inputs with TN=0, 5, 8 and 10% . DPE = 95%.**

### 5.10.6 Localization algorithm testing with real experimental a database

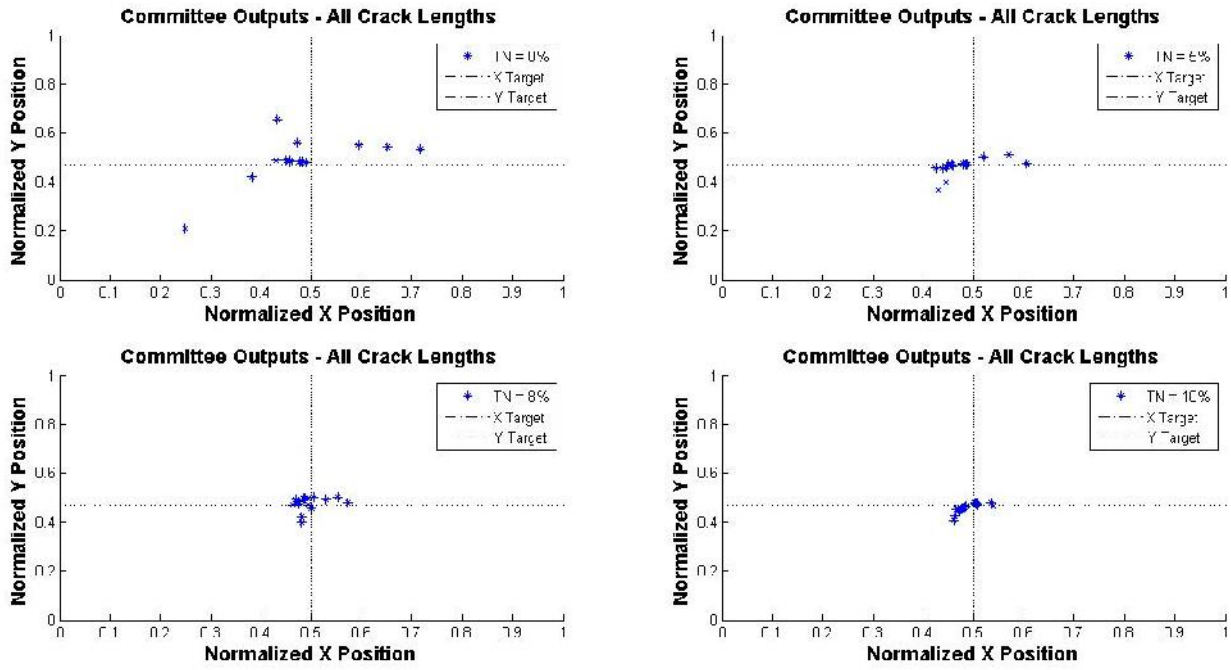



**Figure 5.33 Localization testing with real centre crack positioned on panel using COMT ANNs with TN=0% (upper left), TN=5% (upper right), TN=8% (lower left) and TN=10% (lower right)0%. DPE=0%.**

As stated in **section 5.8.3** the final agreement on the optimal additive noise for localization algorithm is the minimum value which is 0%. Relying on the result of section 5.8.3, it is expected that the generalization performance decreases as TN (training noise) increases. Nonetheless, the ANNs inference shows an improvement for position of a centre crack, provided by both simulated and real experiment (**see Figures 5.33 and 5.34**). This is a conflict that can lead to a misunderstanding of the true effect of the additive noise to the training database and the performance of localization algorithm. To overcome this confusion, it might be possible to associate this to the way that the weights of ANN synapses are adjusted during training phase, and therefore answering this exceptional generalization during testing.

In fact, this unusual observation in inference behavior can be seen in any localization ANN which is trained with high level of additive noise in its training data. To be specific, wherever the damage is on panel, the localizing ANNs trained with high noise level tend to infer the damage site close to the panel centre (**see Figures 5.26 and 5.27**). This is so true for small crack sizes. Finally, by again referring to **Figures 5.26 and 5.27,** it is agreed that ANNs trained with the null additive noise in their training database show the best generalization performance, when testing is carried out with crack propagation of other positions.

