National Technical University of Athens

School of Naval Architecture and Marine Engineering

Division of Marine Structures

Shipbuilding Technology Laboratory

Diploma Thesis

Damage Identification in Thin-Walled Girders through a Finite Element-Based Digital Twin

Nikolaos E. Silionis

Thesis Supervisor: Konstantinos N. Anyfantis, Assistant Professor

ATHENS, NOVEMBER 2020

One generation passeth away,
and another generation cometh:
but the earth abideth forever.
Ecclesiastes 1:4

Acknowledgments

This thesis marks the final chapter of an academic journey, during which much has changed for me personally and for the world as a whole. Now standing at the threshold of an uncertain future, I find myself driven and content.

First of all, I would like to express my gratitude towards my thesis supervisor, Assistant Professor Konstantinos Anyfantis without whose guidance, support and direction the completion of this work would not have been possible. His knowledge on the subject and his desire to share it, his patience and willingness to provide assistance at all times and his deep intuitions proved invaluable during the course of this thesis. But most importantly, his personal integrity and human qualities proved to be the cornerstone of a successful and prosperous collaboration.

To the members of my supervisory committee, Professor Nicholas Tsouvalis and Assistant Professor George Papalambrou, I would also like to extend my deepest thanks for evaluating the quality of my thesis.

I would also like to thank the technicians of the Shipbuilding Technology Laboratory, Mr. Athanasios Mrkoulis and Mr. Haris Xanthis, for their assistance in the experimental activities and the valuable knowledge they shared with me during this time.

To my family, for their unconditional support during all my years as a student, for their patience and belief in me, I extend my deepest thanks.

Finally, to all those close to me who supported me during this undertaking, whose names I find unnecessary to state here as I hope that when they see this they may know themselves, I extend my deepest and most sincere gratitude.

Abstract

Humanity stands on the verge of an industrial revolution which will drastically alter the way people live, work, and interact with one another. At the core of this stands the digital twin, a virtual representation of physical assets aiming to enable the condition-based monitoring of all their operational aspects. This thesis draws inspiration from this concept and aims to provide an initial approach towards the implementation of a digital twin for ship hull structures, albeit in a simplified form.

The main goal of this work is to develop damage-identification methods for thin-walled girders, based on machine learning and optimization concepts. The choice of a thin-walled girder subjected to three-point bending to model the hull structure under still water loads was based on the principles of hull girder strength. A FE-based Digital Twin of a thin-walled girder subjected to three-point bending was developed and used to determine the features of the Structural Health Monitoring system used to facilitate damage detection. This system's capabilities were tested by the inclusion of a feature simulating the effects of damage on the strain field, known in this work as a Strain Field Disturber.

After the capabilities of the SHM system's strain monitoring scheme to detect damage were established, two methods aimed at solving the inverse problem of predicting the damaged state of the girder, using only strain data as input were developed to complete the SHM framework. The first utilized techniques found in the field of optimization, specifically Genetic Algorithms, and treated the problem as an optimization problem where damage detection corresponds to the minimization of an appropriate error function. The second used Artificial Neural Networks, trained using data obtained from the digital twin to enable the prediction of damaged states based on strain inputs. The capabilities of both methods were tested within the virtual environment, using the digital twin as the means to provide the requisite data.

Finally, the damage identification framework was tested against actual experimental data as well. A series of three-point bending experiments were executed and the strain data obtained from them were used to test the efficacy of the developed methods in real-world conditions.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Types of Damage in Ship Structures

An overarching goal for all structural designers is the avoidance of structural failure. Obviously, this applies to naval architects as well, who must be able to identify the various modes of failure which are common to ship structures and also be familiar with the methods used to predict their occurrence.

Normally, a ship hull in the intact condition will clearly sustain applied loads smaller than the design loads in loading, still-water and sea-going conditions without suffering catastrophic collapse (Paik & Frieze, 2001). However, structural degradation suffered by ships during their lifecycle, may lead to collapse under applied loads even smaller than the specified design loads. It is thus made clear, that being able to first identify, and then monitor such types of damage as may cause structural degradation and consequently lead to collapse, is of paramount importance to the safety of the ship's crew and cargo during the vessel's lifetime. Such types of damage arise mainly due to the structural configuration of ship structures, the nature of the loads applied to them and the nature of their operating environment as discussed below in greater detail.

A distinctive characteristic of the structural configuration of ships is that they are mainly comprised of numerous stiffened plate panels, as illustrated in Figure 1.1, some plane and some curved, which make up the side and bottom shell, the decks, and the bulkheads (Mansour & Liu, 2008).

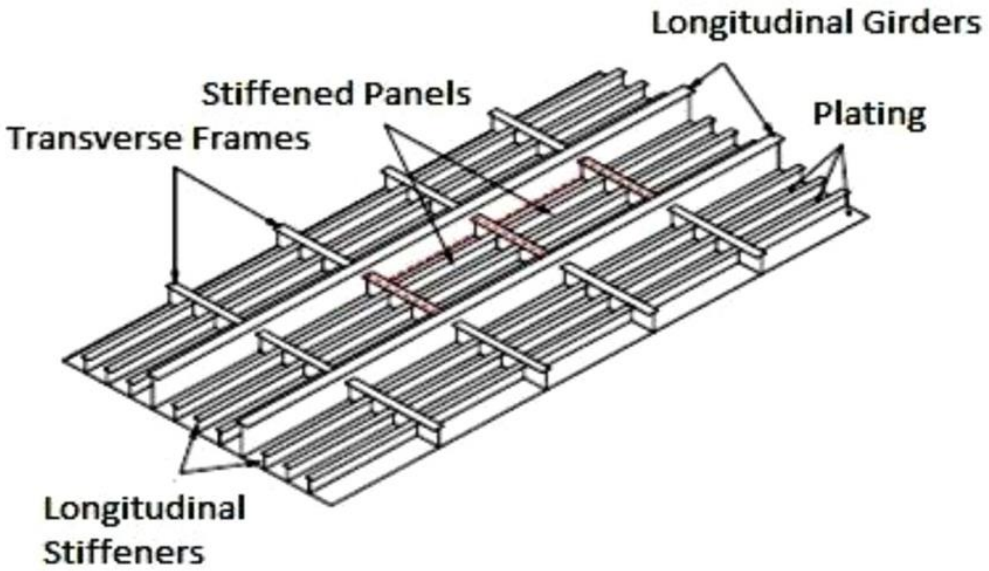


Figure 1.1: Typical Stiffened Panel Structure in Ships

As the hull girder is subjected to longitudinal bending, the resultant primary stresses cause these plates to be subjected to in-plane compression which may lead to buckling

when extreme bending moments act. In this case buckling is a failure mechanism that a structural member such as a plate, a stiffened plate, a stiffener, a column, etc., which are under thrust load deflect in an out-of-plane direction when the load reaches a certain critical value (Fujikubo & Yao, 2016). When a certain structural member undergoes buckling, its load-carrying capacity decreases leading to a redistribution of internal forces among non-buckled members, which in turn increases the internal forces in them and may lead to them undergoing buckling as well. If the applied load is increased, progressive buckling may occur leading to the collapse of the entire structure. One such case, is presented in Figure 1.2 below.

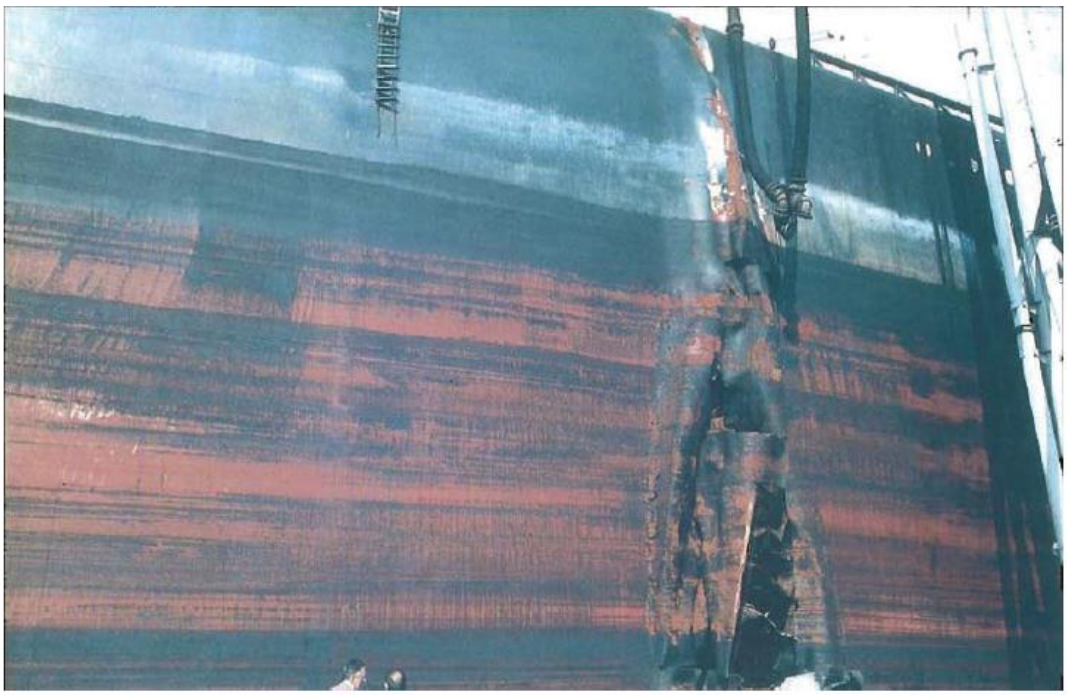


Figure 1.2: Side Shell Collapse due to Buckling.

As was mentioned earlier, a key contributing factor to the structural degradation of ship hulls is the nature of their operating environment as well as the cargo carried in their holds, insofar as they are responsible for the corrosion of the ship's plating and other structural members (see Figure 1.3). It is generally accepted that the marine environment, that combines the effects of saline seawater, salt laden air, rain, dew, condensation, localized high temperature and the corrosive effects of combustion gases is the most corrosive of naturally occurring environments (Saathesh Babu, Mathiazhagan, & Nandakumar, 2014). In marine structures, both general corrosion which leads to a uniform reduction in plate thickness, and other forms of localized corrosion (e.g., pitting or stress corrosion cracking) are found. In all cases, the advent of corrosion leads to a reduction in the load-carrying capacity of structural members. This may ultimately lead to failure of the entire ship's structure even at loads below the design load.

Figure 1.3: Instance of Corrosion in Water Ballast Tank Air Vent

It should be noted, however, that damages found in ships that lead to structural failure vary with respect to severity. Cracks and deformations in minor structural members are inevitable, however they do not jeopardize the structure's ability to perform its function, unless they become accumulated. On the other hand, buckling and corrosion, which were described in detail above, are most likely to lead to progressive collapse and ultimate failure of the entire structure. And although this rarely occurs, still appropriate measures must be in place to ensure it is averted.

Obviously, during the design stage the factors mentioned above are considered during the process of calculating the maximum design load applied to the structure. The framework for the calculation of the maximum design load and subsequent ultimate strength of the hull structure is provided by rules and regulations set forth by classification societies. In order to provide a unified framework of rules and regulations to be followed by these societies in the classification of ships, the International Association of Classification Societies (IACS) was formed by the ten leading classification societies. These rules and regulations are established to ensure that sea-going vessels meet certain standards of acceptable risk associated with the conduct of maritime transport, which is dictated by the needs of the international community and facilitated through its representation in the International Maritime Organization (IMO).

Compliance to these rules is the principle requirement for a vessel's classification which, insofar as this thesis is concerned, means that the standards for its hull's structural strength are met. However, class designation awarded to a ship being built according to the relevant rules and regulations does not guarantee its structural integrity

throughout its lifecycle. As mentioned earlier, the effect of structural degradation may lead to significant failure even at loads lower than the design load. To safeguard against that and ascertain that the ship remains in compliance with the relevant rules, classification societies mandate that it must be subjected to periodical and non-periodical surveys.

That is achieved with a through-life survey regime ships must follow if they are to be retained in class. Those surveys which are of interest to this work include the class renewal (also called “special survey”), intermediate survey and annual survey. Special surveys take place every five years when the ship is in dry-dock and include an extensive examination of the hull, including ultrasonic thickness measurements if deemed necessary. According to IACS, the survey is intended to assess whether the structural integrity remains in conformance with the standards contained in the relevant rules and to identify areas that exhibit substantial corrosion, significant deformation, fractures, damages or other structural deterioration. The intermediate survey is held approximately half-way between special surveys and is intended to verify if the ship is in a general condition that satisfies the rules. Again, an examination of the hull is performed, and ultrasonic thickness measurements are carried out if deemed necessary by the attending class surveyor. Finally, in annual surveys the ship is generally examined and only an external general examination of the hull is performed.

Specific guidelines are issued by IACS and individually by classification societies that provide guidance to surveyors on the areas where the aforementioned types of damage are more likely to occur and that should be inspected. For example, the internal surfaces of tanks and holds are to be examined for the presence of general corrosion, found in the form a thin layer of rust. Also, areas of stress concentration, such as those in and beside welds are to be examined for both the appearance of grooving corrosion as well as the initiation of fractures. Special attention should be paid to welds at toes of brackets, at cut-outs, and at intersections of welds. Regarding the presence of deformations, plates as well as stiffeners are to be inspected to verify that buckling has not occurred, especially in areas where corrosion is also in evidence. Finally, any evidence of damage on shell plating due to contact with other objects must be carefully investigated from the internal side as well, as often internal members are more heavily damaged than external deformations suggest.

1.2 Digitalization in the Maritime Sector

Although the aforementioned scheduled maintenance and inspection regime has been proven to be reliable and successful, it does not represent an optimal solution to the problem of effective and timely monitoring of hull structural integrity. Because of its rigid nature, it offers no information regarding structural integrity in real time which in the case of a wrong decision might prove catastrophic. For example, if a fatigue fracture due to repeated stresses exists in a frame, it may propagate into shell plating and affect the watertight integrity of the hull. In extreme weather conditions the shell fracture

could extend further resulting in the loss of part of the shell plating and consequent cargo hold flooding. Also, even though current inspection guidelines identify key locations of the hull structure that warrant special attention, some uncertainty is still present after the inspection, especially when considering the number of inaccessible spaces to the surveyor which can be assessed only during the special survey. Also, the fact that the physical presence of the surveyor is at all times required, highlights the limited flexibility current practices afford. Therefore, the need for a condition-based monitoring and inspection scheme in the maritime sector, which would assist in the decision-making of all stakeholders involved, becomes apparent. And there is no better option, to embrace the trend of digitalization, sweeping across all of industry.

Already, steps are being taken towards this direction. Many of the world's foremost shipbuilding powers have been laying out the groundwork for the transformation towards "smart" shipyards. This refers to the integration of traditional shipbuilding practices with technologies under the umbrella of what has become known as Industry 4.0, such the Internet of Things (IoT), Cyber-Physical Systems (CPS), Big Data, Cloud Computing and Artificial Intelligence. The centerpiece of these technologies is the concept of the Digital Twin.

A digital twin in its simplest form is a virtual replica of a physical asset, whether that may be a product, process, or an entire system. Within the context of the IoT however, it may act as a bridge connecting the virtual and physical worlds by using data collected from sensors on the physical asset, in real time, to capture its operational condition. This virtual model can then be used to analyze and optimize performance, and when combined with machine learning and artificial intelligence tools, to be able to predict future problems, paving the way for the optimization of maintenance and the extension of the asset's operational lifetime.

The adaptation of such technological innovation in the maritime sector, has not only captured the attention of shipbuilders. Operators of both ships and other offshore assets, as well as major classification societies have seen the potential the adaptation of digital twin technologies may have in optimally managing and extending the prospective lifetime of their assets. An overview of the digital twin as it may be implemented in shipping, along with the underlying technologies it encompasses is shown in Figure 1.4 of the following page.

In the context of this work, the asset in question is the ship's steel structure itself, so it naturally focuses on the concept of the hull structural digital twin. Current practice when it comes to the monitoring of structural integrity was discussed in detail in the previous section and it was made clear that current methods, although effective, pose certain limitations. Specifically, the very nature of a scheduled inspection regime offers little to no assurance regarding the timely detection of damage and thus presents owners and operators with a foregone conclusion at the time of the survey.

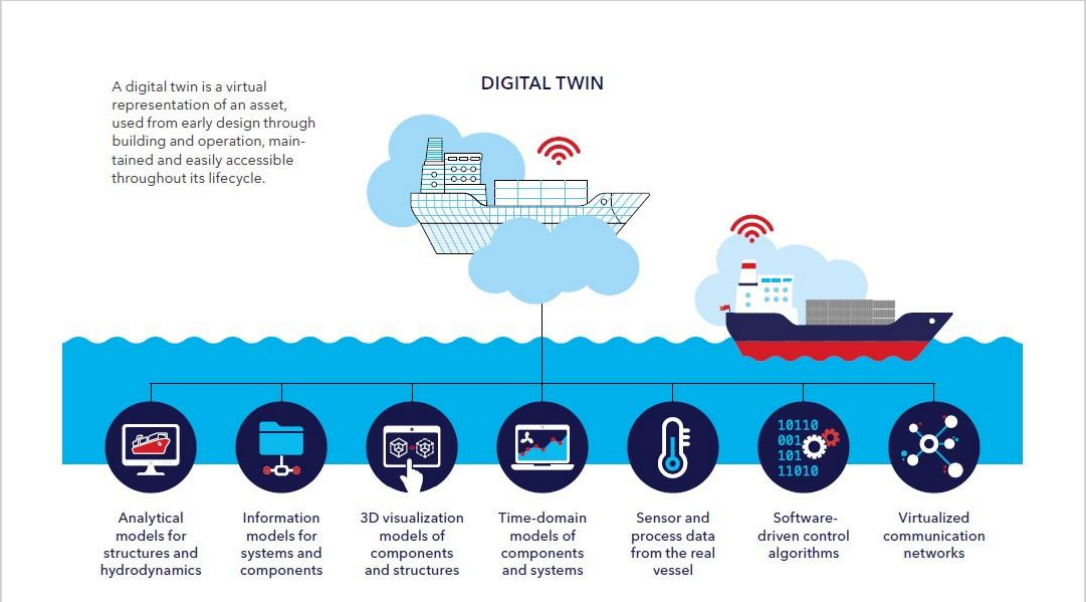


Figure 1.4: The Digital Twin in Shipping (Smogeli, 2016)

On the contrary, the ship structural digital twin offers operators and technical managers unprecedented flexibility in scheduling maintenance and inspection. It, being a physics-based digital model that represents its actual counterpart in absolute detail and accuracy, while also having the ability to be updated, in real time, with data corresponding to loading conditions and actual measurements from on-site inspections, enables structural assessment to be made in the "as is" condition from any place and at any time. Thus, it is made possible for the managers to realize through it several important operational objectives. These include:

- The identification of critical areas in need of immediate inspection, maintenance, and repair.
- An increased ability to perform inspections in hard-to-reach spaces without putting any crew members in danger.
- The ability to prepare for upcoming unforeseen conditions, such as extreme weather events or emergency repairs, by using the simulated environment to assess alternative courses of action.
- Finally, to extend the life of various assets safely by producing more realistic results about the estimated remaining fatigue life than the most conservative ones, produced by more conventional simulation software currently in use.

Already, major classification societies such as DNV GL (Smogeli, 2016) or ABS (ABS Global Engineering and Technology Overview, 2018) are allocating significant research and funding towards that direction and are also looking into potential joint venture applications with industrial partners. The long-term goal of their strategy being, to transform the current inflexible plan-based inspection and maintenance regime to a more flexible condition-based one, with increased capabilities for remote inspections, thus eliminating what difficulties may arise from surveyor availability and scheduling conflicts.

1.3 Structural Health Monitoring

Aside from the benefits presented and the opportunities afforded by the implementation of a structural digital twin in modern ocean-going vessels and offshore structures, it should be noted that the relevant research and applications are still at an early stage.

In general, research on the subject is more focused on the implementation of the principles of Structural Health Monitoring (SHM) in order to gather insights on how to approach the problem of damage detection in ship structures, thus making possible the creation of a data enabled structural digital twin. Structural Health Monitoring refers to systems which enable the automatic and on-line observation of structural integrity of any given component during service (Preisler, Schröder, & Schagerl, 2018). Farrar, Doebling, & Nix (2001) describe SHM by way of a ‘‘statistical pattern recognition paradigm,’’ the final step of which may be broken down into four levels (Rytter, 1993), progressively increasing in the depth of information required:

- Step 1: Operational evaluation.
- Step 2: Data acquisition, normalization, and cleansing.
- Step 3: Feature selection and information condensation.
- Step 4: Statistical model development for feature discrimination
  - Level 1: Detection
  - Level 2: Localization
  - Level 3: Characterization
  - Level 4: Prediction.

This is indeed the paradigm adopted by most researchers working on SHM applications. In the context of this thesis, attention was focused mainly on Step 4, and its application on an idealization of ship hull structures, as will be explained in due course.

However, ship structures present unique difficulties -not present in most land-based applications- mostly due to their increased complexity. More specifically, the multitude of enclosed inaccessible structural components as well the sheer number of components in general, does not allow for an “obvious” decision to be made regarding the location of the sensors which constitute the backbone of any SHM system. Therefore, the development of a data assimilation method arises as an essential element for the realization of a digital twin for ship structures. In the context of this work, this refers to a method which enables the prediction of stresses and strains in a large number of unmeasured or unmeasurable locations using a limited number of monitored responses (Fujikubo, 2019).

Because of these considerations, earlier research on the subject was focused on specific structural components, rather than focusing on the hull structure as a whole. Such works, although containing important and fruitful insights were not taken into consideration for this thesis, as it sought avenues to take on the task of monitoring the entire hull structure. Recently however, there has been research focused primarily on this type of problem, coming from both academia as well as the industrial sector.

Namely, Stull, Earls, & Koutsourelakis (2011) proposed the development and application of SHM techniques to full-scale ship structural systems using a model-based paradigm which employs two different approaches to solve the resultant SHM problem. The goals of a model-based approach are the same as those described earlier (Step 4 etc.), however it does not seek to attain them like traditional SHM approaches do, that is by comparing actual observed data to established statistical models and treating damage detection as merely the detection of statistical anomalies. It is concerned with the solution of the inverse problem, that is how to infer, given an observed response from an existent potentially damaged structural system, a model of a structural system which accurately captures this response. The two approaches to the solution of the problem mentioned earlier, are divided to one which follows a classical route and one which utilizes a probabilistic formulation to account for the noise and uncertainties of measurements. While the second approach is novel to the subject and very promising, the first one proved to be a significant influence on the methods developed later in this thesis and will be further expanded upon then.

Another approach taken towards the development of a structural digital twin for the complete ship structural system, is detailed by Kefal, Olgun, & Oterkus (2015). In their work, they aim to develop a method that utilizes strain measurements from on-board sensors located on various key locations on the ship's hull to reconstruct, in real time, the complete structural displacement, stress and strain fields, thus achieving the basic goals of structural health monitoring, as they were described earlier. To achieve that, they first used hydrodynamic analysis to simulate ship loading and rigid body motion, then used Finite Element Analysis to obtain the structural response and simulated sensor strain data. Finally, they used an algorithm called inverse Finite Element Method (iFEM) to reconstruct the structural response by using the simulated strain data obtained from virtual sensors at various locations in the structure as input. Their methods were tested on the model of a barge and optimum locations for sensor placement to achieve the SHM were identified.

Of particular interest also were the methods developed by Akselos, a private company working mainly with the oil and gas industry, that has currently successfully deployed a structural digital twin for an FPSO, owned by Shell and currently operating off the coast of Nigeria. Instead of using the conventional approach for the physics-based simulation of structural integrity i.e., Finite Element Analysis, which -as has been mentioned earlier in this work as well- is computationally intensive, they have developed their platform based on Reduced Order Modeling (ROM). Reduced order modeling refers to a family of numerical methods aiming to reduce the computational cost associated with evaluating high-fidelity models such as those produced by FEA.

Specifically, they have developed a reduced order modeling approach called RB-FEA (Reduced Basis-FEA), which leads to a rapid decrease in time required to solve models, without sacrificing in accuracy or detail. This is key to the implementation of the digital twin technology since it enables true condition-based monitoring of large and critical assets. Also, it allows for all relevant structural details to be included in the model,

along with sensor and inspection-based condition data such as cracks, corrosion, and damage due to impact or collision.

1.4 Thesis Objectives

In the previous section, it was made clear that there is great potential in the implementation of the digital twin concept in ships and floating structures, to implement condition-based monitoring schemes and most importantly to make possible the real-time detection of structural damage. This thesis, drawing inspiration from the works presented earlier, aims to take a first step towards the implementation of these concepts, albeit following a scaled-down approach.

At the core of this work, is the development of a Finite Element-based Digital Twin for a thin-walled square girder subjected to three-point bending, which represents an idealization of the hull girder under still water loading. This idealized geometry may be regarded as an abstract slender vessel, where the two opposite flanges represent the deck/bottom and the two opposite webs represent the side shell of the hull. The introduced damage is a drilled hole and may be considered to act as stress disturber. The main objective is to develop methods and assess their effectiveness in detecting a single damage which is present in thin-walled structures. The problem at hand considers the solution of the inverse problem: finding the location of damage given a known structural response. To that end, two schemes were employed: an optimization-based and a machine learning-based approach. The optimizer selected for the first approach were Genetic Algorithms (GA) while in the second Artificial Neural Networks (ANN) were used. Changes in strain readings were considered as the damage feature parameter and these were monitored at specified locations.

This thesis was organized as follows. In Chapter 2, the necessary theoretical background for the methods employed herein is provided. In Chapter 3, the development of the FE-based digital twin model is presented, along with the framework of the Structural Health Monitoring (SHM) system used to facilitate damage identification. The basic particulars of the two damage detection approaches mentioned before are also laid out in this chapter. In Chapter 4, the ability of the developed framework to detect damage is tested using data obtained from the digital twin and corresponding to indicative damage scenarios. Finally, in Chapter 5 the same framework is tested, this time using data obtained from a series of laboratory tests carried out on actual specimens in order to test its effectiveness in real-world conditions.

2 Theoretical Background

2.1 Introduction

The purpose of this chapter is to provide in a brief yet detailed fashion, the theoretical background necessary for the understanding of the various methods and techniques employed during the course of this thesis. In Chapter 2.2, the fundamental concepts of Finite Element Analysis (FEA), used to build the digital twin, are presented. In Chapter 2.3, an overview of the key concepts behind Genetic Algorithms (GA) is given while in Chapter 2.4, core ideas in the field of Artificial Neural Networks (ANN) are discussed, these two methods being the basis of the damage detection schemes developed and discussed in Chapters 3.6 and 3.7 of this work.

2.2 Fundamental Concepts of Finite Element Analysis

2.2.1 The Finite Element Method

In the course of the past half century, the Finite Element Method (FEM) has become a powerful and ever more increasingly used tool in various engineering applications, ranging from problems of stress analysis in various types of structures to field problems such as heat transfer or magnetic flux as well as other continuum problems. As far as this work is concerned, the FE method is used in the context of structural analysis. In all applications however, the analyst seeks to calculate a field quantity e.g., in stress analysis that is the displacement or stress field. The FE method aims to produce a numerical solution to a specific problem. A FE analysis does not produce a formula as a solution, nor does it solve a class of problems (Cook, 1995). In addition to that, the nature of the method dictates that the solution is approximate, except in cases where the problem is simple enough that an exact formula already exists.

Simply put, the FE method involves discretizing a complex structure or region into simpler geometric shapes called finite elements. The material properties and governing equations are applied over these elements, expressed as functions of unknown values taken at element corners, known as nodes. After taking into account the loading and supports, this process leads to a set of simultaneous algebraic equations and the continuous problem now becomes discrete. In the case of stress analysis, these equations are the equilibrium equations at the nodes. Solving these equations, which routinely number in the thousands or even hundreds of thousands, gives an approximation of the behavior of the continuum.

The process is described by Cook (1995), in more rigorous terms. Namely, the FE method is regarded as piecewise polynomial interpolation of a field quantity, such as

displacement, which is interpolated over an element from its values at the element's nodes. As the elements are then connected together, this field quantity becomes interpolated over the entire region in piecewise fashion, by as many polynomial expressions as there are elements and the problem instead of continuous now becomes discrete. The values of the field quantity at the nodes are such that they minimize some function (e.g. strain energy). The set of algebraic equations for nodal values of the field quantity which is the result of this minimization process, can be written in matrix form as $\boldsymbol{KD} = \boldsymbol{R}$, where $\boldsymbol{D}$ is a vector of the field quantity at the nodes, $\boldsymbol{R}$ is a vector of known loads and $\boldsymbol{K}$ is matrix of known constants. In the context of stress analysis $\boldsymbol{K}$ is known as the stiffness matrix.

2.2.2 Elements and Nodes

As mentioned earlier, this work is concerned with the use of the Finite Element Method in structural problems. This class of problems was the one for which this method was originally developed, arising as an extension of matrix frame analysis to continuum structures such as plates and shells (Hughes & Paik, 2010). Therefore, the basic concepts in both methods are practically the same; that is, as mentioned earlier, the representation of the structure as an assemblage of individual structural elements interconnected at a discrete number of nodes. In frame structures, this type of discretization is evident as they indeed consist of discrete beam members connected at various joints. However, this is not the cases in continuum structures such as panels or plates where a natural subdivision does not exist. Therefore, their artificial discretization in finite elements, which as mentioned before are simple geometric shapes, is made necessary.

These elements essentially represent fragments of the structure and may be one-dimensional (e.g., beams or rods), two-dimensional (e.g., triangles or rectangles) or even three-dimensional (e.g., cubes). In any case, nodes exist on the boundaries of those elements and serve to tie them together. In Figure 2.1, a bracket modelled using two-dimensional rectangular plane elements each having nine nodes is shown. The nodes in this case are indicated by dots and are located not only in element corners, but also at their center as well as the middle of each element edge.

When seen in a more superficial way, it would appear that a FE model of a structure or structural component could be produced by simply dissecting it into finite elements and then putting them together by connecting them at their nodes. However, actual structures such as the component shown in Figure 2.1 are continuous and therefore complete displacement continuity must be satisfied across all element boundaries. Clearly this would not be the case if elements were to be connected only at their nodes, as it would lead to an assemblage unrepresentative of the actual structure because of strain concentrations at nodes, possible sliding and overlapping of elements and even gaps appearing between different elements.

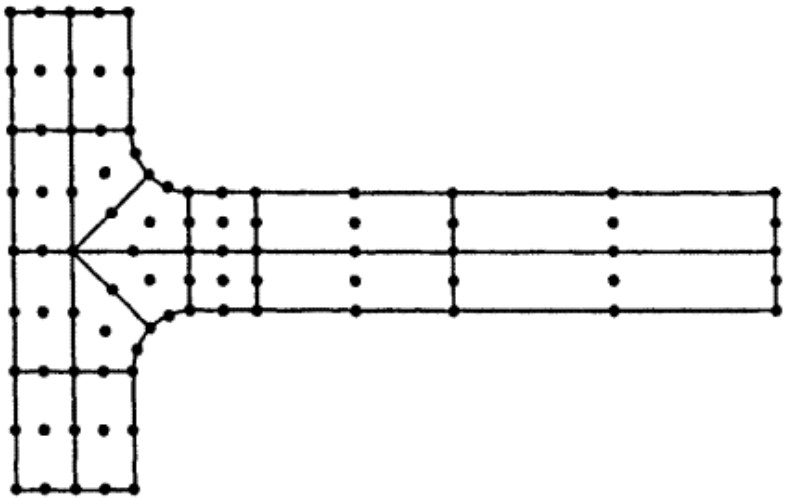


Figure 2.1: Bracket Modelled by 9-node Rectangular Elements.

In order to avoid these defects and minimize error, thus permitting the convergence of the method towards exact results, one way is to use more numerous and smaller elements, as it would increase the number of nodes and consequently the number of points on which displacement continuity is achieved. As has been mentioned before however, a discrete model cannot, by its very nature, produce an exact representation of a continuum regardless of the number of discrete variables it employs. Some error will always be present, and the resultant solution will always be an approximate one. Also, it is important to note that increasing the number of discrete variables, increases the number of algebraic equations needed for the solution of the problem and ultimately the computational cost of the model.

It is thus made clear, that besides using smaller and more numerous elements to minimize error, a more "elegant" way of achieving that is required. One such way is to carefully define and choose element properties; namely to precisely define each element's mode of deformation. This is the most fundamental feature of the Finite Element Method, as it allows both equilibrium and interelement continuity to be satisfied while only being explicitly enforced at the nodes. The way this is achieved, is by using a function which describes the distribution of a certain field quantity (e.g. displacement) across the element, using only nodal values. These functions are known as shape functions and are mostly polynomials whose order ultimately defines both the element's mode of deformation, as well as its behavior and applicability in general.

To further illustrate the role of shape functions in the FE method, as well as how they ensure interelement compatibility as mentioned before, let's take for example an element such as the one shown in Figure 2.2. This finite element is a flat triangle-shaped element of constant thickness and isotropic material properties and is referred in the literature as the Constant Strain Triangle (CST) element. This element is very commonly used due to the versatility of its geometry, as most two-dimensional shapes can be modeled as an assemblage of triangles. In this case, only in-plane displacements

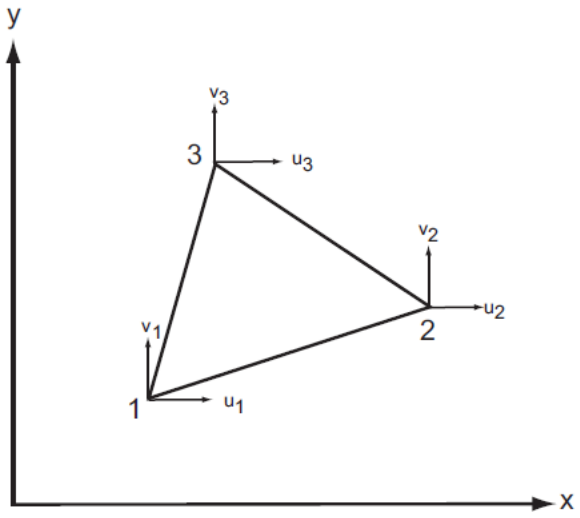


Figure 2.2: Constant Strain Triangle (CST) Element

are considered at the nodes, so each node has two degrees of freedom[1] and as the element has three nodes in total it is said to have six degrees of freedom.

Regarding the displacement within the element, it is specified to be linear which dictates that the resultant shape functions will also be linear polynomials in x and y, with constants dependent solely on the nodal displacement of each element. Also, the linearity in displacement applies along the element edges. Since adjacent elements share a common displacement at each of their two common nodes, then by virtue of the definition of the shape functions they must also have equal displacements all along their common boundary. Hence, the requirement of interelement compatibility is satisfied.

2.2.3 Finite Elements for Thin-Walled Girders

Thin-walled structures such as plates and shells are arguably the most commonly found construction elements, regardless of the scale of the application. This preference to apply walls as thin as possible is a natural optimization strategy to reduce dead load and to minimize construction material (Bischoff, Bletzinger, Wall, & Ramm, 2004). Marine structures are predominantly consisted of thin-walled members, ranging from plates and stiffened panels to essentially the entire hull structure, which is idealized as a thin-walled box girder to analyze its response to longitudinal bending.

Traditionally, two-dimensional plane stress elements have been considered suitable for the modeling of thin-walled structures, and indeed have proven to do so satisfactorily in cases where non-complex problems are concerned. However, when localized effects must be studied, or when members are curved or exhibit geometric discontinuities (e.g. holes) which cause stress concentration, then three-dimensional solid or shell elements must be used.

[1] Degrees of Freedom (DOF's) of any mechanical system are defined as the number of independent variables needed to describe its current state.

The same is in principle true when thin-walled beams subjected to bending loads are concerned. Namely, although using traditional beam elements of the appropriate cross-section is sufficient to simply analyze structural response, the same elements cannot be used when any geometric discontinuity is present. Therefore, in such cases too, three-dimensional solid elements or shell elements must be used. As this thesis is concerned with such a problem, that is a thin-walled girder on which structural damage has been introduced and a bending load is applied, it was deemed useful to provide further information on those element types mentioned before, because they were what was used in creating the finite element model.

In the case of three-dimensional solid elements, they are primarily tetrahedra or hexahedra, with three translational degrees of freedom for each node. They are essentially extensions of plane elements, such as the CST element mentioned before, in that they are built using the same assumptions in defining the shape functions and thus the element's properties, with the addition of another coordinate and displacement component. An 8-node hexahedral element, which is known as an 8-node brick and is one of the most commonly used solid elements, along with its typical nodal degrees of freedom, is shown in Figure 2.3 below.

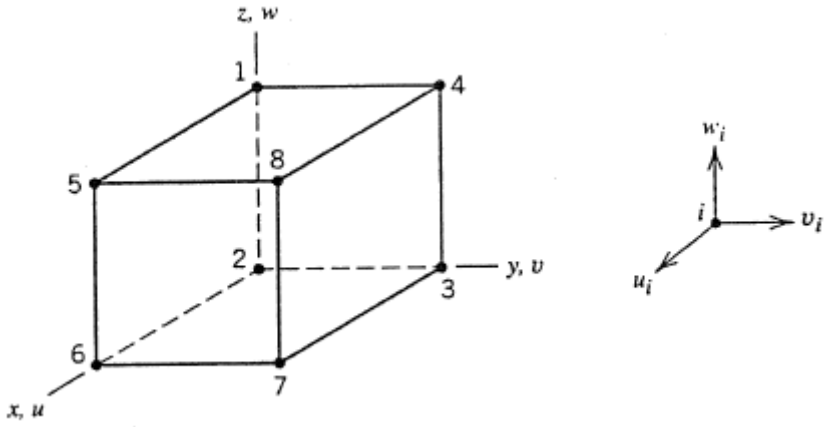


Figure 2.3: 8-node Hexahedral Element and Typical Nodal Degrees of Freedom

Solid elements are very versatile, as they are suitable for modeling any three-dimensional solid and therefore suitable for beam bending, plane stress, simple plate problems and so on, all of which are considered special cases of a three-dimensional solid. Regarding their internal formulation, they support linear displacement distribution in 8-node elements such as the one shown above, as well as higher order elements such as the quadratic 20-node hexahedron, which contains mid-edge nodes in addition to those contained in the 8-node brick.

The other option when modeling thin-walled beams is using shell elements. As is well known, the geometry of a shell is defined by its thickness and midsurface. Therefore, it is advantageous for thin-walled structures of constant thickness to be modelled using midsurface modeling, as it does not require a solid three-dimensional model to be explicitly created. Also, shell elements by virtue of their formulation are able to produce

results for both the outermost and innermost fibers of the surface giving the analyst an advantage against solid elements.

The most commonly used shell elements in modern FEA are flat triangular or quadrilateral elements. In a thin-walled structure, elements will generally be subjected to both bending and in-plane stresses. As described by Zienkiewicz & Taylor (2006), for a flat element these cause independent deformations, provided the local deformations are small, and therefore the ingredients for obtaining the necessary stiffness matrices are available in elements already described in the literature.

Therefore, a flat shell element can be directly obtained by combining a plane stress or membrane element with a plate bending element. Consider the flat rectangular element shown in Figure 2.4 below in a local coordinate system xyz, subject simultaneously to "in-plane" (far-right) and "bending" (middle) actions.

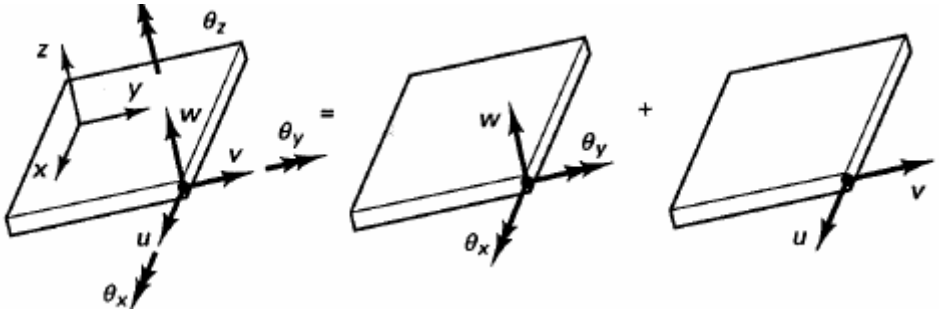


Figure 2.4: Flat Element Subject to "In-Plane" and "Bending" Action.

Considering that stiffness matrices are known for both cases in the local coordinate system and are denoted as $\boldsymbol{K^*_P}$ and $\boldsymbol{K^*_B}$ for the plane stress and plate bending elements respectively, then the stiffness matrix of the resultant shell element can be formulated as follows.

$$K_s^* = \begin{bmatrix} K_P^* & 0 \\ 0 & K_B^* \end{bmatrix} \tag{2.1}$$

It should be noted that $\boldsymbol{K^*_P}$ is an 8×8 matrix while $\boldsymbol{K^*_B}$ is a 12×12 matrix, therefore the resultant matrix $\boldsymbol{K^*_S}$ is 20×20. As these matrices have been derived for a local coordinate system, transformation to the global coordinate system is required to assemble the elements for the entire structure. This is achieved, using a transformation matrix $\boldsymbol{T}$, by the following formula:

$$K_S = T^T K_S^* T \tag{2.2}$$

The resultant element has six degrees of freedom per node, three translations and three rotations around the three main axes for a total of twenty-four degrees of freedom. An alternative way to obtain flat shell elements, is using isoparametric shell elements, which occupy a middle ground between curved and flat shell elements. To obtain them, one starts with a three-dimensional solid element such as the one show in Figure 2.5 -

(a), which for a thickness t small in comparison to the other dimensions, can adequately model a shell. Subsequently, the element is transformed by reducing the number of nodes from 20 to 8, by expressing translational DOF's from the 20-node element in terms of translational and rotational DOF's on the midsurface of an 8-node element such as the one shown in Figure 2.5 - (b). Figure 2.5 - (c) shows these DOF's for the 8-node element at a typical midsurface node.

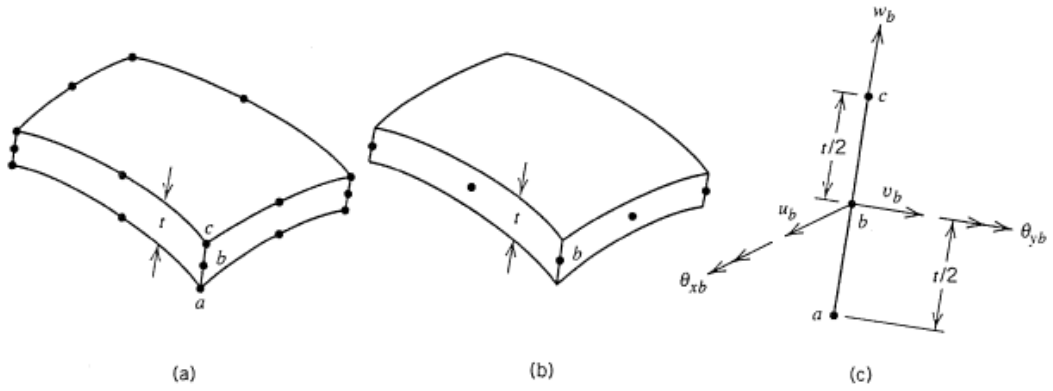


Figure 2.5: 20-Node Isoparametric Solid Element (b) Reduction to 8-Node Shell Element (c) DOF's at Node B

2.2.4 Modeling and Validation

Modeling is the simulation of a physical structure or process by means of a substitute analytical on numerical construct and therefore, does not simply consist of preparing a mesh of nodes and elements. It requires first and foremost, a deep understanding of the physical action of the problem, which in structural applications means understanding the loading and resultant mode of deformation. As mentioned previously, the various types of elements deform in specific ways, so being able to understand the way the entire structure deforms is integral in choosing the suitable type of element for the application.

Also, anticipating the variations and gradient in stress and strain is key in deciding on the density of the mesh used. This must be such as to avoid elements too large which might fail to represent important variations of stress and strain, or too small which are essentially more than needed to accurately represent the stress field, leading to over-refinement and a waste of resources. Furthermore, identifying regions of the structure where stress concentrations appear, such as near openings or the point of load application, is vital as it allows the analyst to decide whether the use of a "graded" mesh, that is a gradation of element sizes in those regions mentioned, is required. Such a technique is very useful as it reduces computational intensity without sacrificing any accuracy. It is however, tied to the experience of the user as no general rule exists on neither the number nor the size of elements which guarantee good accuracy.

To complete the modeling of a structural or other problem using FE analysis, boundary conditions must be prescribed correctly. They include the modeling of both the support conditions of the structure, as well as the loads applied to it. In FEA, loads and boundary

conditions are imposed at the nodes. Therefore, it is often not correct to simply apply the classic notions of support conditions on the FE model, as it will more often than not lead to misrepresenting the actual case. It is thus again made clear, that understanding the actual deformed shape of the structure is critical in this stage also. Regarding the application of loads, again attention must be paid to correctly represent notions from classical mechanics to the FE environment. One such typical example is that of distributed loads; namely, as loads in the FE model are applied at nodes, a distributed load is impossible to model in a continuous way and has to be discretized to equivalent point loads applied at the nodes.

Finally, the last step in the process of building a FE model is to validate its results. Commercial software calculates displacements; strain and stress are subsequently calculated using the displacement information. The resultant displacement field is the first step in verifying whether the model is acceptable, as it allows the analyst to judge whether the deformed shape is as expected or whether boundary conditions have been correctly enforced without any calculations. A further and most crucial stage of validation is to compare the resultant stresses with either analytical or experimental results if the latter are available. However, it is best that this phase of verification is done in advance of FE analysis, as otherwise a tendency often appears to obtain analytical results that agree with the FE ones. If the obtained FE results are reasonable, and the error estimates stemming from the aforementioned comparison are small, then the FE model is acceptable. If not, the process should be repeated after appropriate revisions have been made to the model until the results are satisfactory or cannot be further improved within reason.

2.3 Genetic Algorithms: An Overview

2.3.1 Basic Concepts of Optimization

As is denoted by its title, this section deals with Genetic Algorithms (GAs), a modern, metaheuristic method of optimization belonging to a wider family of modern optimization methods, known as Evolutionary Algorithms (EA). However, before presenting the basic aspects of this method, it was deemed necessary to familiarize the reader with those basic definitions and concepts of optimization in general, which will be needed during the course of this thesis.

According to Rao (2020) an optimization or mathematical programming problem can be generally defined as follows:

$$\text{Find } x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \text{ which minimizes } f(\text{x})$$

Subject to the constraints:

$$g_j(x) \leq 0, \qquad j = 1,2,\dots,m$$

$$l_j(x) = 0, \qquad j = 1,2,\dots,p \tag{2.3}$$

where **x** is an n-dimensional vector called the decision or design variable vector, $f(\mathbf{x})$ is termed the objective function, and $g_j(\mathbf{x})$ and $l_j(\mathbf{x})$ are known as inequality and equality constraints, respectively. The number of variables n and the number of constraints m and/or p need not be related in any way. Also, obviously **x** can be a vector of continuous or discrete variables and f can be either continuous or discrete. Following is a brief description of the practical meaning of the above mathematically expressed terms.