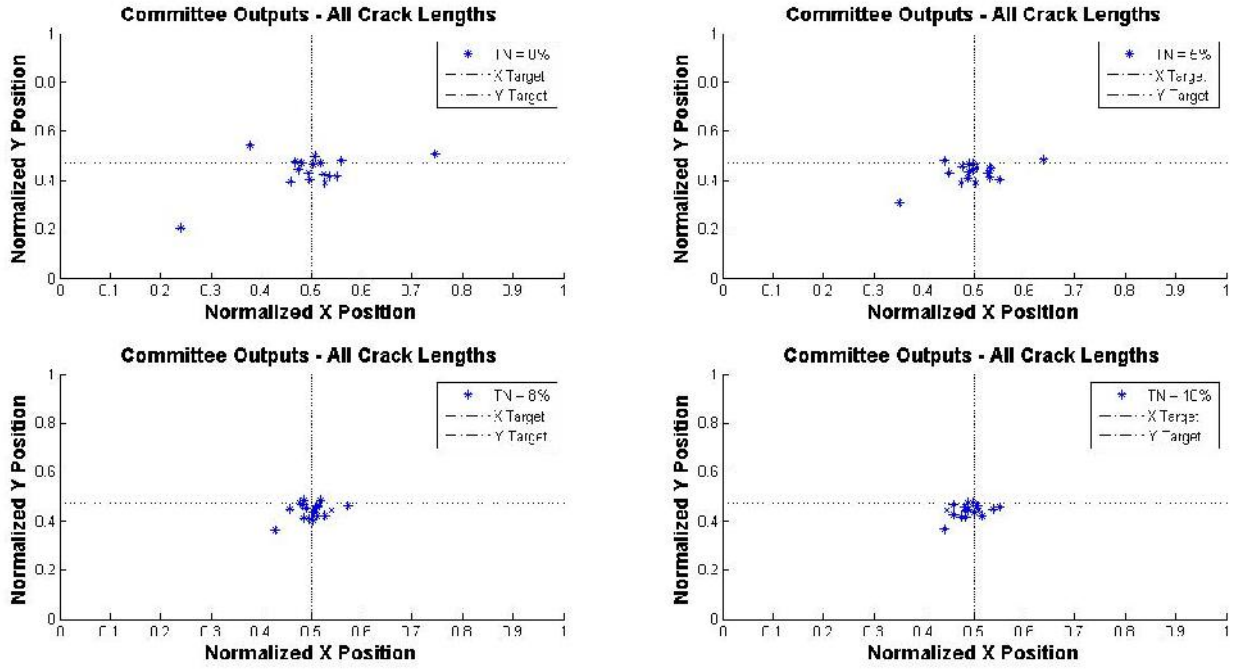



**Figure 5.34 Localization testing with simulated centre crack positioned on panel using COMT ANNs with TN=0% (upper left), TN=5% (upper right), TN=8% (lower left) and TN=10% (lower right)0%. DPE=0%.**