Any engineering system or component during the design stage is characterized by a set of certain quantities, some of which have predefined fixed values. The rest, which are treated as variables during the design process, are known as design or decision variables and are collectively represented by the design or decision variable vector **x** mentioned above. If an n-dimensional Cartesian space with each coordinate axis representing a design variable x_i $(i = 1, 2, \dots, n)$ is considered, the space is called the design variable space or simply design space. Each point in the n-dimensional design space is called a design point and represents either a possible or an impossible solution to the design problem.

In most practical applications the design variables cannot be chosen arbitrarily; rather, they must satisfy certain requirements, functional or otherwise. These restrictions that must be satisfied to produce an acceptable design are known as design constraints and are expressed mathematically by the equality and inequality constraints mentioned earlier. These constrains may be functional or behavioral when they are related to restrictions on the components performance or geometric when related to physical limitations associated with the component.

Traditionally, design procedures aim at finding an adequate or acceptable design that merely satisfies the basic requirements of the problem. In most cases, more than one acceptable design may exist. The purpose of optimization is precisely to determine which one of the designs available is the best one, relative to a specific parameter. Therefore, a measure of that acceptability has to be defined to enable the comparison of the alternative acceptable designs. This measure, with respect to which design is optimized, when expressed mathematically is known as the objective function. It should be noted that the choice of an objective function is governed by the nature of the problem and is of great overall importance in the process of optimization.

In many situations however, problems arise naturally in which more than one criterion must be satisfied simultaneously. An optimization problem involving multiple objective functions is known as a Multiobjective Optimization Problem (MOP). Coello Coello, Lamont, & Van Veldhuizen (2007) describe an MOP as the proccess of finding a vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence,

the term “optimize” means finding such a solution which would give the values of all the objective functions that are acceptable to the decision maker.

Regarding the mathematical definition of this problem, the only difference with the single objective optimization problem is found in the objective function, which is not a scalar but rather is a vector function comprised of several objective functions. These are designated as $f_1(x)$, $f_2(x),\ldots,f_k(x)$, where k is the number of objective functions in the MOP being solved. Finally, the vector function f(x) is defined as follows:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_k(x) \end{bmatrix} \tag{2.4}$$

In addition to the n-dimensional design space mentioned earlier, a k-dimensional space of the objective functions in which each coordinate axis corresponds to a component vector $f_k(\mathbf{x})$ is part of the MOP. Every point in the first space represents a solution and gives a certain point in the second space, which determines the quality of this solution in terms of the objective function values.

The nature of mutiobjective problems dictates that there is not a single objective function to optimize, but rather a set thereof that must be optimized simultaneously. This may involve the minimization or maximization of all functions or a combination of minimization and maximization of them. Consequently, there is no unique solution to the problem but a set of solutions obtained through Pareto Optimality Theory. For their selection, a decision maker is required to make a choice of x_i values. This selection is essentially a compromise (or “trade-off”) between one complete solution **x** and another in the multiobjective space.

These solutions are known as Pareto optimal solutions and are those whose corresponding objective vector componnents cannot be simultaneously improved, that is no vector **x** exists which would decrease some criterion without simultaneously causing an increase in at least one other (assuming minimization). These solutions are also termed non-inferior, admissible, or efficient solutions and their corresponding vectors are termed nondominated. The set of those solutions is known as the Pareto optimal set while that of their corresponding objective vectors the Pareto front set. These solutions may have no apparent relationship besides their membership in the Pareto optimal set and together they form the set of all solutions whose associated vectors are nondominated.

2.3.2 Fundamental Concepts of Genetic Algorithms

In recent years, some optimization methods that are conceptually different from the traditional mathematical programming techniques have been developed. These methods are labeled as modern or nontraditional methods of optimization. Most of these methods are based on certain characteristics and behavior of biological, molecular, and neurobiological systems. In this thesis, only one of these methods is used and its basic

concepts are what this section deals with. The method in question is known in the literature as Genetic Algorithms (GA).

In modern engineering applications, most optimum design problems are characterized by increased complexity due to mixed continuous-discrete design variables and discontinuous and non-convex spaces. If standard nonlinear programming techniques are used for this type of problem, they will be inefficient, computationally expensive, and, in most cases, find a relative optimum that is closest to the starting point. Genetic algorithms (GAs) are well suited for solving such problems, and in most cases, they can find the global optimum solution with a high probability.

As their name suggests, GAs draw their inspiration from Darwin's theory of evolution. They are based on the principles of natural genetics and natural selection. The basic elements of natural genetics – reproduction, crossover, and mutation – are used in the genetic search procedure. Following is an overview of the basic concepts of GAs as well as the way they are used for the solution of an optimization problem. In Figure 2.6 below a schematic detailing what is discussed afterwards is provided.

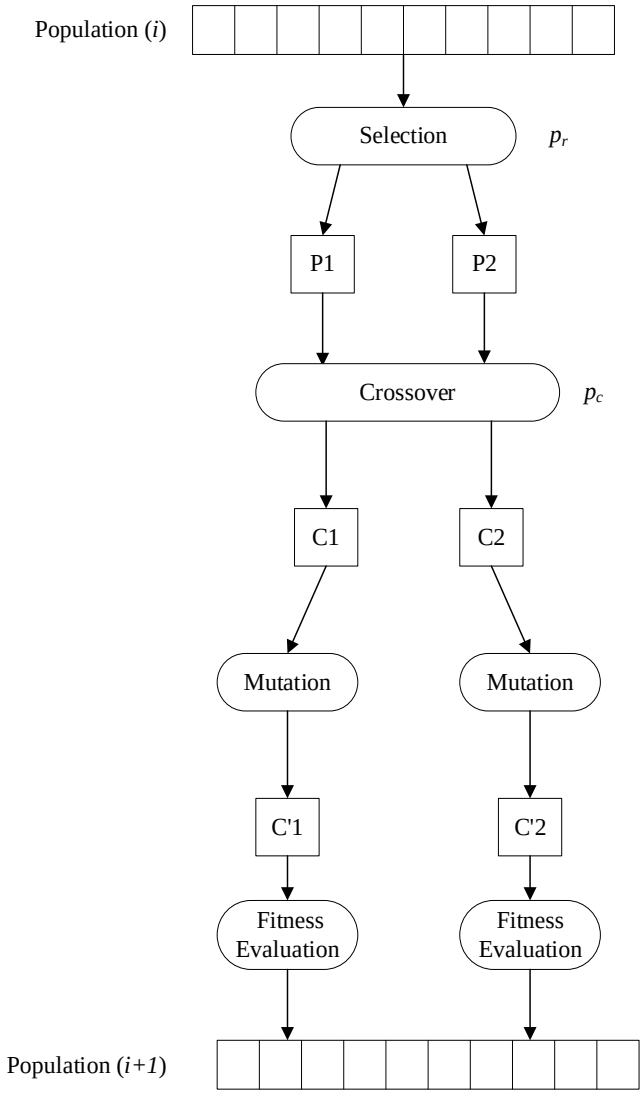


Figure 2.6: Schematic of Basic GA Concepts and Process

Design variables in GAs are represented as strings of binary numbers, 0s and 1s, that correspond to chromosomes in natural genetics. This makes the method naturally applicable for the solution of discrete problems. Obviously, it is also applicable for continuous variables where the length of the binary string may be adjusted to achieve the desired resolution.

Because genetic algorithms are based on the survival-of-the-fittest principle of nature, they try to maximize a function called the fitness function. The fitness function, $F(\mathbf{x})$, can be taken to be same as the objective function $f(\mathbf{x})$ of a maximization problem so that $F(\mathbf{x}) = f(\mathbf{x})$. A minimization problem can be transformed into a maximization problem before applying the GA. Usually the fitness function is chosen to be nonnegative. The commonly used transformation to convert an unconstrained minimization problem to a fitness function is given by the following formula:

$$F(x) = \frac{1}{1 + f(x)} \tag{2.5}$$

This formula does not alter the location of the minimum of $f(\mathbf{x})$, but it does successfully convert a minimization problem to an equivalent maximization problem (Rao, 2020).

The solution of an optimization problem using a GA begins with an initial sample, known as a population, of random design vectors. These vectors, known as individuals, are represented by a string of chromosomes, in keeping with the theme of natural selection. The size of the population is usually fixed and is defined by the user. Then, each individual is evaluated to find its fitness value. After their fitness values have been evaluated, three genetic -as they are called- operators act on the individuals. These operators, again using a biological analogy are reproduction, crossover, and mutation.

Reproduction is the first operation applied to the population to select fit individuals to form a mating pool. Individuals with above average fitness are selected from the population and multiple copies of them are inserted to the mating pool, based on a probabilistic procedure. A common reproduction operator assigns a probability of selection, p_r, to an individual proportional to its fitness. This means that individuals with high fitness values in the population, probabilistically, get more copies in the mating pool. No new strings, i.e. individuals, are formed in this stage; only existing ones in the population are copied to the mating pool. The reproduction stage ensures that highly fit individuals survive and reproduce while less fit ones become extinct, thus mimicking the natural principle of "survival-of-the-fittest".

After reproduction has been completed and the mating pool formed, the crossover operation takes place with the purpose of creating new strings by exchanging information from individuals existing in the mating pool. The individuals selected to participate in the crossover operation are known as parents while those generated from them are known as child strings. In most crossover operations, two individual strings, i.e. decision variables, are randomly selected from the mating pool generated during reproduction and some portions of the strings are exchanged between them. More

specifically, in the most common process known as single-point crossover, a crossover site is chosen at random along the strength length, and all binary digits (alleles) lying on one side of it are exchanged between the two strings.

Since the crossover operator combines chromosomes from parents with good fitness values, it is expected that their offspring will be better in that regard, provided that a suitable crossover site is selected. However, this is not known in advance and thus is selected randomly. It is therefore possible, that the offspring may have worse fitness than their parents. This does not ultimately affect the success of the GA, as the less fit offspring generated will not survive long and will possibly be eliminated in the next reproduction stage. This ambiguity in the result of the crossover operation, indicates that not all individuals from the mating pool should be used; rather, the better ones should be preserved as part of the population in the next generation. This is achieved in practice using a crossover probability, p_c, to select the parent strings from the mating pool.

The final genetic operator is mutation. As was mentioned above, crossover is the main operator responsible for producing new strings, known as offspring which constitute a new generation of the population. The mutation operator is applied on those new strings, with a small mutation probability, p_m, changing the binary digits from 0 to 1 and vice versa. There can be either single-point mutation, where a mutation site is chosen randomly along the length of the string and the binary digit at that site changed, with a probability p_m and also there can be bit-wise mutation where each bit in the string is considered separately in a sequence, and its value is changed with regard to a mutation probability p_m. The purpose of mutation is; first, to generate an individual, that is a string or design point, in the neighborhood of a current individual, thereby accomplishing a local search around the current solution; also, to safeguard against a premature loss of important genetic material at a particular position; finally, to maintain diversity in the population.

Summarizing the above, the reproduction operator selects individuals with above-average fitness for the mating pool, the crossover operator recombines the substrings of good strings (individuals) of the mating pool to create offspring, i.e. the next generation of the population, and the mutation operator alters some individual's genotype locally. This cycle of reproduction, crossover, and mutation and the evaluation of fitness values is known as a generation. The use of these three operators successively yields new generations with improved values of average fitness of the population. The convergence of the method is tested by a relevant criterion, which most commonly is a permissible value of the standard deviation of the population's fitness and secondarily the maximum number of generations. The population is iteratively operated on by the three genetic operators and the new generation evaluated for its fitness values, until the convergence criterion is satisfied, and the process is terminated.

2.3.3 Multiobjective Genetic Algorithm (MOGA)

Evolutionary algorithms, and by extension Genetic Algorithms, seem particularly suitable to solve multiobjective optimization problems, because they deal simultaneously with a set of possible solutions (the so-called population). This allows them to find several members of the Pareto optimal set in a single "run" of the algorithm, instead of having to perform a series of separate runs as is the case in traditional mathematical programming techniques (Coello Coello, Lamont, & Van Veldhuizen, 2007).

An adaptation of GAs suited for multiobjective problems, called "Multi-objective Genetic Algorithm" (MOGA), was proposed by Fonseca & Fleming (1993). In their work, they introduced a method that uses the notions of Pareto optimality, which was described in Section 2.3.1 earlier, to evaluate the fitness of the population. As has been mentioned numerous times in this section, the individuals comprising a population in a GA, are essentially solution vectors of the optimization problem. Therefore, in a multiobjective problem, through Pareto Optimality Theory they may be classified according to their dominance.

Fonseca & Fleming (1993) defined a ranking system based on the dominance of individuals within their generation. Consider an individual x_i at generation t, which is dominated by $p_i^{(t)}$ individuals in this current population. This individual, is assigned a rank according to the following:

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{2.6}$$

Therefore, all non-dominated individuals are assigned a rank of one and so on. After the ranks of all individuals have been defined, the assignment of fitness is done the following way. First, the population is sorted according to rank. Then, fitness is assigned to each individual, interpolating from the best (rank 1) to the worst using an adequate function usually but not exclusively linear. Finally, the fitness of individuals with the same rank is averaged, so as to sample all of them at the same rate.

The assignment of fitness is the essential difference between a single-objective and a multiobjective GA. Afterwards, the genetic operators are applied in exactly the same way on the individuals. However, another point of difference may arise in some cases in the criterion used to terminate the process. It has been mentioned previously, that in single-objective problems, a permissible value of the standard deviation of population fitness and the number of generations are common termination criteria. In addition to these, in MOPs a Pareto percentage criterion is used. This essentially refers to the percentage of Pareto optimal solutions in a population, at a given generation and it ensures that the process is terminated only when an acceptable number of individuals are distributed in Pareto optimal regions.

2.4 Artificial Neural Networks: An Overview

2.4.1 Basic Concepts

The human brain contains a highly interconnected set of some 10^{11} neurons to facilitate cognitive and neurological functions. Each of the biological neurons has the complexity of a microprocessor. Remarkably, the human brain is a highly complex, nonlinear, and parallel computer. It has the capability to organize its structural constituents, that is, neurons, so as to perform certain computations many times faster than the fastest digital computer in existence today (Keller, Liu, & Fogel, 2016).

The study of Artificial Neural Networks (ANNs) has been inspired in part by the above observation, that is that biological learning systems are built of very complex webs of interconnected neurons. In rough analogy, artificial neural networks are built out of a densely interconnected set of simple units, where each unit takes a number of real-valued inputs (possibly the outputs of other units) and produces a single real-valued output (which may become the input to many other units). Although the motivation behind ANNs are biological neural systems, there are many complexities to them which are not modeled by ANNs; also, many features of ANNs are not consistent with what is known to be true for biological systems. For example, in many ANNs their individual units output a single constant value, whereas biological neurons output a complex time series of spikes (Mitchell, 1997).

As previously mentioned, ANNs are composed of a large number of simple computing units, which following the pattern of biological imitation, are known as neurons. The strength of the connection, or link, between them is known as the weight. The values of those weights are an important network parameter and are the subject of the ANN's learning procedure. Neurons are typically organized into layers in which all neurons usually possess the same activation functions. To be considered a genuine neural network, an ANN must have an input layer and at least two layers of neurons; a hidden layer and an output layer.

Considering what was mentioned earlier, it is made apparent that ANNs aim to derive their computing power through a massively parallel distributed structure and also an ability to learn and therefore generalize. Learning refers to the process through which the weights in the network are adjusted to improve the accuracy of the results. The characteristic of generalization refers to the neural network producing reasonable outputs for inputs that were not encountered during training. These two information processing capabilities make it possible for neural networks to solve complex and large-scale problems that are currently intractable. In the following sections, an attempt is made at presenting such concepts as neurons, network architecture and learning in a brief yet rigorous manner.

2.4.2 Models of a Neuron

A neuron is an information processing unit that is fundamental to the operation of an ANN. In Figure 2.7 below, a typical neuronal model is shown.

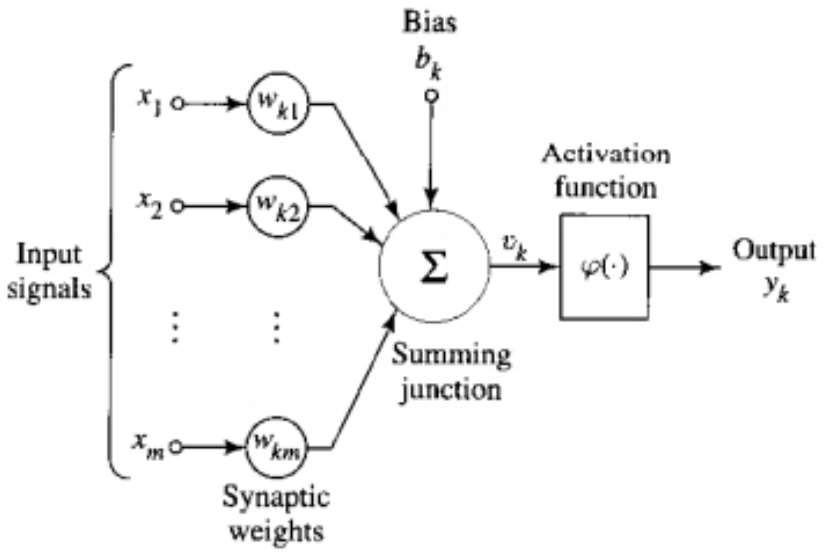


Figure 2.7: Non-Linear Model of Neuron

The three basic characteristics of a neuronal model are identified by Haykin (1998) as being the following:

1. A set of synapses or connecting links, each of which is characterized by a weight or strength of its own. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj}, where the first subscript k refers to the neuron in question, while j refers to the input end of the synapse to which the weight refers.
2. An adder for summing the input signals, weighted by the respective synapses of the neuron.
3. An activation function for limiting the amplitude of the output of a neuron. Typically, the activation function normalizes the neuron's output amplitude to the unit interval [0,1] or alternatively to [-1,1].

Essentially, a neuron receives a set of input signals -which can either be input data or outputs from other neurons- which are then weighted according to the weight corresponding to each synapse. As mentioned earlier, this weight is the result of the learning procedure of the ANN. Then, the weighted input signals are summed and in the case of a neuron such as the one shown here, an externally applied bias b_k is added. The effect of this bias is to either increase or lower the net input of the activation function. In biological terms, one can think of the bias as a measure of how easy it is to get the neuron to fire. A neuron with a large positive bias is extremely likely to fire, while a large negative bias yields the opposite result. Mathematically, a neuron k may be expressed as follows:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.7}$$

and

$$y_k = \varphi(u_k + b_k) \tag{2.8}$$

Where $x_1, x_2, \ldots, x_m$ are the input signals; $w_{k1}, w_{k2}, \ldots, w_{km}$ are the synaptic weights of the neuron k; u_k is the linear combiner output due to the input signals; b_k is the bias; φ is the activation function; and finally, y_k is the output signal of the neuron. The use of the bias b_k has the effect of applying an affine transformation to the output u_k of the linear combiner of the model shown in Figure *2.7*, as shown by:

$$v_k = u_k + b_k \tag{2.9}$$

In particular, depending on whether the bias b_k is positive or negative, the relationship between the induced local field or activation potential v_k of neuron k and the linear combiner output u_k is modified accordingly. However, the bias b_k is an external parameter to the network and its presence is accounted for in Eq. (2.9). To account for the presence of the bias, without considering it as an external parameter of the network, the model of the neuron is reformed in the manner shown in Figure 2.8.

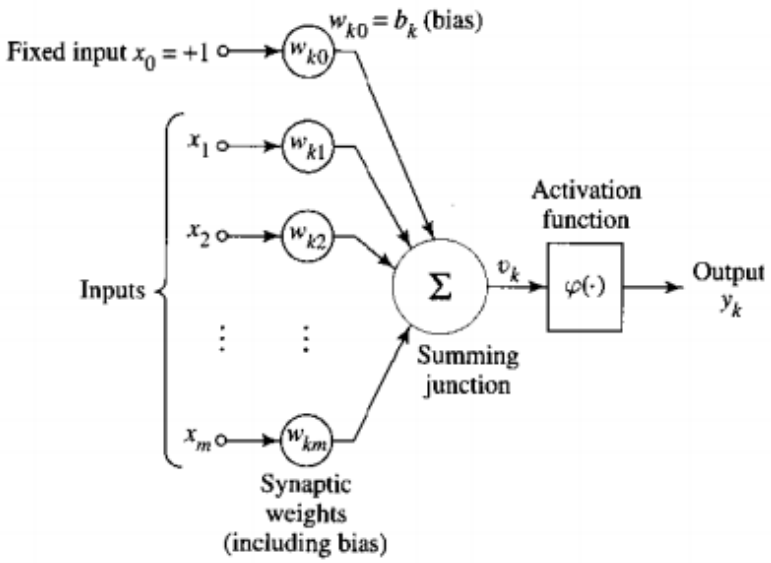


Figure 2.8: Reformed Non-Linear Model of Neuron

Essentially, a new synapse is added with an input $x_o = +1$ and a weight equal to the bias, $w_{k0} = b_k$. Therefore, the presence of bias is accounted by adding a new input signal fixed at +1 with a new synaptic weight equal to the bias b_k. Equivalently, the equations previously presented are reformulated and the model of the neuron can now be described mathematically as follows:

$$v_k = \sum_{j=0}^{m} w_{kj} x_j \tag{2.10}$$

And

$$y_k \approx \varphi(v_k) \tag{2.11}$$

To complete the description of the model of a neuron used in ANNs, the role and types of the activation function must be addressed. As mentioned before, the activation function is denoted by *φ(υ)* and it defines the output of the neuron with respect to the induced local field *υ*. This function essentially normalizes the output of the neuron to a given interval, most commonly the unit interval; also, it is defined in such a way that it ensures the network's ability to learn during training.

The simplest form of an activation function is known as the threshold function. This function is mathematically expressed in the following way:

$$\varphi(v) = \begin{cases} 0, & v < 0 \\ 1, & v \geq 0 \end{cases} \tag{2.12}$$

The output of a neuron employing such an activation function is given by:

$$y_k = \begin{cases} 0, & v_k < 0 \\ 1, & v_k \geq 0 \end{cases} \tag{2.13}$$

Where v_k is the induced local field of the neuron, given by:

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{2.14}$$

Such a neuron is known in the literature as a perceptron or Rosenblatt neuron and is the simplest model of a neuron found; however, it is considered greatly influential in the advancement of neural networks.

The perceptron model, although groundbreaking, was limited to the solution of a small class of problems. The reason behind that lies precisely in the nature of its activation function. Generally, it was previously mentioned that the way ANNs learn, is by adjusting synaptic weight values using specialized algorithms and methods, such as the backpropagation algorithm, which itself is a special case of gradient decent. The goal is to ultimately minimize a performance function *P*, which most commonly is defined as an appropriate norm of the difference between the desired output of the network and its actual output. Therefore, the direction of the gradient, that is the most rapid decent, is followed in a space that gives *P* as a function of the weights *w*. So, it is obvious that to obtain valid results the function *P* must be differentiable, which in the case of the perceptron is not.

The answer to this problem was given by the sigmoid neuron. The sigmoid neuron draws its name from the activation function it employs, the sigmoid function, an example of which is shown below.

$$\varphi(v) = \frac{1}{1 + e^{-v}} \tag{2.15}$$

Where v is the same as in Eq. (2.14). So, in the limit when υ becomes large and positive, the sigmoid approaches unity, while on the contrary when υ becomes large and negative it approaches zero. In that sense, it indeed behaves in the same way as a perceptron does. At the same, its aforementioned differentiability as well as some interesting properties of its derivatives make it a very appealing choice to be used as an activation function. And indeed, the sigmoid activation function is by far the most commonly used in the construction of artificial neurons.

2.4.3 Architecture of Artificial Neural Networks

At this point, having provided a familiarization with the concept and design of an artificial neuron, it was considered important to provide some basic knowledge on the architecture of ANNs, in order to complete this overview of the relevant subject. In the context of this thesis, two different classes of network architectures are of interest: *Single-Layer Feedforward Networks; and Multi-Layer Feedforward Networks*

Single-Layer Feedforward Networks

In a layered neural network, neurons are organized in the form of layers. In the simplest form of a layered neural network, an input layer of source nodes projects onto an output layer of nodes (computation nodes), but not vice versa. Such a network, having four nodes in both the input and output layer is shown in Figure 2.9 below.

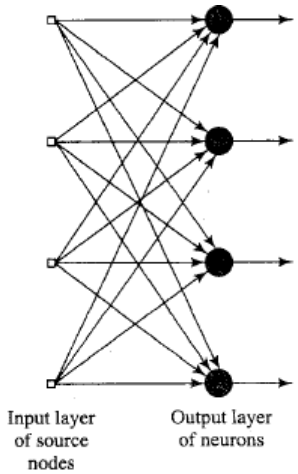


Figure 2.9: Single-Layer Feedforward Network

In other words, this network is strictly a feedforward or acyclic type. Such a network is called a single-layer network, with the "single-layer" designation referring only to the output layer of neurons (or computation nodes); the input layer is not counted as no computations are performed there.

Multi-Layer Feedforward Network

The second class of a feedforward neural network is distinguished by the presence of one or multiple hidden layers, whose computation nodes are correspondingly known as hidden neurons. Their function is to intervene between the external input and the network output in some useful manner. By adding more hidden layers, the network is enabled to extract higher-order statistics, an ability which proves invaluable in cases when the size of the input layer is large.

The way this type of network works, is described by Haykin (1998) in the following manner. The neurons in the input layer supply respective elements of the activation pattern, i.e. the input vector, which constitute the input signal applied to the neurons of the second layer (i.e. the first hidden layer). The output signals of the second layer are then used as inputs to the third layer, and so on for the rest of the network. Generally, neurons in each layer receive as inputs the output signals of the preceding layer only. The set of output signals in the output (final) layer of the network, constitutes the overall response of the network to the activation pattern supplied by the neurons of the input layer. An example of such a network is shown in Figure 2.10, containing only one layer of hidden neurons. This type of network is known as fully connected, meaning that every neuron in each layer of the network is connected to every other neuron in the adjacent forward layer.

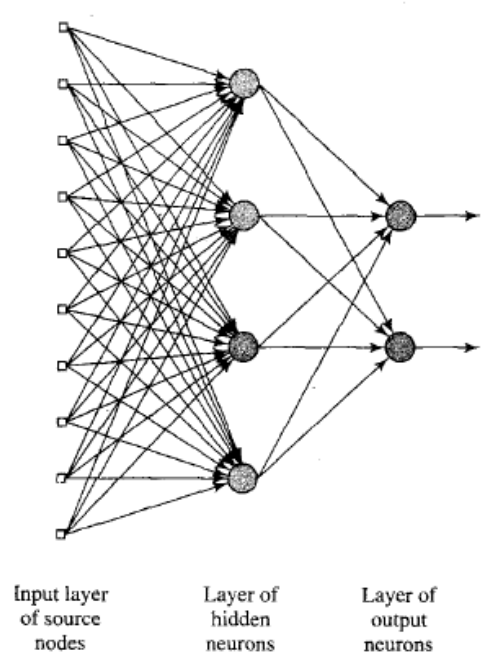


Figure 2.10: Feedforward Neural Network with a Single Hidden Layer of Sigmoid Neurons

3 Development of a Digital Twin-Based SHM System

3.1 Introduction

The primary focus of this chapter is to detail the process that was followed for the development of a FE-based structural digital twin of a thin-walled square girder subjected to three-point bending. This was then used to develop an SHM system based on modern optimization and machine learning techniques. The chapter is organized as follows.

In Chapter 3.2, the reasons for selecting this particular case are explained, on the basis of its relation to the study of hull girder structural response to still-water loading. Having established this conceptual connection as the basis of the digital twin, in Chapter 3.3 the author goes into further detail on its development using FE analysis. The process of constructing the model, the selection of elements, its discretization and meshing and finally the application of boundary conditions and loading are thoroughly explained. In Chapter 3.4, the fundamental design of the SHM system is addressed. An additional FE model built to assist in determining the monitoring scheme is presented and the reasoning behind it explained. In Chapter 3.5, potential damage is introduced to the digital twin by means of a Strain Field Disturber, whose nature and purpose are described in detail. Finally, the SHM system formulation is completed by introducing the methods that were developed to detect damage, based on Genetic Algorithms and Artificial Neural Networks, and detailed in Chapters 3.6 and 3.7, respectively.

3.2 Selection of an Adequate Simplified Digital Twin Model

Modern ships and offshore structures are some of the most complex structural assemblies known to man. This complexity may be mainly attributed to:

- The fact that they are inherently complex structures, thus requiring a very detailed Finite Element model to accurately recreate the changes in geometry and complexity of the various structural components.
- The fact that the structural loads they are subjected to, mainly due to the cargo they carry as well as their own weight and sea pressure, vary over time as well as along the structure itself. This makes detailed modeling of these structural loads an equally demanding task.
- The environment on which a ship operates, i.e. the sea, which makes the description of the boundary conditions imposed on the structure significantly more complex.
- Finally, the non-linearity of the response, due to the non-linearity of materials or the presence of large deformations.

Thus, it is made clear that a detailed modeling of the entire hull structure, taking into account at the same time all the different loads and responses of its various components, is a very demanding task requiring great effort and expertise as well as computational power. In addition to that, it is practically impossible at this point to carry out experiments on actual ship structures, in order to obtain the measurements needed to create a data-enabled structural digital twin. Therefore, such a model was deemed to not be within the scope of this present work and an adequate alternative was sought in the classical methods used in the study of ship strength and especially, in the concept of the hull girder.

Traditionally, the study of ship strength is compartmentalized in various stages, during which the entire structure or certain of its components are considered as if subjected individually to some form of structural loading. In the case when the ship is afloat in still water, static loads act on it due to the combination of gravity forces, i.e. weight, and buoyancy forces. Buoyancy forces are essentially the resultant forces due to the distribution of hydrostatic pressure on the immersed external area of the ship. A typical case of this type of loading, as well as the resultant shear force *(Q)* and bending moment *(M)* distributions are shown in Figure 3.1 from Mansour & Liu, (2008).

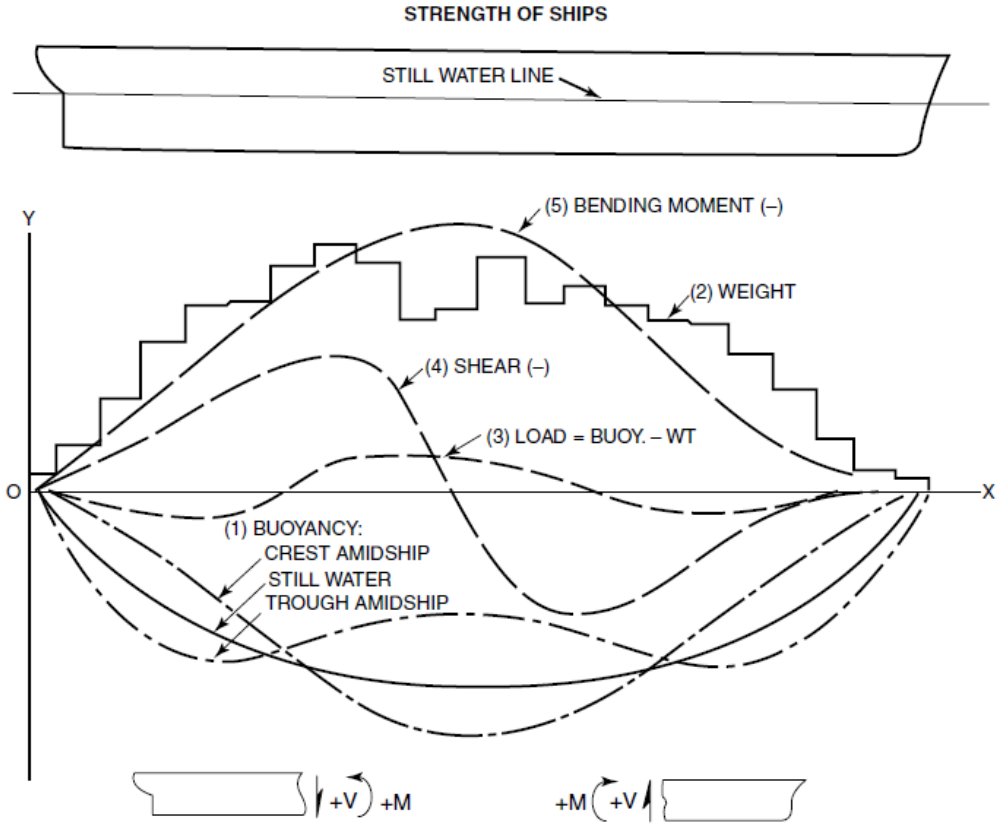


Figure 3.1: Ship Static Loads, Shear Force and Bending Moment Distributions

The study of the response of the entire structure to this type of loading is traditionally the subject of hull girder theory. In this, the ship's hull is considered to be an idealized thin-walled girder, known as the hull girder. The study of the ship as the hull girder

deals with the bending of the ship's hull on the vertical and horizontal planes, due to longitudinally distributed lateral and vertical bending loads, as well as torsion due to twisting loads. This particular idealization was what provided the necessary theoretical connection for the final decision made in the context of this thesis, on what would be the adequate alternative to a detailed hull model, considered unfeasible due to the reasons described earlier.

In particular, the following simplified model was adopted for the purpose of this thesis. That is, to model the geometry of the hull as a thin-walled square girder and use that simpler, less demanding yet theoretically sound numerical model as the basis for the development of the digital twin and the corresponding SHM system. However, to complete the simplified equivalent model, this geometrical assumption is not enough. A structural load to be applied must be sought in order to produce a complete, equivalent numerical model. To explain the decision that was made on this matter, first it was considered necessary to provide some additional context behind the overall scope of the methods developed in this present work.

While this thesis is largely concerned with the numerical treatment of the concept of the digital twin as an enabler of structural damage detection, it also aims to make initial steps towards a truly data-enabled structural digital twin. The only way to achieve that is by obtaining actual data, measured on the structure itself. As mentioned before, this along with reasons regarding structural complexity, were the main reasons behind the decision to develop a simplified model. So, the loading and boundary conditions to be applied were chosen to enable the execution of laboratory tests within the capabilities of the Shipbuilding Technology Laboratory. This was made possible by the decision to model the hull structure subjected to still water loads, as a thin-walled square girder subjected to three-point bending.

To further substantiate this assumption, it is important to demonstrate why this simplified load scenario is adequate. According to Euler-Bernoulli beam theory, for the case of a simply supported beam subjected to a concentrated load applied in the middle of the beam i.e., the case of the three-point bending test, the shear forces (Q) and bending moments (M) are given by the following equations:

$$Q(x) = \begin{cases} \frac{P}{2}, & for\ 0 \le x \le L/2 \\ -\frac{P}{2}, & for\ L/2 < x \le L \end{cases} \tag{3.1}$$

$$M(x) = \begin{cases} \frac{Px}{2}, & for\ 0 \le x \le L/2 \\ \frac{P(L-x)}{2}, & for\ L/2 < x \le L \end{cases} \tag{3.2}$$

In addition to the above, a schematic of the three-point bending case as well as the resultant shear force and bending moment distributions, are shown in Figure *3.2*.

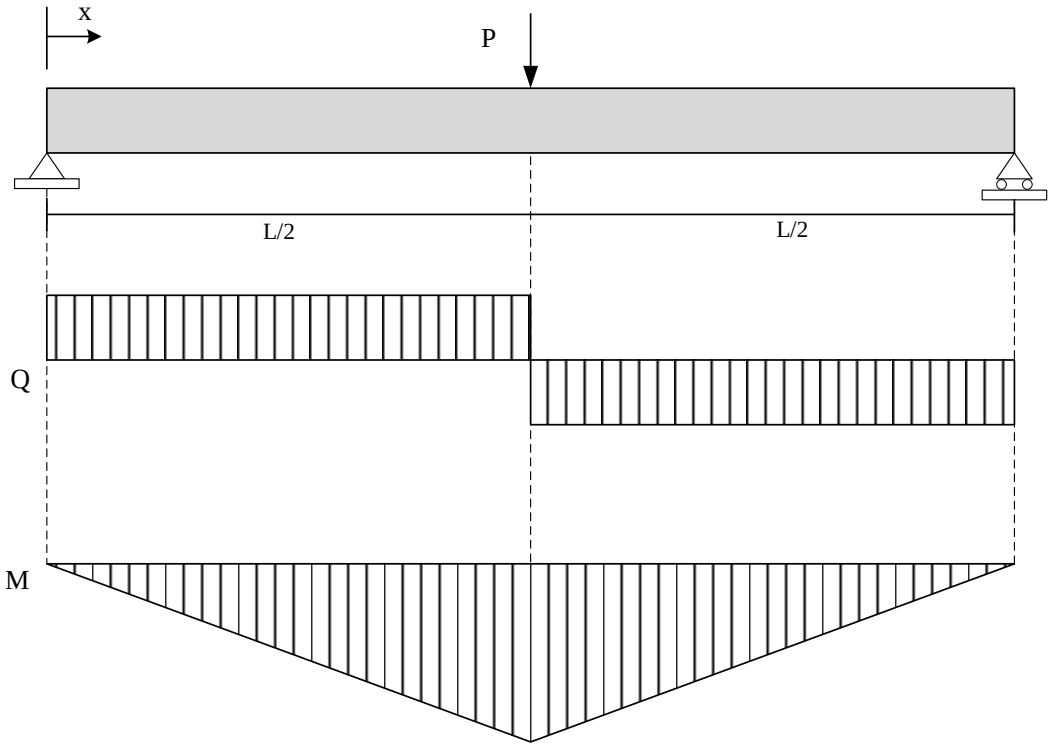


Figure 3.2: Three-Point Bending, Shear Force (Q) & Bending Moment (M)

Comparing Figure 3.2 and Figure 3.2, it becomes evident that bending moment distribution in the three-point bending case may be considered as an adequate approximation of the longitudinal bending moment distribution in actual ship hulls. Also, let us be reminded that the classic formula for determining the bending stress of a beam under simple bending is given by:

$$\sigma_x = \frac{M_y}{I_y} z \tag{3.3}$$

Where M_y is the bending moment about the neutral axis y, I_y is the second moment of area with respect to the neutral axis y and z is the perpendicular distance to the neutral axis. Therefore, Eq. (3.3) shows that the longitudinal distribution of bending stress will be similarly equivalent, and thus the stain field also. That essentially concludes the reasoning behind the decision to use these loading and boundary conditions to complete the simplified model which provided the basis for the digital twin and subsequently developed SHM system.

3.3 Development of the Thin-Walled Girder FE Model

The simplified digital twin model, which was chosen due to the reasons detailed in the previous section and which will hereafter be simply addressed as the digital twin, was developed using the commercial FE software ANSYS 19.2, and more specifically using its built-in programming language, APDL. The reason why APDL scripting was chosen as the way to develop the model, was that it allows for it to be parametrically built as

well as because it allows for certain post-processing functions to be included in the code, which were vital to the SHM methods that were developed and will be detailed further in due course. At this point, a detailed description of the digital twin is made.

As mentioned before, the digital twin is a hollow square beam, subjected to three-point bending. Regarding the geometric and material properties of the beam, they were chosen based on commercially available standards. This was done with the prospect of laboratory experiments in mind, as well as to maintain realistic conditions. The nominal geometric and material properties used in the digital twin FE model, are shown in Table 3.1 below.

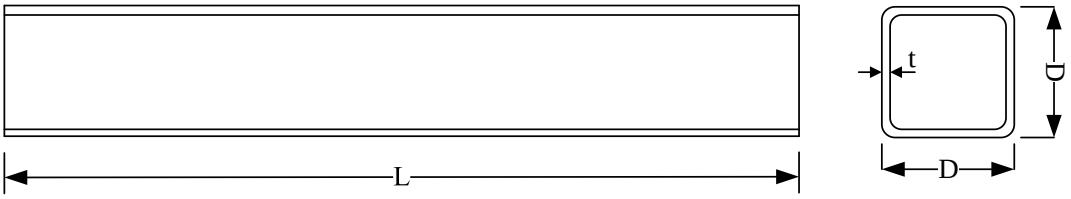


Figure 3.3: Schematic of Thin-Walled Girder used for the FE Model

Table 3.1: Nominal Geometric and Material Properties of Thin-Walled Girder used in FE Model

Physical Properties					Material Properties	
Cross-Sectional Properties				Length	Young's Modulus	Poisson's Ratio
D	t	A	I	L	E	ν
[mm]	[mm]	[mm^2]	[mm^4]	[mm]	[Mpa]	-
100	2.5	952	1486400	1500	207000	0.3

Also, a three-dimensional model of the thin-walled girder digital twin, developed with ANSYS is provided in Figure 3.3 below. Note that the compartmentalization of the model was done to enable the realistic application of boundary conditions and loads.

Having established its physical and material properties, the next step in the development of the FE model is to select the types of elements to be used. In this case, as the ultimate goal is the inclusion of damage, a detailed model is required. Therefore, the use of simple beam elements was rejected from the onset and it was made clear that the use of solid or shell elements would be the more qualified choice. It was decided, that in order to have a multitude of options which could be compared later and given that the model is simple enough, it would be developed using both types of elements.

More specifically, eight models in total were created using both linear and higher order solid and shell elements. The solid elements featured in ANSYS which were used were: SOLID 185, an eight-node brick element with three translational nodal degrees of freedom and SOLID 186, its higher order counterpart, which is a twenty-node brick

element having the same nodal DOFs. The shell elements supported by ANSYS that were used were: SHELL 181, a four-node shell element with six DOFs per node, three translations and three rotations; and SHELL 281, its higher order variant which contains eight nodes having the same six DOFs.

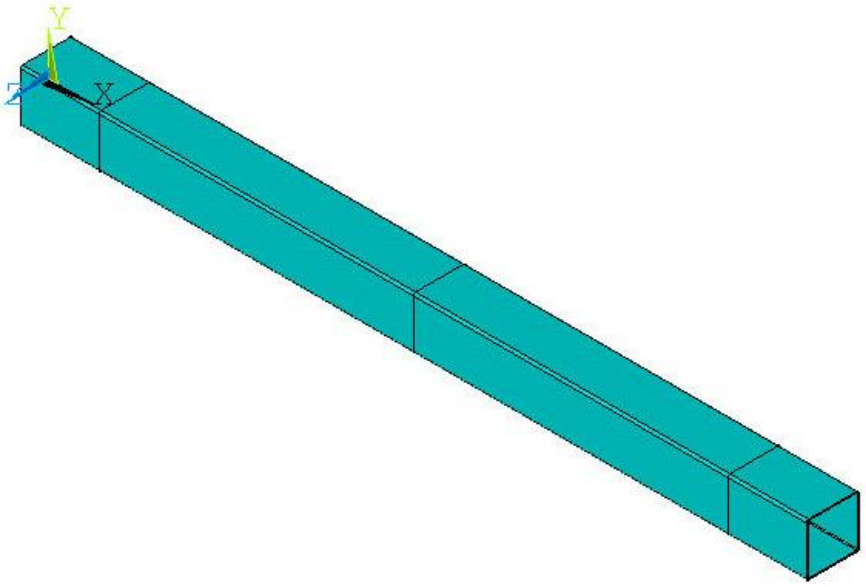


Figure 3.4: Three-Dimensional Thin-Walled Girder Geometric Model

With the element types selected, the next step in the process is the discretization, or meshing, of the model. For each model, it was decided to examine two different instances of meshing to compare both accuracy and more importantly solution time. This latter feature is of vital importance in this work, as the SHM methods developed later will require this model to be solved numerous times, and therefore speed is essential to create a versatile and computationally advantageous method. Also, it should be noted that in the case of the solid elements, a three-dimensional model such as the one shown in Figure 3.4 was used for meshing, while in the case of the shell elements, it was the midsurface of the beam that was meshed.

As the different element types used to develop the model ultimately required different boundary conditions to realistically describe the problem, it was considered best at this point to continue this description in two separate sub-sections, one for each overarching element category. Therefore, the meshing approaches previously described will be further detailed separately for solid and shell elements, along with the application of boundary conditions and loads as well as some indicative results for the validation of the models.

Solid Element Model

As mentioned before, both for lower and higher order elements, two different meshes were examined, a finer one and a coarser one. In the case of the fine mesh, an edge length of 5 mm was assumed for all web and flange elements. For the fillets, this was reduced to 2.5 mm in order for the solid elements to more accurately capture the curvature. In the case of the coarse mesh, the edge length was set to 10 mm for elements

in web and flange areas, while the same size as before was maintained for the fillets. In Figure 3.5, an example of the solid modeling is shown, where the coarse mesh described above has been applied.

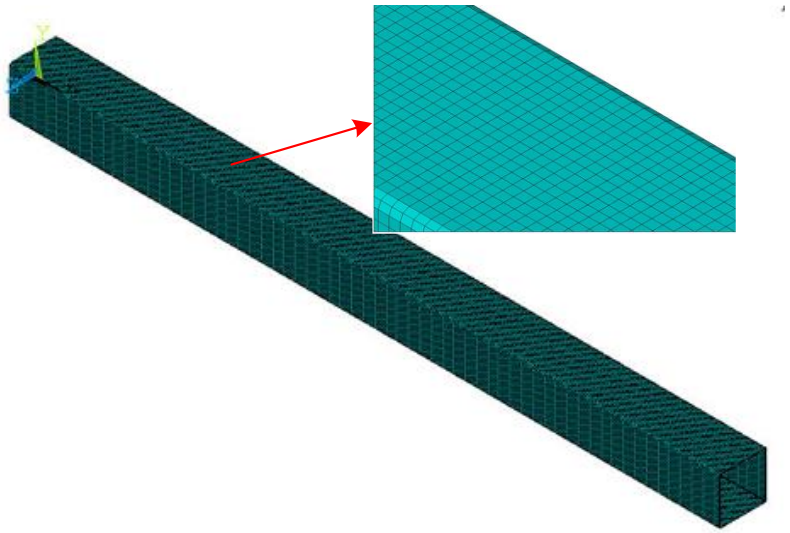


Figure 3.5: Solid Element Model

Regarding the boundary conditions and applied load, these were different with respect to the elements used. They were the same across all four solid element models, whether lower order or higher order elements had been used. More specifically, to simulate the applied load P typically associated with three-point bending (see Figure 3.2), a constant displacement $w=1.5\ mm$ was applied on the line where the midspan of the beam intersects with the top flange. For the supports, applied at 150 mm from each end, so that the span was effectively 1200 mm, the boundary conditions were modeled in the following way: Vertical displacement u_y was constrained at nodes symmetrically located at the edges of the bottom flange, on the longitudinal positions specified earlier. That way, the condition of no vertical displacement was met, symmetrical loading was achieved and at the same time the model did not become over-constrained. To satisfy the condition of no axial displacement, the corresponding translational DOF was constrained at the same nodes as the vertical displacement, however this time only on one support. This was done because the loading was such that no axial loads were applied, so constraining one support only was sufficient to satisfy the condition. Finally, to ensure that torsional rotation and rotation on the horizontal plane were constrained, transverse displacement was constrained at only one of the aforementioned nodes, as shown in Figure 3.6. As no loads were applied which may cause either twisting of the beam or bending in the horizontal plane, constraining only this one DOF was sufficient to satisfy the boundary condition.