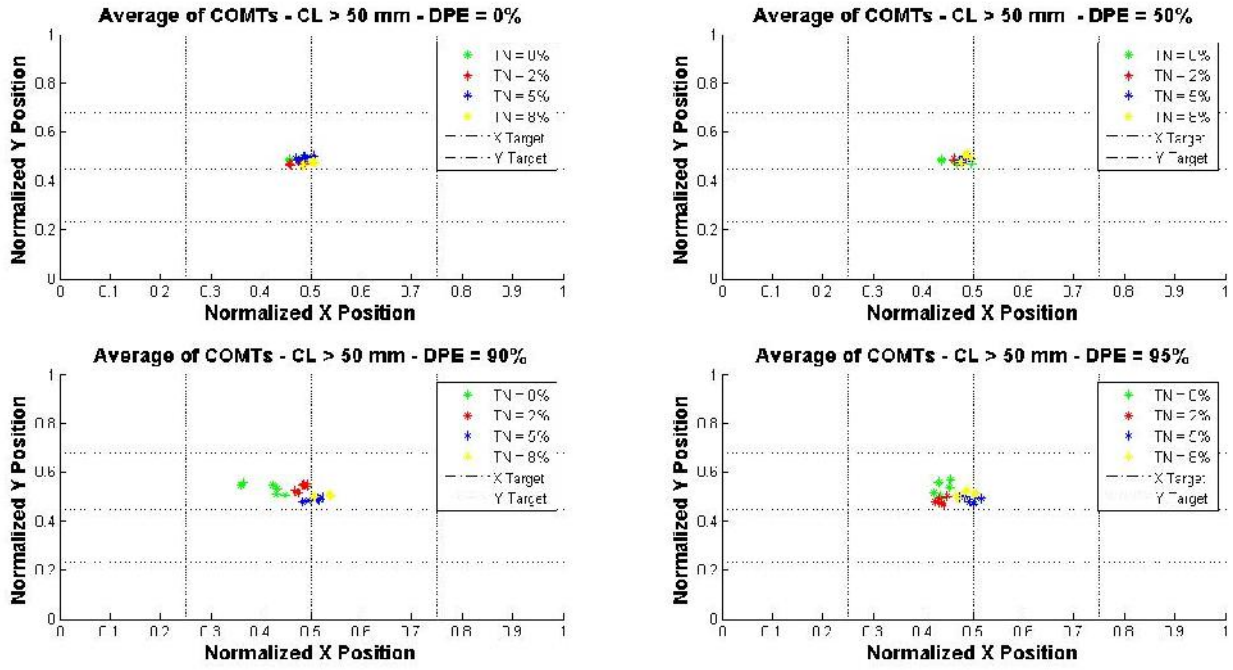



**Figure 5.35 Localization testing of ANNs trained with different additive noise in training data and DPE=0% (upper left), DPE=50% (upper right), DPE=90% (lower left) and DPE=95% (lower right) and tested with real centre crack propagation. The predictions are obtained by averaging 5 outputs of COMT.**

# Chapter 6 Conclusion

The objective of the work described in thesis is the development of the diagnostic system for helicopter fuselage crack propagation which works based on real time sensor acquisition. The scope is to provide a diagnostic unit that makes use of a permanently on-board installed sensor network to make an interpretation of the integrity state of the fuselage in real time. Therefore, this diagnostic system can generate an alarm in case of abnormal structural behavior. Upon the detection of an anomaly, damage parameters (damage intensity and centre position) will be estimated, thus allowing for an evaluation of the health state of the fuselage. Once the user is provided by damage parameters, it would be possible to decide in real-time whether maintenance is required.

In this work two approaches have been taken toward designing a diagnostic system. One approach uses an unsupervised anomaly detection algorithm which is called outlier analysis. The other approach implements a more sophisticated supervised algorithm which is called multi-layer perceptron (MLP) neural network. The viability of the fist approach depends on only availability of baseline databases acquired from the experimental tests. In the application of MLP neural network, the availability of damage experience for sensor network interpretation is the key aspect for the success of the structural health monitoring systems. Data-based and model-based approaches are usually adopted to retrieve such experience for sensor signal interpretation.