Having modeled the applied load and boundary conditions, the FE model was solved, and the results obtained from it were compared to values expected from classical beam theory. Specifically, maximum axial stress was compared for the two cases; namely, in the FE model axial stress results were obtained for a path along the centerline of the top flange, while Eq. (3.2) and Eq. (3.3) were used for the theoretical estimation. It must be noted that for the implementation of these equations, the relevant values from Table 3.1 were used where needed and the point load P was substituted with an equivalent static

load obtained from FEA. In Figure 3.7, these stresses, denoted as $\sigma_{D.T.}$ are plotted, normalized to the equivalent theoretical estimate, denoted as σ_{theory}, with regards to longitudinal position x, also normalized to the overall length L. This was done for all models developed using solid elements. In Figure 3.7 their corresponding results are denoted by the element type and the mesh used in each case.

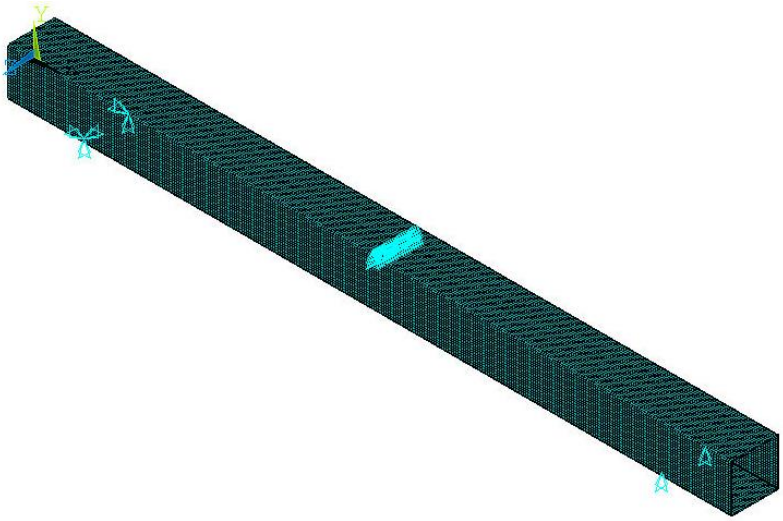


Figure 3.6: Solid Element Model with Applied Loads and Boundary Conditions

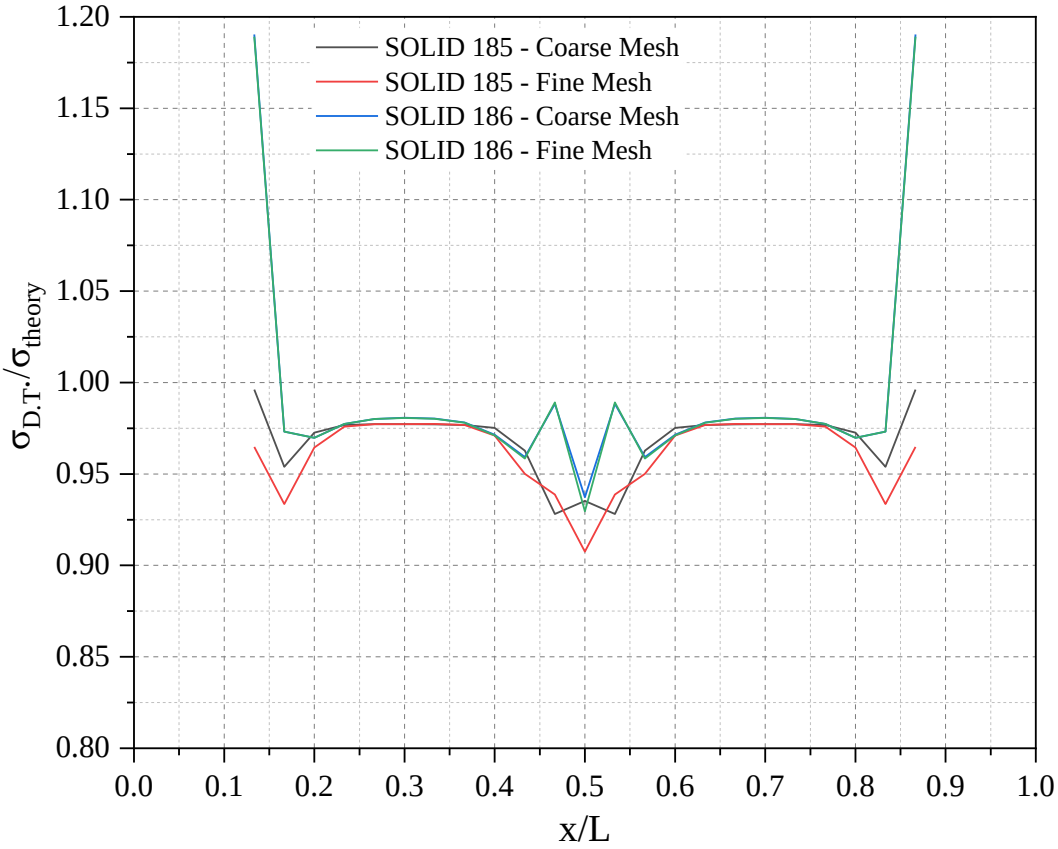


Figure 3.7: Normalized Longitudinal Distribution of Maximum Stress for Solid Element Models

It is made clear from the above, that in the vicinity of the supports as well as the load application point, local effects occur which cause significant deviation from classical

theory. This however was expected, as classical theory refers to an idealized continuum, where all loads and boundary conditions are applied on the centroid of a given cross-section, which is not the case in reality and by extension in this model. That being said, at intervals ranging from 0.2L-0.4L and 0.6L-0.8L, deviation was on average less than 3% which is within acceptable limits. Also, of note was the fact that within those intervals, results corresponding to the same element type converge and that generally mesh refinement plays a negligible role in the outcome. Finally, stress distribution along the web is shown in Figure 3.8 for an indicative solid model case using contour plotting in ANSYS. Note that the localized effects in the vicinity of both the supports as well as the load application point are shown here as well. Also, the reason behind the low stress level is that the load was such that linear elasticity would always be maintained; the reason why will become clear when the SHM system is described later.

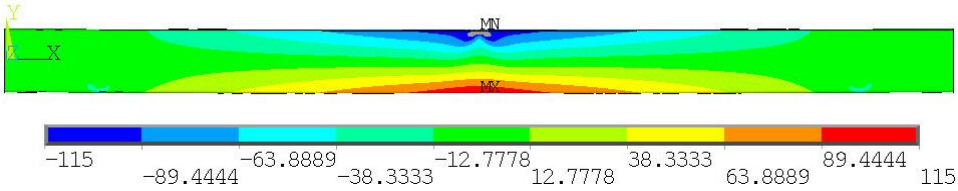


Figure 3.8: Indicative Contour Plot of Web Axial Stress for Solid Element Model

Shell Element Model

In the case of the shell element model, the same two mesh configurations as before were used, with the only difference being in the modeling of the fillet areas. In this case, for those areas an element size of 5 mm was used, as shell elements are by design able to adopt curved shapes and thus two elements of that size are able to model the fillet area while maintaining acceptable levels of deformation. Figure 3.9 below is indicative of the shell element model. Note that the model shown is not three-dimensional but shows the midsurface of the girder. This was chosen to show the fact that when modeling using shell elements, the midsurface is what is essentially being modeled.

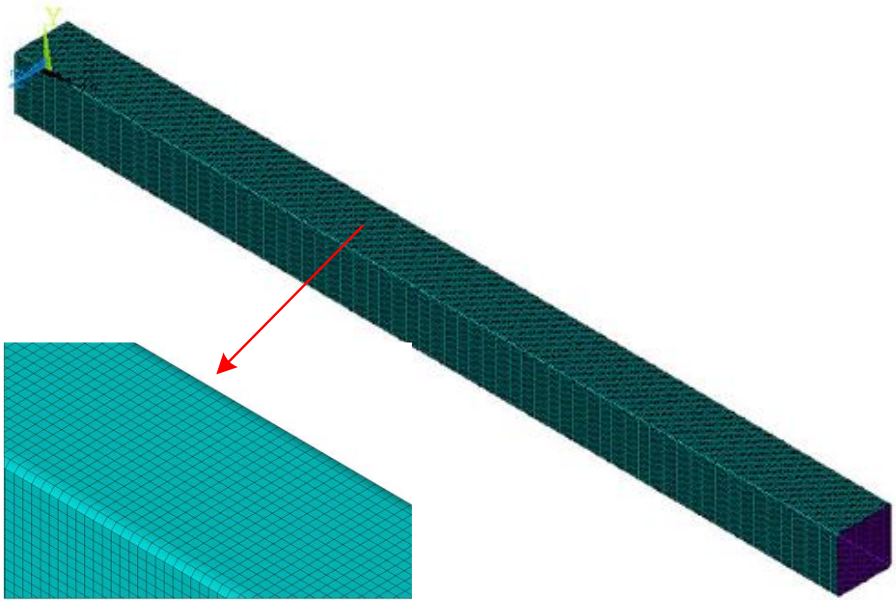


Figure 3.9: Shell Element Model

Regarding the boundary conditions imposed on the model, they naturally must be different than the solid element model developed for this case, because of the fact that shell elements by design have different nodal DOFs. As has been mentioned in Chapter 2.2.3, shell elements also have rotational DOFs, so they must be taken into account as well. Specifically, an equivalent displacement w was applied to simulate the load P, in the same manner as was described for the solid element model. For both supports, vertical displacement u_y was constrained at the bottom flange, along a line at their respective longitudinal positions. To satisfy all other constraints, that is axial and transverse displacement and rotation about the vertical and longitudinal axes, the respective DOFs were constrained at one support. More specifically, they were constrained at nodes located symmetrically at both edges of the bottom flange at this location; the reason behind constraining two nodes being to maintain symmetry in loading and not induce unrealistic rotation and thus shear on the horizontal plane. The shell element model along with applied loads and boundary conditions, as developed in ANSYS is shown in Figure 3.10 below.

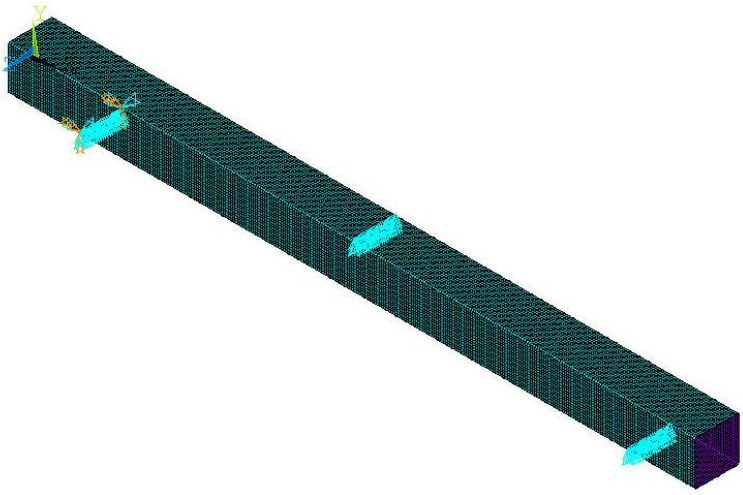


Figure 3.10: Shell Element Model with Applied Loads and Boundary Conditions

As in the case using solid elements, the results obtained from the solution of this model are presented both in the form of a contour plot obtained from ANSYS, as well as in comparison to theoretical estimations for a statically equivalent case. The notation used was the same as the one described earlier, and a further description was omitted for conciseness. What is worth mentioning is that as shell elements use a local coordinate system to calculate stress, it was made sure that the results obtained correspond to the top layer of the element. Following are, first in Figure 3.11 the contour plot showing axial stress distribution along the beam's web and then in Figure 3.12 the normalized longitudinal distribution of maximum axial stress.

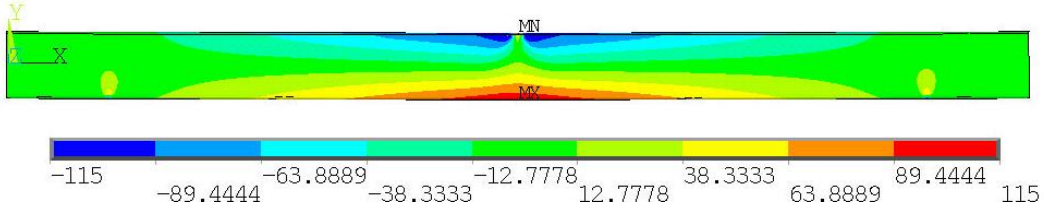


Figure 3.11: Indicative Contour Plot of Web Axial Stress for Shell Element Model

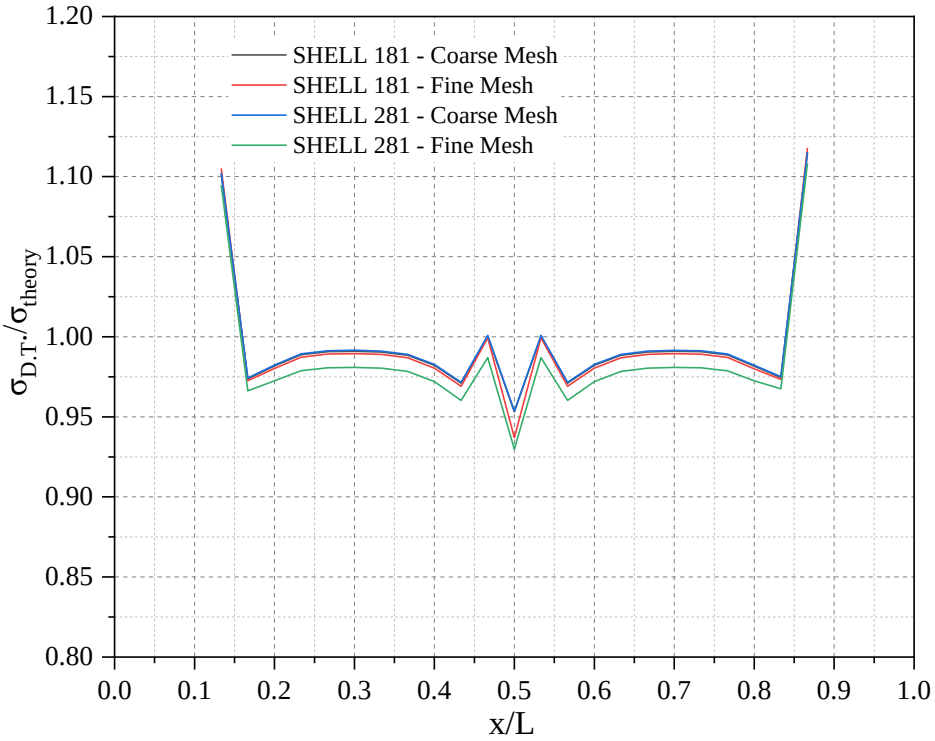


Figure 3.12: Normalized Longitudinal Distribution of Maximum Stress for Shell Element Models

The same general conclusions as in the solid element model can be drawn here as well. Namely, local effects are prevalent near the supports and the load application point, while at the intervals ranging from 0.2L-0.4L and 0.6L-0.8L the deviation from the expected beam theory stresses is lower. Moreover, in this case this deviation is on average smaller than the solid element model, staying well within 2.5 % for most of this interval. This is due to the fact that for shell elements results correspond to the actual outermost fibre of the material, while for solid elements they are calculated at Gauss points, already fixed in the element's geometry.

Final Model Characteristics

At this point, having presented the overall results for the different element types and mesh refinement, the conclusions drawn are summarized and the final model characteristics, that is element type and mesh refinement are chosen. As mentioned before, the two main parameters on which the candidate thin-walled girder models would be judged, with regard to their adequacy in providing the basis for the digital twin to be developed, were speed and accuracy. Table 3.2 presents a comparison of those parameters for all the candidate models that were developed. More specifically, in the first two columns, element type and mesh refinement are presented; the time needed for solution is mentioned in the third column; while in the fourth and fifth, accuracy with regard to theoretical estimations is shown. In the former, the threshold value constantly above which maximum stress remains is showcased for the two regions where local effects have subsided, as mentioned earlier. In the latter, maximum shear stress taken from the FE model at mid-height of the web is presented normalized

to an equivalent theoretical prediction. It should be noted that this theoretical equivalent was calculated using shear flow equations for a box beam in bending.

Table 3.2: Final Comparison of Candidate Thin-Walled Girder Models

Element Type	Mesh Type	Solution Time [s]	$\frac{\sigma_{D.T.}}{\sigma_{formula}}$	$\frac{\tau_{D.T.}}{\tau_{formula}}$
SOLID 185	Coarse	6	> 0.97	0.97
	Fine	11	> 0.965	0.97
SOLID 186	Coarse	12	> 0.97	0.98
	Fine	128	> 0.97	0.98
SHELL 181	Coarse	5	> 0.98	0.97
	Fine	10	> 0.98	0.98
SHELL 281	Coarse	10	> 0.98	0.98
	Fine	43	> 0.97	0.98

In terms of accuracy, it is evident that all candidate models were within acceptable limits; however, shell element models displayed better overall accuracy, something which can also be proven by comparing Figure 3.7 and Figure 3.12. It is also apparent that mesh refinement does not have a significant effect on accuracy, something which admittedly was expected due to simplicity of the problem. Regarding the use of higher order elements, it is clear that it is not an advantageous option as they generally require more time without providing any other significant benefit. Therefore, taking all of the above into account, it was decided to use shell elements, and specifically SHELL 181, to create the final digital twin model. In terms of the mesh used, as for this element type solution time is very small in both mesh instances, the finer mesh was chosen as it would also allow for greater flexibility when damage would be introduced. It is considered worth reminding that the refined mesh in the shell element case refers to an element length of 5 mm for the entire model, with two elements being used to model the fillet areas.

3.4 Strain Monitoring Scheme

3.4.1 Structural Damage Indicators

In the previous sections, the basic characteristics of the FE thin-walled girder model at the core of the digital twin were selected. The next step towards developing the digital twin based SHM system, was the design of an appropriate monitoring scheme. To understand the concept of a monitoring scheme designed to detect damage, the concept of Structural Damage Indicators (SDI) must be first explained.

According to Preisler, Schröder, & Schagerl (2018), the complete understanding of structural behavior is highly beneficial for setting up an SHM system. Primary and secondary load paths must be understood, as well as highly stressed regions of the structure and potential damage mechanisms and their effect on structural behavior. The core concept of SDIs lies in this understanding of structural behavior. Specifically, by comparing the structural behavior of the undamaged and the damaged structure, certain effects can be identified that refer to the presence of damage. Such effects provide very

efficient damage indicators with a clear *True* or *False* statement. Due to their relation to structural behavior, those damage indicators are referred to as structural damage indicators (SDI). A major advantage of those SDIs is that they are sensitive to all kinds of damage which have an influence on the load carrying capacity. In addition, damage size and SDI are directly related. Therefore, the presence of a deviation indicates damage, while the amplitude of that deviation corresponds to the severity of damage. Based on structural analysis, certain predefined thresholds corresponding to certain damage sizes can be defined in order to enable an intrinsic damage assessment. This represents a digital baseline.

In the case of beams in bending, which is the category of problems the case considered in this thesis belongs to, it is known from beam theory that for a linearly elastic material, longitudinal stresses, and therefore strains, are linearly distributed as shown in Figure 3.13 (a) below. In the case considered in this thesis, stress becomes zero at the cross-section's centroid. The line which is connecting all centroids, is referred to as the neutral axis. Therefore, stress and strain along the neutral axis are, according to the definition given earlier, an indicative example of SDIs as any damage would lead to a change in the position of the neutral axis and thus of their value. Also, because of their zero baseline, they are uniquely suited for that task. For that reason, in the SHM system developed in this work, axial strain along the position of the undamaged neutral axis was monitored as an SDI.

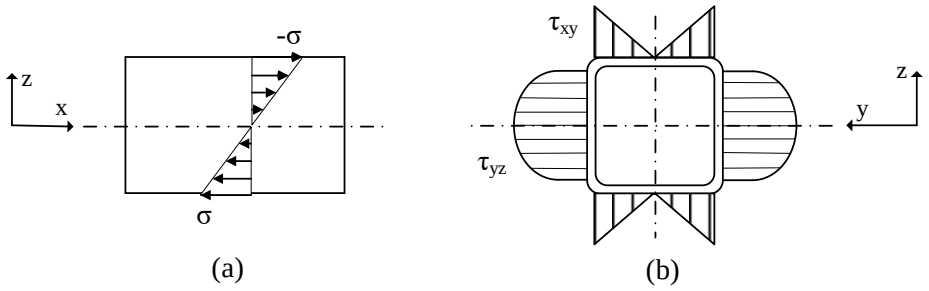


Figure 3.13: (a) Distribution of Longitudinal Stress in Beam Element (b) Shear Flow Distribution in Cross-Section

The other SDI used in the context of this work, is unique to hollow beams in bending. That SDI is shear strain at the centerline of the beam's flanges. It is known from beam theory, that for thin-walled beams with symmetrical cross-sections, shear stress, and therefore strain, is zero along the line where the longitudinal plane of symmetry intersects with the flanges i.e., the flanges' centerline, as shown in Figure 3.13 (b). Therefore, this line constitutes a zero-strain direction for shear and was also chosen to be monitored in the context of the SHM system developed in this thesis.

In conclusion, for the development of the SHM system this work deals with, first a monitoring scheme must be decided. That means the selection of certain physical quantities, which when monitored may -under the right circumstances- allow for the detection of damage to be made possible. To identify them, the concept of SDI was used and through that the following SDIs, suitable for this application were identified.

These were: axial strain monitored along the position of the undamaged neutral axis and shear strain, along the centerline of the flanges.

3.4.2 Development of an Auxiliary Contact Model

Although the SDIs to be used in this case have been conceptually identified, it would be unwise to simply develop an SHM system around monitoring axial and shear strain along their entire corresponding zero-strain directions. The reason why it would be unwise is that it would be solely based on a purely theoretical notion, which ignores local effects whose presence has been observed in the results obtained from the FE models in Chapter 3.3. Therefore, it is important to use those models to make an informed decision which will maximize the system's potential to detect damage.

As mentioned above, the results from the initial FE models indicated that in the vicinity of the supports and the point of load application, local effects disturbed stress distribution causing significant deviations from theoretical estimates. Although intuitively this seemed logical, it was decided that the matter merited further investigation in order to be certain of the model's predictive capacity before beginning the implementation of the SHM system on it.

To that effect, a more detailed FE model was built aiming to realistically simulate the three-point bending test, which is ultimately what the SHM system will be tested on with real-world data. Apart from the beam, which was modeled using ANSYS' SHELL 181 elements with a 5 mm edge length, the rollers used in the actual test were modeled. Only the half in contact with the beam was modeled, to save computational cost and it was modeled as a solid. These half-cylinders were chosen to have radii of 25 mm and a depth of 120 mm across the width of the beam. That depth was chosen to make sure that they are fully in contact with the surface of the beam. They were modelled using SOLID 185 elements, which are the lower-order brick elements provided by ANSYS. Regarding element size, the cylinders were meshed with 10 element divisions across their length. The meshed model of the three-point bending test is shown in Figure 3.14 below along with the areas with contact assignment conditions which will be discussed in the following paragraphs.

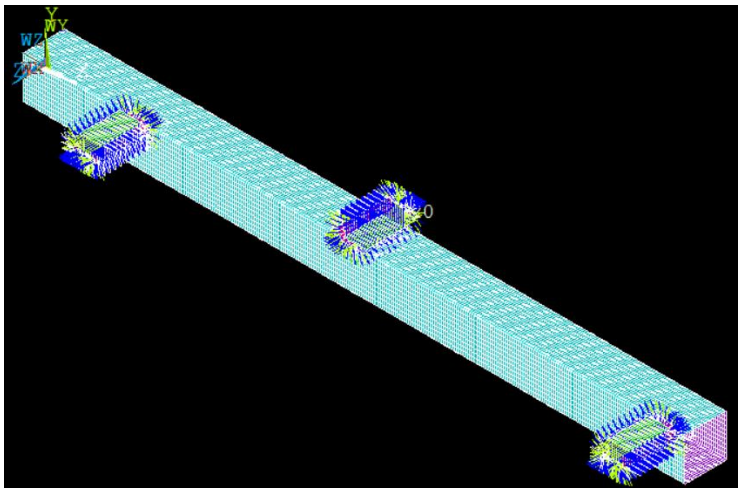


Figure 3.14: Three-Point Bending Test Model with Contact Areas

The application of loads and boundary conditions in this case is essentially done through the rollers using a special class of elements, known as contact elements. These elements facilitate the transfer of nodal forces and displacements from one body to another. Contact elements are surface elements used to model the interaction between two contacting surfaces. In order to create a contact pair, a contact surface and a target surface must first be defined. The target surface is commonly the surface that initiates the contact while the contact surface is the one that "receives" the contact. Contact problems are generally classified as rigid-to-flexible or flexible-to-flexible.

To accurately model the three-point bending test, first it had to be made sure that the rollers behaved essentially as rigid bodies. To simulate that, a pilot node was created at the roller's center of mass to which the roller was "tied". This means that the behavior of this node is emulated by the entire body. Therefore, by applying the necessary boundary conditions on the pilot node the behavior of the entire roller was defined, thus making it a rigid body. Also, to ensure that the stiffness of the rollers was accurately represented, Young's modulus was set at four times greater than that of the girder's material.

After ensuring that the rollers would behave as rigid bodies and have the appropriate stiffness, the contact pair was modeled. As mentioned before, a target surface and a contact surface must be defined. The target surface was set as the cylindrical surface of the rollers while for the contact surface, an area spanning 25 mm on each side of the contact points was allocated on the flanges of the girder model to realistically model the contact area and avoid over-constraining. Regarding the elements used to model these surfaces, three-dimensional contact elements provided by ANSYS were used; namely, TARGE 170 was used for the target surfaces and CONTA 174 for the contact surfaces. The coefficient of friction for the materials used for the contact pairing was taken as 0.2 for both materials, using information from the relevant literature (Chowdhury, Nuruzzaman, & Rahaman, 2014).

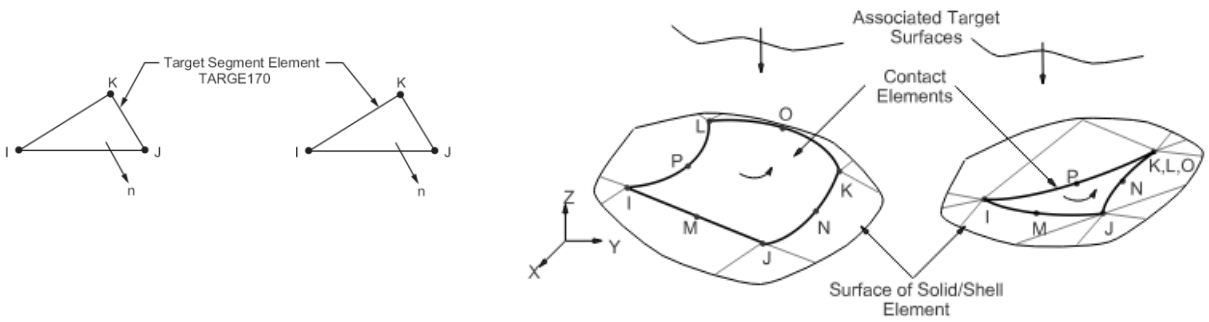


Figure 3.15: ANSYS Contact Elements: TARGE 170 and CONTA 174

After modeling the contact pairing, the final step in the modeling of the problem was to prescribe the necessary boundary conditions and loads. As mentioned before, these are directly applied on the pilot node of each roller. So, for the supports all DOFs of the pilot nodes of the supporting rollers were constrained. For the pushing roller applying the load, all DOFs except vertical displacement were constrained at the pilot node. The

vertical displacement applied to the pilot node was such that a statically equivalent model to the one developed in Chapter 3.3 was produced in order for their comparison to be valid. To do this, it was made sure that both models have the same overall stiffness, as displayed in Table 3.3. Stiffness in this instance was defined as the applied force divided by the displacement it caused, as in the case of a spring. As in the contact model reaction force varies linearly over the applied vertical displacement, the average slope of the resultant distribution, shown in Figure 3.16, was used. Note that the slopes were computed for each time step and as they were slightly different, their average was taken instead.

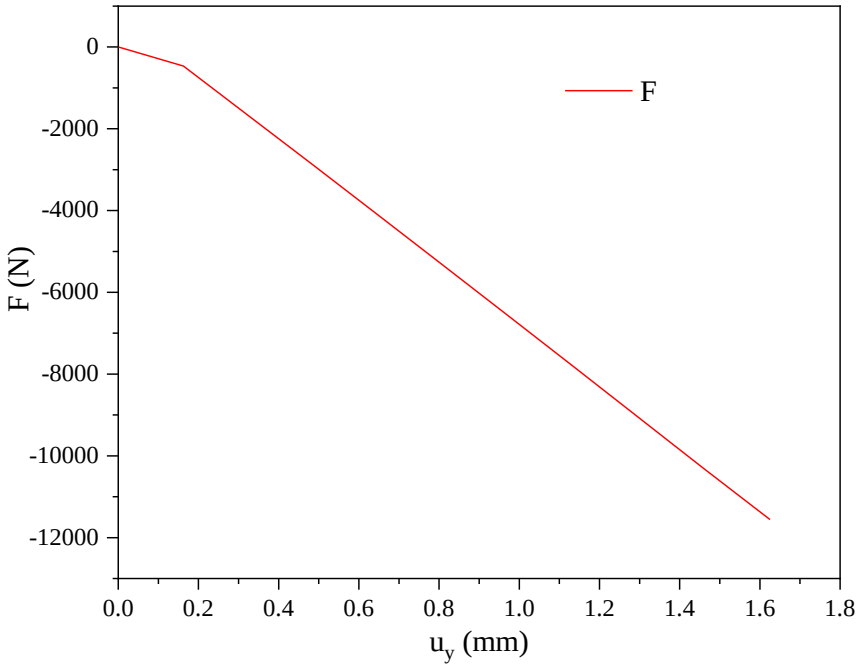


Figure 3.16: Force-Displacement Diagram for Contact Element Model

Table 3.3: Stiffness and Force Equivalency between Contact and Non-Contact Models

	Total Reaction Force [N]	Stiffness [N/mm]
Non-Contact FE Model u_y = -1.5 mm	11449.42	7632.95
Contact FE Model u_y = -1.625 mm	11559.8	7585.27
Error %	0.955	0.625

In the following page the results from the contact analysis are presented. First, a contour plot of the resultant longitudinal stress is shown in Figure 3.17, in true scale with regard to the deformed shape. Then, in Figure 3.18 maximum axial stress measured on the top flange is plotted normalized to the equivalent theoretical estimate, as a function of longitudinal position normalized to overall length. The stresses plotted correspond to the contact case and the equivalent non-contact case and the reason behind it was to determine the extent of the intervals where local effects occurred and then decide on the monitoring scheme configuration.

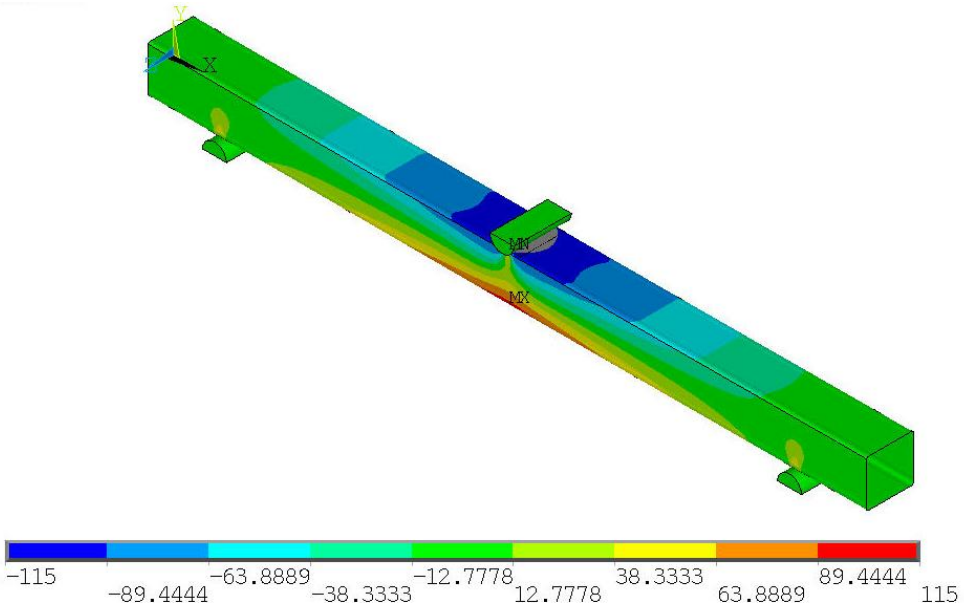


Figure 3.17: Contour Plot of Longitudinal Stress for Contact Element Model

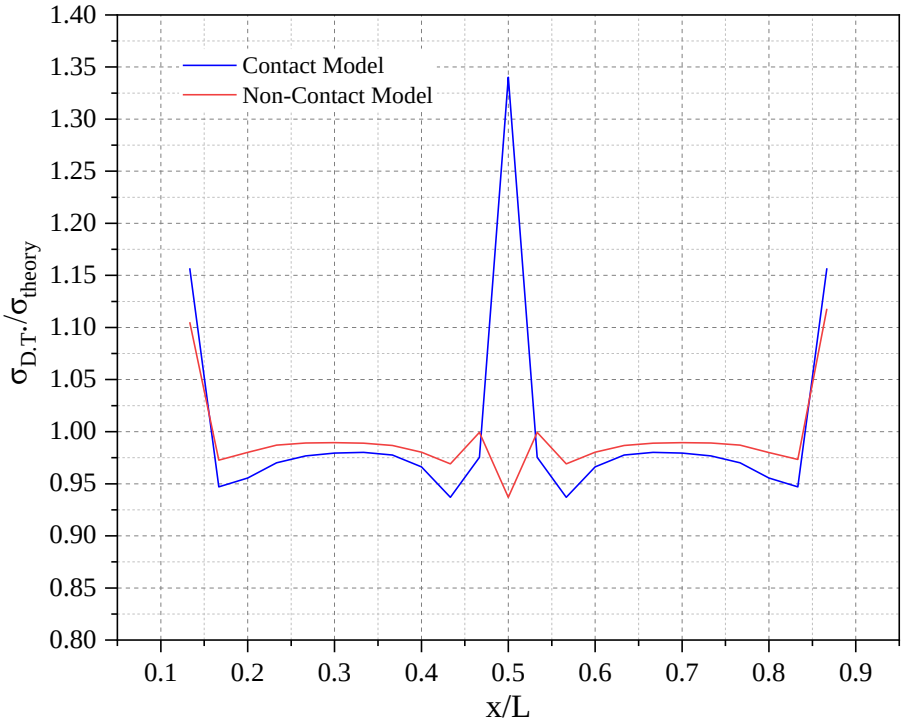


Figure 3.18: Comparison of Normalized Longitudinal Distribution of Maximum Stress Between Contact and Non-Contact Models

It is evident from the comparative plots in Figure 3.18 that the local effects were present at the same locations in the contact model as well, albeit with greater intensity near the load application point. However, this was expected as local stress concentration does appear near the point where pressure is applied by the roller. Also, the model using contact elements clearly showed greater deviation from its estimated theoretical equivalents. It should be noted that those theoretical predictions were obtained by applying the equations of beam theory, as mentioned before, substituting as the point

load the resultant reaction force as it was obtained from FEA. However, in this case the assumptions made in classical beam theory, that is that all loads are applied to the cross-sectional centroid and so on, are not satisfied for the contact model as they are technically not satisfied in the real problem as well. Even so, the deviations that arose within the intervals where local effects have subsided i.e., 0.2L-0.4L and 0.6L-0.8L, were well within 5% and therefore acceptable. Also, when comparing the absolute stress distributions obtained from the two FE models, as shown in Figure 3.19, it became clear that they were virtually identical apart from the area near the point of load application, where the contact model expectedly displays greater stress concentration.

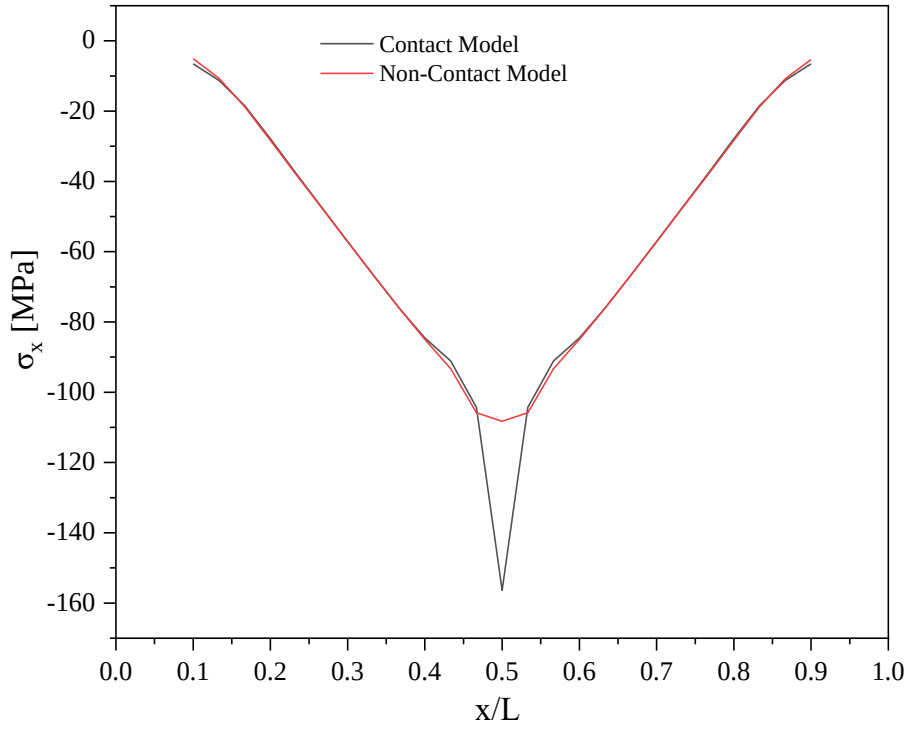


Figure 3.19: Longitudinal Stress Comparison between Contact and Non-Contact FE Models

At this point, there is ample evidence to suggest that the two models are equivalent and therefore that the thin-walled girder FE model described in Chapter 3.3 accurately represents the actual three-point bending case, as this was modeled in detail in the FE model using contact elements detailed here. Thus, this model was deemed suitable for providing the basis for the digital twin model at the core of the SHM system.

3.4.3 Description of the Strain Monitoring Scheme

At this point enough evidence have been gathered to be able to confidently make an informed decision on the strain monitoring scheme to be used in the SHM system. The reason why this decision was included at this section of the thesis, was because this scheme will be realized by the digital twin model through the inclusion of the requisite output commands in the APDL code.

The concept of the Structural Damage Indicator (SDI) was discussed in detail in Chapter 3.4.1. Essentially, SDIs are physical properties which when compared between the damaged and undamaged cases are able to provide an effective indication of

damage. More specifically, when these properties are monitored, any deviation of their value from a previously defined threshold indicates the presence of damage. In the case considered in this thesis, that is of a thin-walled girder subjected to three-point bending, the SDIs recognized as suitable were: axial strain ε_{xx} monitored along the position of the undamaged neutral axis, which is essentially the longitudinal axis of symmetry of the webs; and also, in-plane shear strain ε_{xy} monitored along zero-shear direction on the flanges, that is along their longitudinal axis of symmetry.

These SDIs were selected because of their zero baseline; meaning that zero strain values are expected from them according to the fundamental principles of structural analysis, as they are applied here. In addition to that, a fundamental understanding of the structural behavior of the real-world problem, as well as initial results obtained from FEA, indicated that these theoretically expected zero values are not in fact zero everywhere along the aforementioned directions because of local stress concentrations that arise in the vicinity of the supports and load application point. Therefore, to be able to make an informed decision on the extent of the monitored regions in the SHM system, which are vital to its success, a complementary, more detailed "one-off" model was created, which was the focus of Chapter 3.4.2. Indeed, this model further reinforced the conclusions drawn before and led to a final decision about the strain monitoring scheme to be implemented on the digital twin, which constitutes one of the key features of the SHM system.

Following in Figure 3.20 is a schematic of the monitoring scheme which will be included in the digital twin to be used in the development of the SHM system. In total, eight intervals along those directions, from now on known as paths, will be used to monitor the SDIs. These paths correspond to the sections of the zero-strain directions on the girder spanning the 0.2L-0.4L and 0.6L-0.8L intervals, on which local effects were found to be negligible. Two paths are located on each face, one on each side of the load application point; these sides shall be known from now on as spans. In total, two paths are located on each face and four on each span: of those paths, four monitor axial strain ε_{xx} and the other four in-plane shear strain ε_{xy} along the web and flange paths respectively.

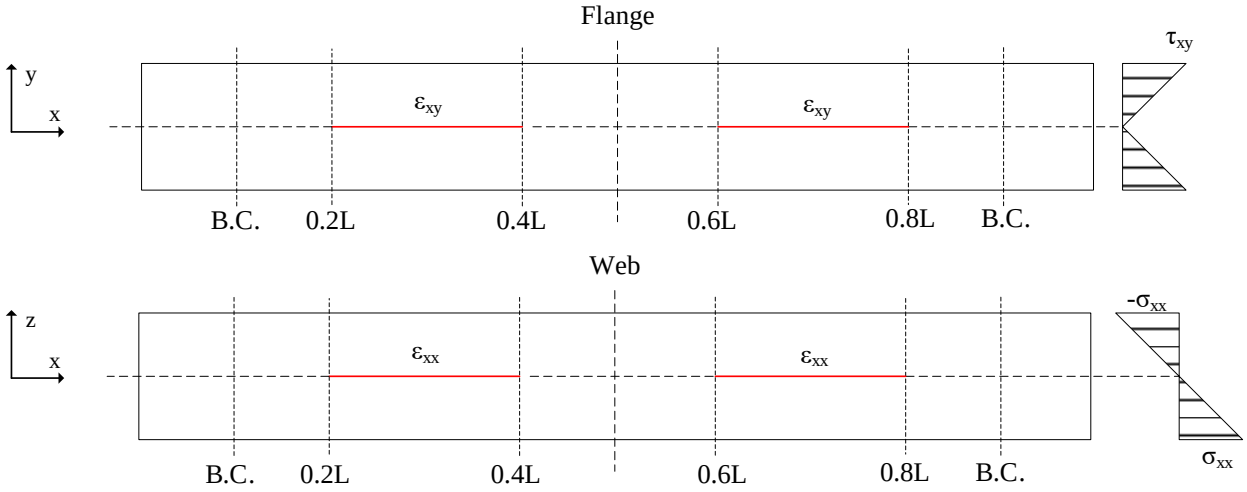


Figure 3.20: Selected Locations for Strain Monitoring

3.5 Introduction of Damage to the Digital Twin

3.5.1 Introduction of a Strain Field Disturber (SFD)

The final step to complete the digital twin is the introduction of some damage scenario, which will be used to test the SHM system. As mentioned before, SDIs are by definition sensitive to all kinds of damage. Therefore, in this case it is not of primary importance to introduce some particular indicative type of damage e.g., thickness reduction representing corrosion; rather, it is more useful to introduce some factor which may lead to a disturbance in the stress and strain field. For this reason, the concept of a Strain Field Disturber (SFD) is introduced. This refers to an artificially introduced discontinuity in the geometry of the model, which leads to a redistribution of stress and strain and thus to a change in the location of the neutral axis and eventually the value of the monitored SDIs. In the present work, this is achieved by the introduction of a circular hole on the surface of the girder. During the course of this thesis, the terms Strain Field Disturber (or SDI), hole and defect are used interchangeably.

Specifically, the model developed in Chapter 3.3 was modified in such a way, as to enable the opening of a hole on either the webs and flanges of the girder. The areas on the faces of the girder where the hole may be opened were defined taking into account two major parameters. First, that the hole should be located, in terms of its longitudinal location, within the limits posed by the onset of local phenomena, which were discussed in the previous section and shown schematically in Figure 3.20. And secondly, it was constrained by factors related to the meshing of the model. Because this reason is highly significant as it relates to the accuracy of the model, it is hereby discussed in greater detail.

It is well known from FEA, that mesh integrity must be maintained at the vicinity of a geometric discontinuity, in order for results to not become contaminated and thus their accuracy to be compromised. Therefore, it is common to divide the area in the vicinity of the hole into appropriate sub-areas, which will then be able to be discretized in such a way that the mesh remains structured. So, this approach was followed when re-structuring the thin-walled girder FE model to accommodate the feature of the SFD, which in this case was chosen to be a hole as mentioned before.

To achieve this in this work, two approaches were initially considered. In the first, the existing model would be first built and afterwards the hole was to be opened on a defined location. However, this approach proved to be very inflexible and was quickly abandoned. The second approach, which was ultimately used for the scope of this work, essentially treated the hole as a feature of the model and integrated its creation to the modeling of the midsurface itself. This necessitated that the appropriate area division around the hole was decided in advance and integrated to the section of the APDL code that deals with the modeling of the beam. An indicative example of the area subdivision that was adopted, corresponding to a hole located on the top flange of the girder is presented in Figure *3.21* below. Also, the trade-off between the extent of the possible

locations of the hole in this present model and at the same its ability to effectively maintain mesh integrity and the relevant decisions made going forward, are discussed.

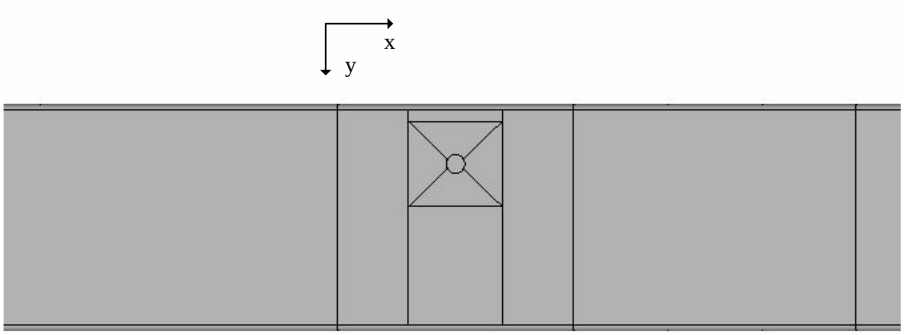


Figure 3.21: Detail Showing the Area Subdivision around the Hole

As was mentioned, this detail is located on the top flange of the girder, with the longitudinal direction denoted as x and the transverse as y, as shown in the relevant coordinate system. As one observes, the area subdivision around the hole was effectively done in three stages. In the first stage, the area in the immediate vicinity of the hole was addressed. Specifically, it was enclosed in a square which was itself divided into four triangular pieces. The dimensions of this square were fixed beforehand and were equal to four element lengths. This unit of subdivision was preferred at this stage as the author believes that it enables a more practical understanding of the matter, as opposed to a conventional unit of measurement, i.e. millimeters.