The main issues that were discussed within and the accomplishment that were achieved in this thesis are as the following:

- The outlier analysis (OA) theory proved to be successful in determining the panel health state only for the first level of diagnosis hierarchy which is simply damage detection. But, as an unsupervised algorithm, there is no training phase involved in building the algorithm. This is an advantage comparing to the algorithm provided by artificial neural network, where the availability of simulated model and the following training phase are necessary.
- Concerning outlier analysis a conceptually rudimentary approach based on the MSD is performed on a multivariate experimental data set. In performing OA analysis a salient assumption is made which is the presence of only one single outlier (a single damage site on the panel) at a time. This assumption is of a great help to avoid the effect of masking the damage sensitive features in strain readings.
- Concerning the application of MLP neural network, the availability of a validated numerical model offers the potential to simulate a relatively "unlimited" number of damage conditions, taking into account much more damage variables (e.g. damage intensity and position)

- The performance of MLP neural network is strictly reliant on the training database provided by FE model. Moreover there is the issue of FEM database deviation from the real experimental model data which introduces the concept of uncertainties. This issue is treated by considering the uncertainty as a parameter which is included in the training phase as Gaussian noise. One very important assumption is made by neglecting the effect of unknown localized loads that might be present and exert a local stress distribution that helped enormously to the simplification in the use of ANNs.
- Application of MLP networks offers the design of a complete diagnostic unit, therefore handing two extra level of inference that regards the state of the damage in a step further of its detection. These extra inferences of damage intensity and location are of extreme necessity of development of a maintenance strategy referred as SHM in introduction. Installing the diagnostic system provided by MLP on a machine a continuous screening of the machine's integrity state is possible, whereupon the case of damage presence the system addresses, firstly, the damage intensity which can be used as a decision making parameter to call off the machine from operation or not and, secondly, the defect position which notably reduces downtime of the machine by preventing the dismantling of the unnecessary components before reaching the faulty component.

# Bibliography


[1] H. Assler, J. Telgkamp, *Design of aircraft structures under special consideration of NDT*, Airbus Deutschland GmbH, Hamburg, Germany

[2] C. BOLLER, N. MEYENDORF (2008), *State-of-the-Art in Structural Health Monitoring for Aeronautics*, Fraunhofer Inst. for Nondestructive Testing (IZFP), Saarbrücken & Dresden, Germany

[3] Charles R. Farrar, Keith Worden, Nicholas A. J. Lieven, Gyuhae Park, *Structural health monitoring for aerospace applications*, Encyclopedia of Aerospace Engineering in 2010 by John Wiley & Sons, Ltd

[4] Claudio Sbarufatti (2012), *Fatigue crack monitoring of helicopter fuselage and life evaluation through sensor network*

[5] Farrar, C. R., Worden K. (**2007**), *An introduction to structural health monitoring,* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 365(1851), pp. 303-315.

[6] Alan D. Kersey, Michael A. Davis, Heather J. Patrick, Michel LeBlanc, K. P. Koo, C. G. Askins, M. A. Putnam, and E. Joseph Friebele, *Fiber Grating Sensors*, JOURNAL OF LIGHTWAVE TECHNOLOGY, VOL. 15, NO. 8, AUGUST 1997

[7] D. Chetwynd, J. A. Rongong, S. G. Pierce, K. Worden(2008), *Damage detection in an aluminum plate using outlier analysis.*

[8] F. Mustapha, G. Manson, K. Worden, S.G. Pierce (2006), *Damage location in an isotropic plate using a vector of novelty indices,* journal of mechanical systems and signal processing

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

[10] Charles R. Farrar, Keith Worden (2007), *structural health monitoring: a machine learning perspective*

[11] K. Worden, G. Manson (1999), *Damage detection using outlier analysis,* Journal of sound and vibration (2000) 229(3), 647-667

[12] K. Worden (1996), *Structural novelty detection using a novelty measure*, Journal of Sound and Vibration (1997) 201(1), 85-101

[13] C.Bishop, *Neural networks for Pattern Recognition*, Oxford University Press, 1995.

[14] C.Bishop, *pattern regognition and Machine learning*, Springer, 2009.

[15] Tshilidzi Marwala (2000), *Damage identification using Committee of neural networks,* Journal of Engineering Mechanics.

[16]Worden, K. and Manson, G. (2007), ***The application of machine learning to structural health monitoring,*** *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **365**(1851), pp. 515-537.

[17] Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth, *Neural Network Toolbox™ User's Guide* MATLAB R2012b.