In the second stage, the transverse position of the hole was addressed. To that end two rectangles were used, each spanning lengthwise four element lengths. They act as a "guide" which enables the transverse movement of the inner square and ostensibly of the hole itself, without jeopardizing the integrity of the mesh. This was made possible by making their transverse dimension a variable, which is calculated internally in the APDL code, by factoring the transverse coordinate of the circular hole's center as well as the fixed dimension of the inner square.

However, in achieving that some limitations arose regarding the transverse extent on the face where it can be effectively located. Namely, by the design of the square surrounding it, it may be no closer than four element lengths from the edge of the face. Also, to maintain this three-stage subdivision, a minimum distance of one element length from the edge of the face is dictated. Thus, the location of the hole was constrained widthwise to effectively half the face. In the scope of this work, this limitation was considered acceptable and was taken into consideration when the model was fully parametrized, which will be discussed in the following section.

Finally, in the third stage two rectangles were added on either side of the area subdivided in the last two stages. Their role was to maintain the integrity of the mesh while enabling the hole to be located in different longitudinal positions. Each of these spans widthwise a length of twenty elements, i.e. it covers the entire width of the face, and lengthwise a length of eight elements. Thus, the total area of the face which was

subdivided for the structuring of the mesh around the hole is a rectangle with sides of twenty element lengths each. It is important at this point to note that although it would seem to be a square, this is not the case technically. The reason is that even though the element length was set to a constant 5 mm, the presence of the fillets leads to a smaller width of the faces, which leads to a small deformation of the element edge lengths, which however was deemed acceptable. This was another reason why the element length was preferred as a unit of subdivision. In Figure 3.22 the same detail is shown after it has been meshed, showcasing the effects of the subdivision on its structuring.

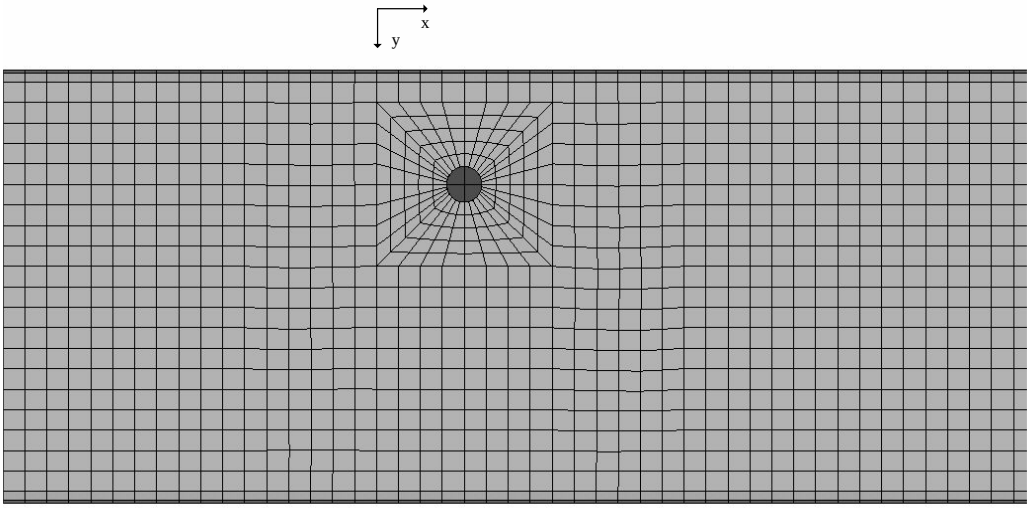


Figure 3.22: Detail Showing the Mesh Structuring around the Hole

It is evident that the area subdivision selected to ensure that the mesh remained structured around the hole was successful. The use of element length as a unit of subdivision for the sides of the sub-areas is further elucidated here, in its role in maintaining mesh integrity. Specifically, it allowed for opposing sides to have the same number of element subdivisions when meshed, thus creating the result shown above. Furthermore, the use of "flanking" individual rectangles to contain the mesh refinement at the immediate vicinity of the hole is displayed.

It is worth noting at this point however, that an additional limitation arose regarding again the possible locations where the defect may be placed while at the same time maintaining the structure of the mesh. Particularly, apart from the limits imposed to widthwise location that were discussed previously, to be able to maintain this pattern the center of the hole must be located at integer multiples of element edge length for both coordinates. However, the very fine mesh selected compensates for this and therefore it was not found to be limiting to the application considered here.

In addition to the limitations discussed in this section, it must be brought into attention at this point that the possible locations of the SFD are also bound by the same considerations regarding local effects as the strain monitoring scheme, detailed in Chapter 3.4.3. This means that these possible locations are also longitudinally constrained. The regions on the webs and flanges where ultimately the defects can be introduced, are shown in Figure 3.23 below.

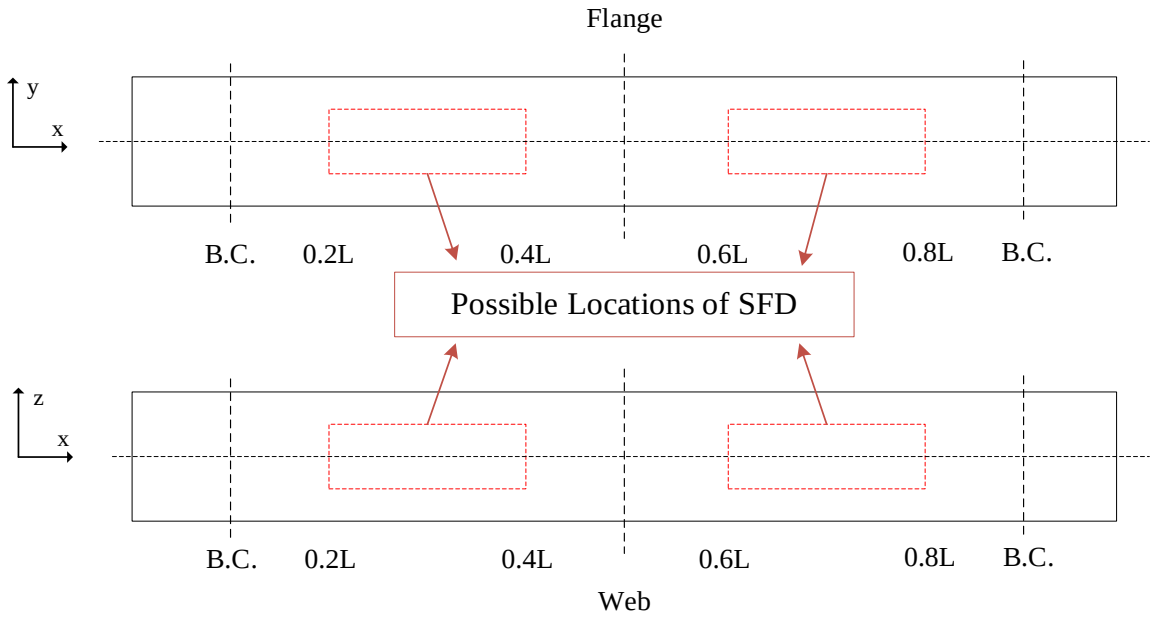


Figure 3.23: Areas of Possible Defect Locations

3.5.2 Full Model Parametrization

It has been alluded to various times earlier that the thin-walled girder model at the basis of the digital twin has been parametrically developed using the ANSYS Mechanical APDL programming language. In this section, the procedure used to develop the parametric model is explained in detail, to provide the reader with a now complete overview of the digital twin model before dealing with its predictive capabilities, and the methods employed to create the SHM system discussed in the following sections.

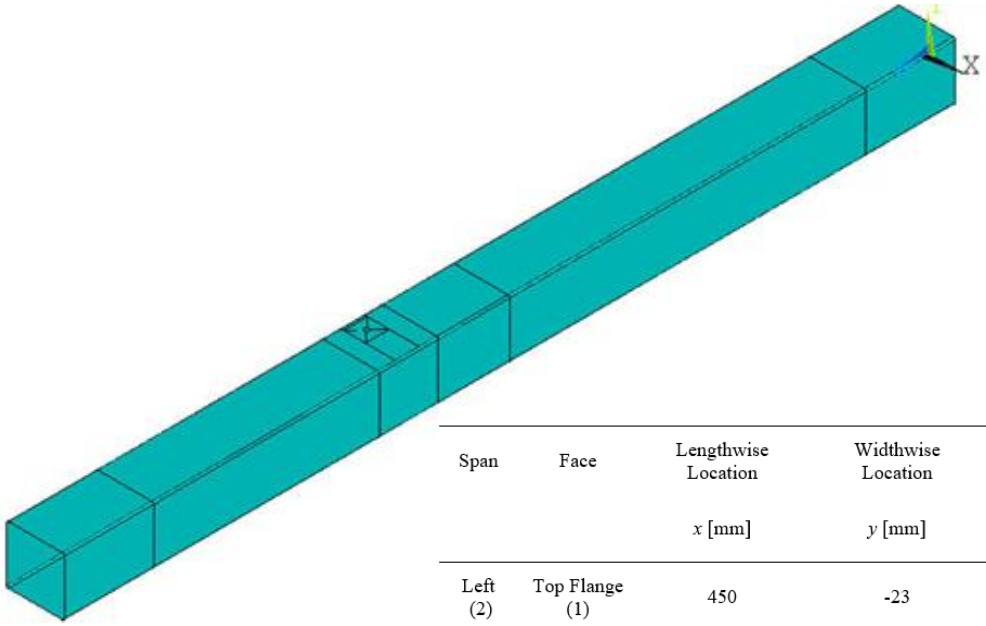
In Chapter 3.3, the decision to adopt a shell element formulation was explained. It was also detailed how in such models, the midsurface of the three-dimensional solid is modeled and the appropriate thickness is provided as an element property. Also, in the previous section, it was mentioned that the inclusion of the SFD to the model was done during the modeling of the midsurface. The main reason behind that, was that the parametrization of the model was "tailored" around this specific task. In particular, it was parametrized in such a way that it could be constructed for any given location of the hole within the defined areas, as specified in Figure 3.23, by initially setting the values of certain parameters, introduced as variables in the code, related exclusively to the defect.

These parameters, which will be interchangeably referred to as damage identifiers during the course of this thesis, constitute the identity of the SFD and are an integral part not only of the digital twin, but of the SHM system in general. They are the principle contributor to the digital twin's versatility and ability to be used repeatedly, in various applications with virtually no additional computational cost. It is through them, that the development of the damage detection schemes presented later is made possible. These parameters are the following:

- The span of the girder, meaning the side to the right or the left of the load application point, on which the hole is located. In the APDL code, the identifier uses 1 to indicate right span and 2 to indicate left span.
- The face of the girder on which the defect is located. Each of the four faces was given an identifying number, starting from 1 for the top flange and assigning clockwise up to 4 for the left web. It should be noted that the right web refers to the one on the positive side of the coordinate system etc.
- The longitudinal (x) distance of the circular hole's center measured from the nearest support.
- The transverse (y) positions of the circular hole's center measured from the centerline of each face.
- The radius r of the circular hole.

In the APDL code, these parameters are set at the top of the script. Then, through a series of if statements the code is able to proceed with a direct modeling of the girder, including modeling the defect in the manner described in Chapter 3.5.1. Meshing is achieved by explicitly setting the number of divisions for each line on the midsurface, to ensure it remains structured. Material properties and element selection remain the same, as previously defined in Chapter 3.3, as do the imposed boundary conditions and loads.

In Figure 3.24 the midsurface model of the digital twin is shown, including a defect located on the top flange, as in the case detailed in the previous section. The location of the hole, expressed in the terms of the parameters described above, is also provided. Note that the girder retains its initial four-piece compartmentalization, done to allow the boundary conditions and displacement load to be applied across a line spanning the entire width of the flanges, thus realistically simulating the actual effect of the rollers on the three-point bending test.



Span	Face	Lengthwise Location	Widthwise Location	Radius
		x [mm]	y [mm]	r [mm]
Left (2)	Top Flange (1)	450	-23	4

Figure 3.24: Midsurface Model of the Digital Twin

At this stage, with the complete digital twin at our disposal, its predictive capabilities were put to the test, by examining whether the selected SDIs were indeed capable of damage detection.

3.5.3 Assessment of Damage Detection Capacity

In the previous two sections, the digital twin model was completed by first introducing damage in the form of a Strain Field Disturber (SFD), whose characteristics and introduction to the model were detailed in Chapter 3.5.1 and then by fully parametrizing the model, enabling it to be constructed each time simply by using as input the damage identifiers described in Chapter 3.5.2. At this point, the digital twin was assessed with regard to its capacity to detect damage. Specifically, for some pre-defined damage cases the structural response of the girder was examined by monitoring the SDIs identified earlier, along their corresponding monitoring paths detailed in Chapter 3.4.3. This was done to test the capacity of this monitoring scheme to detect damage as well as whether the resultant response was what was expected according to the author's understanding of the structural behavior of this particular case.

The characteristics of the indicative damaged case considered first are presented in Table 3.4, and the corresponding model is shown in Figure 3.25, taken from the ANSYS environment.

Table 3.4: Indicative Damage Case No.1 – Defect Located on Web

Span	Face	Lengthwise Location x [mm]	Widthwise Location y [mm]	Radius r [mm]
Right (1)	Left Web (4)	300	14	4

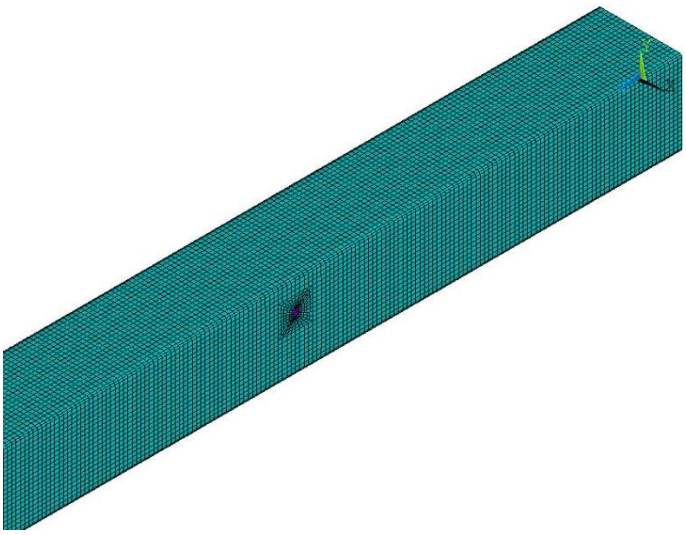


Figure 3.25: Indicative Damaged Case No.1

In the following page, the strains monitored along all possible paths are shown. The results from the indicative damaged case described above are plotted compared to those

corresponding to the pristine model. Each graph corresponds to a separate face and each section corresponds to one of the two respective paths located on that face, one on the right span and one on the left span. These graphs (Figures 3.26-3.29) aim to showcase at first, the ability of the monitoring scheme to detect damage and secondly to illustrate the local nature of damage, which is expected according to the author's theoretical understanding of the problem.

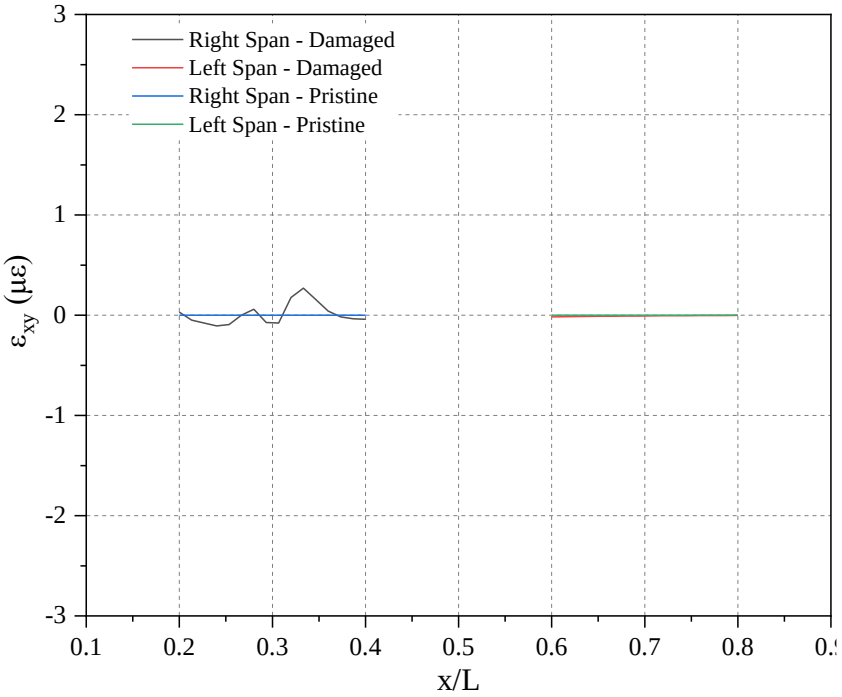


Figure 3.26: Top Flange In-Plane Shear Strain Monitoring Results – Damaged Case No. 1

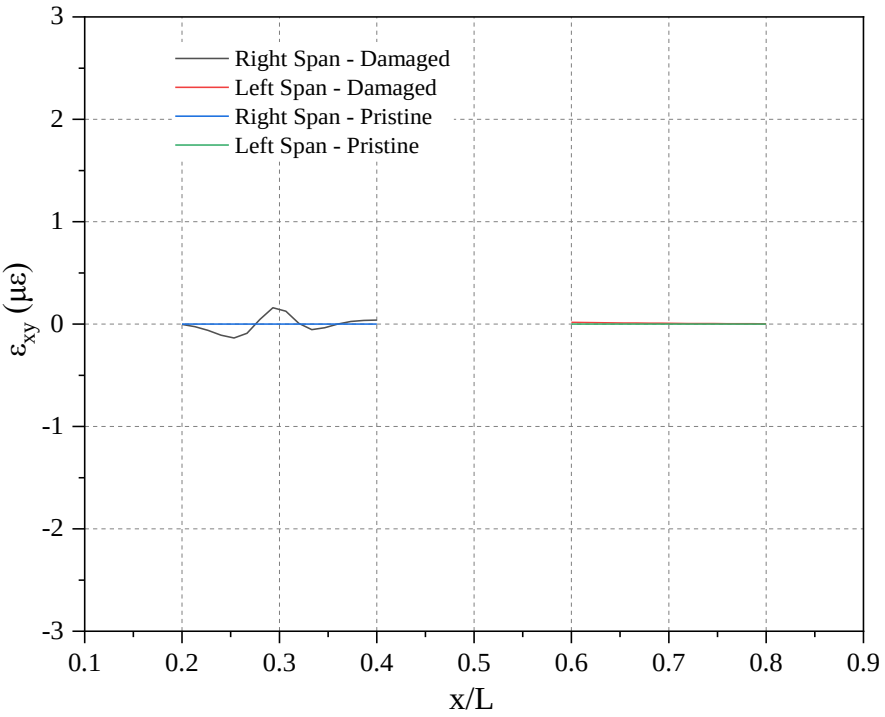


Figure 3.27: Bottom Flange In-Plane Shear Strain Monitoring Results – Damaged Case No. 1

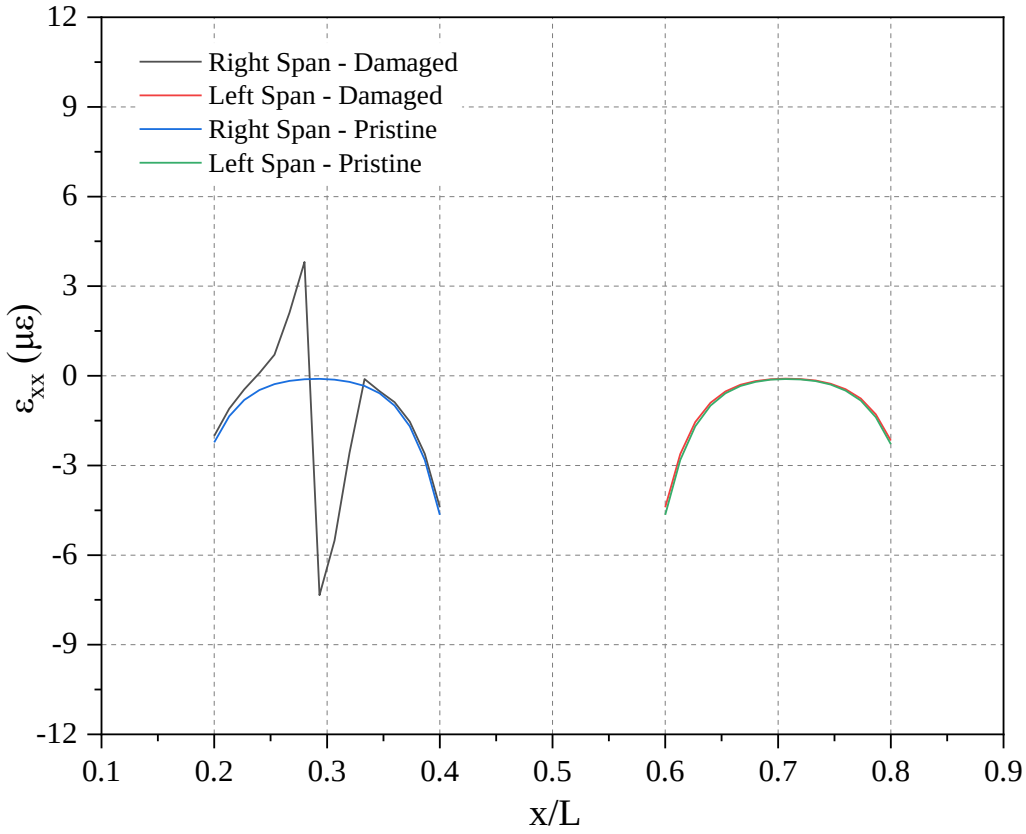


Figure 3.28: Left Web Axial Strain Monitoring Results – Damaged Case No. 1

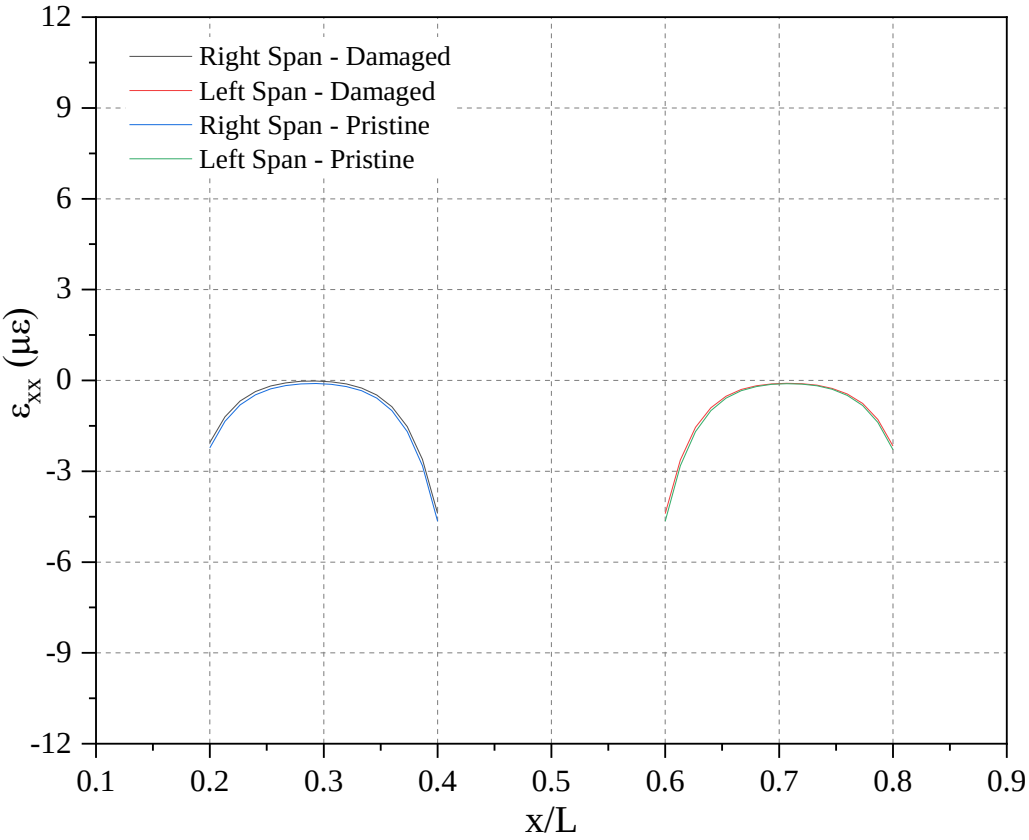


Figure 3.29: Right Web Axial Strain Monitoring Results – Damaged Case No. 1

It is evident from Figures 3.26-3.29 presented above that the local nature of damage, expected from the theoretical understanding of the problem was substantiated by the numerical results obtained by the digital twin. Specifically, Figure 3.28 clearly illustrates that the deviation of the strain field caused by the presence of damage was confined to the path in the immediate vicinity of the defect. Even more so, it was confined to an area spanning 0.1L on both sides of the hole's center. Therefore, the local nature of damage is strongly supported by the digital twin results. Furthermore, it is also shown than in all other paths, the monitored SDIs are practically identical between the pristine and the damaged case. It should be noted that although Figure 3.26 and Figure 3.27 indicate some deviation from the pristine case around the defect location, its magnitude is negligible and as such is not considered capable of realistically detecting damage. To highlight these even more, Figure 3.30 is provided showing contour plots of top flange in-plane shear strain (Figure 3.30 (a)) and axial strain (Figure 3.30 (b)) in the immediate vicinity of the hole.

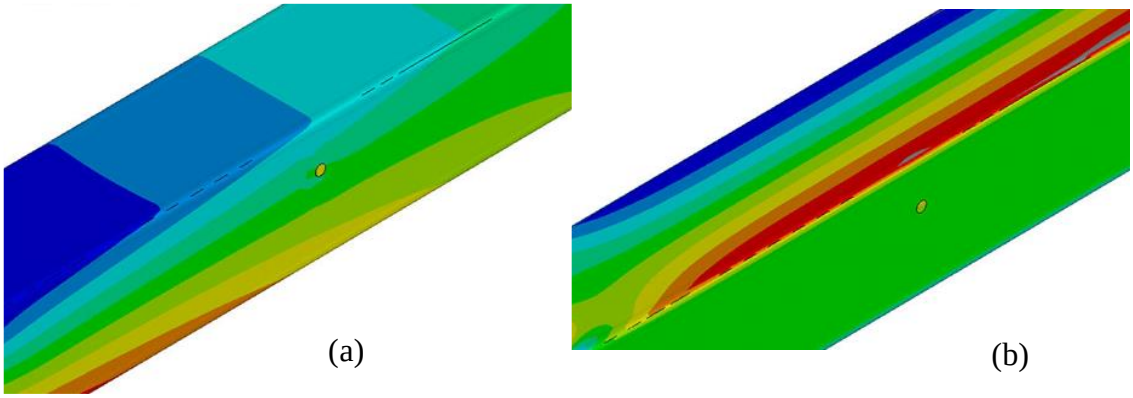


Figure 3.30: Contour Plot of (a) Axial Strain and (b) Top Flange In-Plane Shear Strain for Indicative Damaged Case No.1 in the Vicinity of the Hole

In the case of axial strain, it is shown in the contour plot that indeed the strain field disturbance caused by the hole is very localized and small in magnitude, as indicated by the relevant graphs presented earlier. In the case of in-plane shear strain on the top flange, it is illustrated here too that the presence of the hole on the web has no effect on that strain component.

To provide more evidence supporting these conclusions, an additional damaged case is presented in the same manner as before. This time the SFD is located on a flange, in order to test the sensitivity of the shear strain SDI. The characteristics of this damaged case are shown more illustratively in Table 3.5 and Figure 3.31.

Table 3.5: Indicative Damage Case No. 2 – Defect Located on Flange

Span	Face	Lengthwise Location x [mm]	Widthwise Location y [mm]	Radius r [mm]
Left (2)	Top Flange (1)	400	23	4

Figure 3.31: Indicative Damaged Case No.2

For this case, not all the graphs are shown as in the previous case. As the locality of damage has been clearly shown, it was decided to show only the graph corresponding to the top flange monitoring paths, shown in Figure 3.32 and another one displaying the monitoring paths for an indicative web as well, in this case the left web showcased in Figure 3.33. The first graph is used to show the effectiveness of the in-plane shear SDI while the second is used to further reinforce the conclusion that the presence of a defect affects strain distribution in an absolutely local manner.

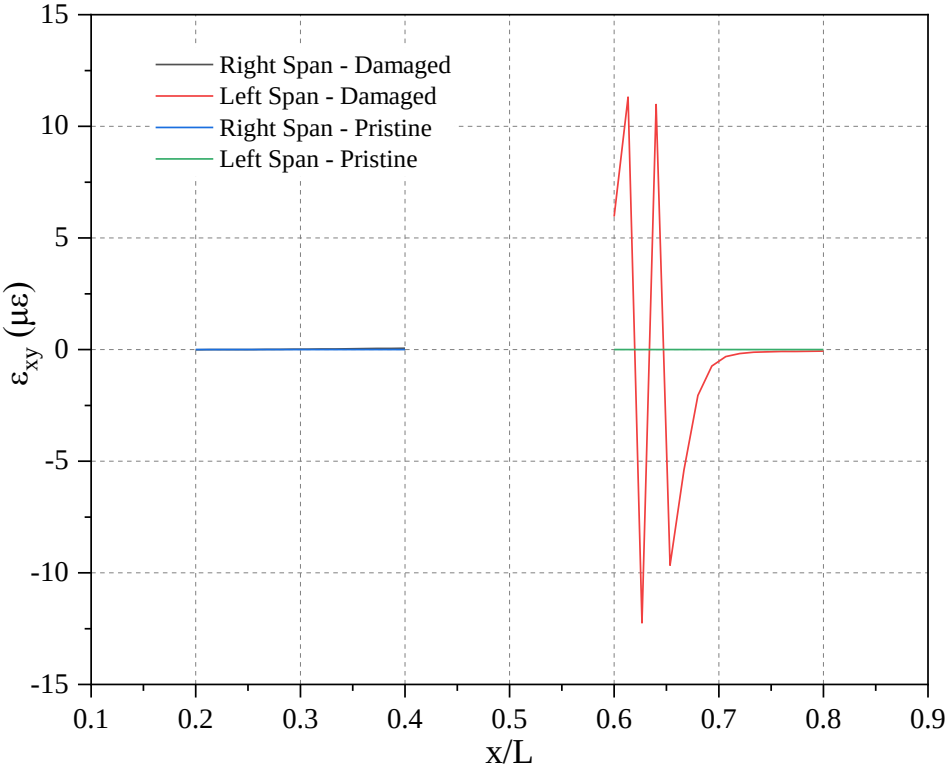


Figure 3.32: Top Flange In-Plane Shear Strain Monitoring Results – Damaged Case No. 2

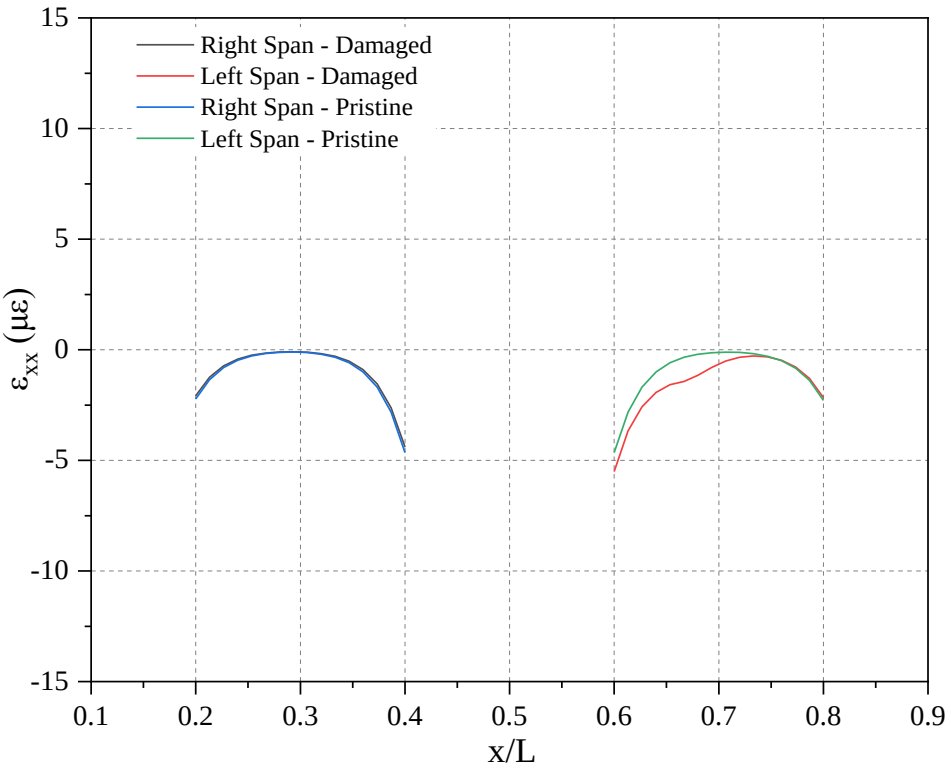


Figure 3.33: Left Web Axial Strain Monitoring Results - Damaged Case No.2

In Figure 3.32 it is again clear that the presence of damage effects a region spanning 0.1L, on both sides of the hole location. Also, it is interesting to note that in this case the deviation of strain due to damage was greater than in the web case presented before. In any case, the local nature of damage is further substantiated here as well. Also, in Figure 3.33 the right span distribution showed some deviation around the damage location. The fact that it was present on the left web paths is something that was expected as the defect was indeed located on the left side of the top flange. It should be noted however, that the magnitude of deviation is significantly smaller than the deviation observed in the immediate vicinity of the SFD. This adds more credence to the conclusion about the local effect of damage on structural behavior. Finally, in Figure 3.34 contour plots of axial (a) and in-plane shear strain (for the top flange) (b), are presented as before to better showcase the effect of the hole on the strain fields corresponding to the two strain components.

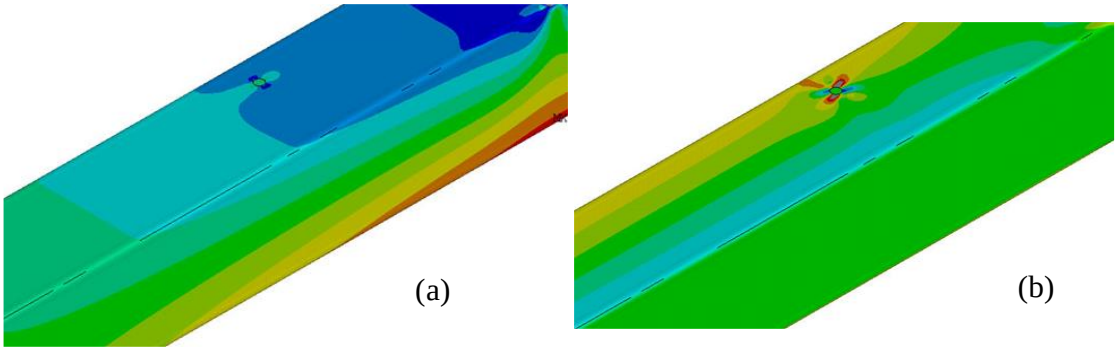


Figure 3.34: Contour Plot of (a) Axial Strain and (b) Top Flange In-Plane Shear Strain for Indicative Damaged Case No.1 in the Vicinity of the Hole

Again, the results shown in the graph are supported by the contour plots as well. It is worth noting that the greater magnitude of the disturbance caused by the hole in the in-plane shear strain field is clearly shown. Also, it ought to be mentioned that it does affect the axial strain field on the top flange as well, which was expected but not illustrated by the graphs as this property is not one being monitored.

In conclusion, these results essentially reinforced the decision made to monitor these particular SDIs along those specific paths. They also provided important insights regarding the local effects of possible damage, which proved to be invaluable in the development of the SHM system, as well as the design of the laboratory tests further down the road. Finally, they proved conclusively that a solid digital twin framework has been built, which can become the foundation of a robust SHM system.

3.6 Genetic Algorithm-Based Approach for Damage Identification

In the previous sections, a FE model of a thin-walled girder subjected to three-point bending was developed. A strain monitoring scheme was proposed based on the concept of SDI. Damage was introduced in the form of what was defined as an SFD and the capabilities of the monitoring scheme to detect it were investigated and verified. The final FE model, featuring all of the above, constitutes the digital twin on which the SHM system will be based.

To realize the Structural Health Monitoring system, a method must be introduced to effectively detect damage. In this Chapter, the first of two such methods developed for this thesis is introduced. This method takes advantage of the basic ideas of optimization and utilizes one of its most versatile and effective techniques, that is Genetic Algorithms (GA). The basic concepts of both optimization and GAs were detailed earlier in Chapter 2.3 and thus will not be repeated here. This Chapter will deal only with explaining the parameters involved in this method and with a description of its implementation.

First and foremost, the assumption was made that a collection of strain data gathered from sensors located on the paths used for strain monitoring, as they were specified in Chapter 3.4.3, was readily available. This collection of data may correspond to actual data measured on the physical asset, or data generated by digital twin-based simulations. With that knowledge in hand, the problem of damage detection was effectively treated as a minimization problem, where the function to be minimized was an error function between actual strain data, and strain data obtained from the digital twin for various instances of damage. This, being essentially an optimization problem can be defined in appropriate terms in the following way.

First, let us consider a design vector denoted as $\boldsymbol{x}$ which contains the damage identifiers used to parametrize the digital twin model, as mentioned in Chapter 3.5.2. This can be defined as follows:

$$x = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5]^T \tag{3.4}$$

Where, $x_1 = \{1,2\}$ is a discrete variable that represents the span where the defect is located; $x_2 = \{1,2,3,4\}$ is also discrete and acts as the identifier of the face on which the hole is located; x_3 is the longitudinal distance from the nearest support and although commonly a continuous variable in this problem it was considered discrete, to satisfy the condition that the defect may be positioned at one element length intervals. Therefore, it was defined as $x_3 = \{0,5,\ldots,300\}$. The transverse distance from the face's longitudinal axis of symmetry is represented as x_4 and was also assumed to be discrete to satisfy the same condition as x_3; it was defined as $x_4 = \{-22.91,-18.328,\ldots,22.91\}$. It should be noted that the values are such because of the effect of the fillets on the faces, which cause a small, but acceptable, deformation of element size. Finally, x_5 represents the circular hole's radius and was assumed to be a discrete variable as well, defined as $x_5 = \{3,3.2,\ldots,5\}$.

The strain vector obtained from the asset monitoring, which shall be denoted as ε_{act}, can now be defined as follows:

$$\varepsilon_{act} = \begin{bmatrix} \varepsilon_{act}^{(1)} \\ \varepsilon_{act}^{(2)} \\ \vdots \\ \varepsilon_{act}^{(n)} \end{bmatrix} \tag{3.5}$$

Where $\varepsilon_{act}^{(i)}$ is the strain vector monitored for the corresponding SDI of the i-th path and n is the number of paths monitored, which in the context of this work, as discussed in 3.4.3, is equal to $n=8$. The strain vector monitored in each path can be in turn defined as:

$$\varepsilon_{act}^{(i)} = \begin{bmatrix} \varepsilon_{act,1} \\ \varepsilon_{act,2} \\ \vdots \\ \varepsilon_{act,k} \end{bmatrix} \tag{3.6}$$

Where ε_{act} refers to the strain component monitored on the i-th path and k is the number of sensors located on that particular path.

It was mentioned earlier that the design vector x was defined in such a way that it contains essentially the input parameters for the APDL code used to build the digital twin. Also, as it describes a certain damaged case, there exists a similar strain vector to the one described above which shall be denoted as $\varepsilon_{vir}(x)$ and defined in the following way:

$$\varepsilon_{vir}(x) = \begin{bmatrix} \varepsilon_{vir}^{(1)}(x) \\ \varepsilon_{vir}^{(2)}(x) \\ \vdots \\ \varepsilon_{vir}^{(n)}(x) \end{bmatrix} \tag{3.7}$$

Where $\varepsilon_{vir}^{(i)}$ is the strain vector monitored for the corresponding SDI of the i-th path and ***n*** is the number of paths monitored, which in this case also, is equal to ***n=8***. The strain vector monitored in each path can be in turn defined as:

$$\varepsilon_{vir}^{(i)}(x) = \begin{bmatrix} \varepsilon_{vir,1}(x) \\ \varepsilon_{vir,2}(x) \\ \vdots \\ \varepsilon_{vir,k}(x) \end{bmatrix} \tag{3.8}$$

Therefore, it is now possible to evaluate the "suitability" of any solution ***x*** to correctly identify damage by measuring a distance between its corresponding strain vector and that representing actual damage by means of evaluating an expression such as $\|\boldsymbol{\varepsilon_{act}} - \boldsymbol{\varepsilon_{vir}}\|$. It is made clear, that if this metric were to be minimized, that is essentially to be driven to zero, the $\boldsymbol{\varepsilon_{vir}(x)}$ for which that would be satisfied, would theoretically correspond to a design vector ***x*** containing the exact same parameters of damage as in the actual case. To make this even more clear, Figure 3.35 is provided below.

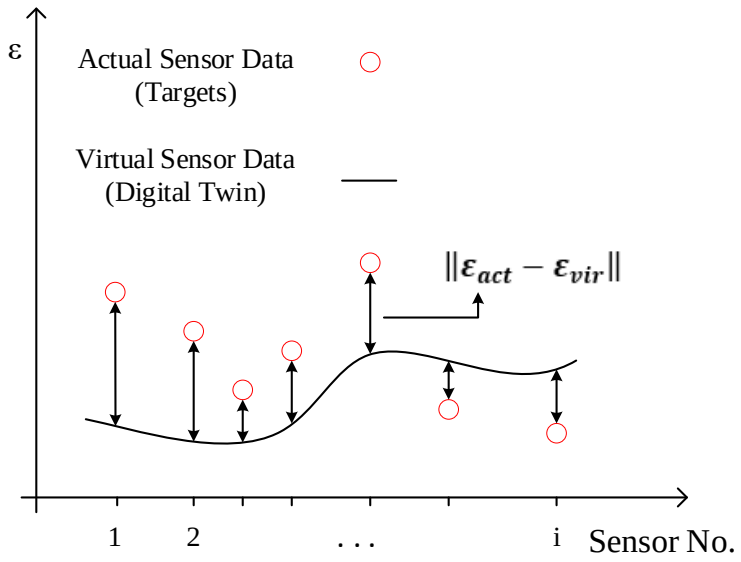


Figure 3.35: Damage Detection as an Optimization Problem

It was thus made clear what the obvious choice for an objective function in this problem is, at least in a conceptual way. However, for it to lead to a successful GA formulation, some more considerations have to be taken into account. The first of those, is the type of norm used to measure the distance between the two strain vectors. In this method a norm commonly used in optimization problems was chosen; that is the Euclidean or L^2 norm which is the standard norm used to calculate the length of a vector in a Euclidean space. For an n-dimensional vector $x=[x_1\ x_2\ \dots\ x_n]^T$ it is given by:

$$\|x\|_2 = \sqrt{\sum_{k=1}^{n} x_k^2} \tag{3.9}$$

The other consideration that had to be addressed to be able to implement the GA, was to determine whether the problem would be considered a single-objective or a multi-objective optimization problem. Essentially, within the scope of implementation in this work this came down to the number of objective functions that should be defined. Ultimately it was decided to use the multi-objective approach, which leads to eight different objective functions being evaluated, one for each path where SDIs are being monitored. Mathematically, this is expressed in the following way:

$$\left\| \varepsilon_{act}^{(i)} - \varepsilon_{vir}^{(i)}(x) \right\|_2 = \sqrt{\sum_{j=1}^{k} \left(\varepsilon_{act}^{(i)} - \varepsilon_{vir}^{(i)}(x) \right)^2}, \qquad i = 1,2,\ldots 8 \tag{3.10}$$

Where *i* denotes the number of the path monitored and *k* the number of sensors per path. The reason why this was chosen was because it takes into account the local nature of damage more effectively.

Having given a rigorous description of the way the traditional notions of optimization were used to enable damage detection, at this point the actual implementation of this method in the context of the digital twin must be addressed. At first, the APDL code was modified to read an input file containing the actual damaged case strain vector $\boldsymbol{\varepsilon_{act}}$. The other modification added to the APDL code concerned the post-processing section. Specifically, the evaluation of the norms described in Eq. (3.10) was added to that section. The strain vector $\boldsymbol{\varepsilon_{vir}(x)}$ is evaluated in the code also from the solution of the FE model produced for the various design vectors.

The implementation of the GA was made possible by the DesignXplorer feature of ANSYS Workbench. This feature allows for the APDL code to be uploaded and the input and output parameters to be chosen by the user. Then, this code is connected to an optimization module where the main parameters of the optimization technique selected can be set. For this method where GAs were used, the Multi-Objective Genetic Algorithm (MOGA) option was selected. The basic parameters for MOGA, which include initial population, population of each generation, maximum number of generations, allowable Pareto percentage and converge stability criterion, are also chosen by the analyst in this module. The domains and types (discrete or continuous) of the design variables i.e., the inputs chosen before, are defined there as well.

After these parameters have been defined, an initial sample population of design vectors is generated, in this case corresponding to various damage cases. For the design vectors generated, which are also the parameters used to create the FE model, the digital twin is built, meshed, the boundary conditions and predefined constant loads applied and then the model solved. Afterwards, the post-processing commands are executed, and the prescribed norms are calculated. This is done internally without any interference from the user. After the initial population has been initialized, the individuals' fitness functions are evaluated and they are ranked, according to the principles laid out in Chapter 2.3.3.

After being ranked, they undergo reproduction, crossover, and mutation in order to create the next generation of offspring. This process is repeated until a convergence criterion is met. In this case, this is based on one hand on a Pareto percentage criterion, according to which a certain percentage of the individuals in a given generation must belong to the Pareto optimal set, in order for the algorithm to converge. On the other hand, a convergence stability criterion is used which essentially judges the stability of a generation with regard to the previous one. This is done by evaluating the minimum, maximum, range, mean and standard deviation for each generation created after the initial population has been defined. These are scaled appropriately and compared to the ones corresponding to the previous generation by way of the variations of the mean and standard deviation. If these variations are below the value set for the convergence stability criterion, then the algorithm has converged.

In the following page in Figure 3.36 a flow chart is provided showing the basic framework of the GA-based damage identification method, as described above.

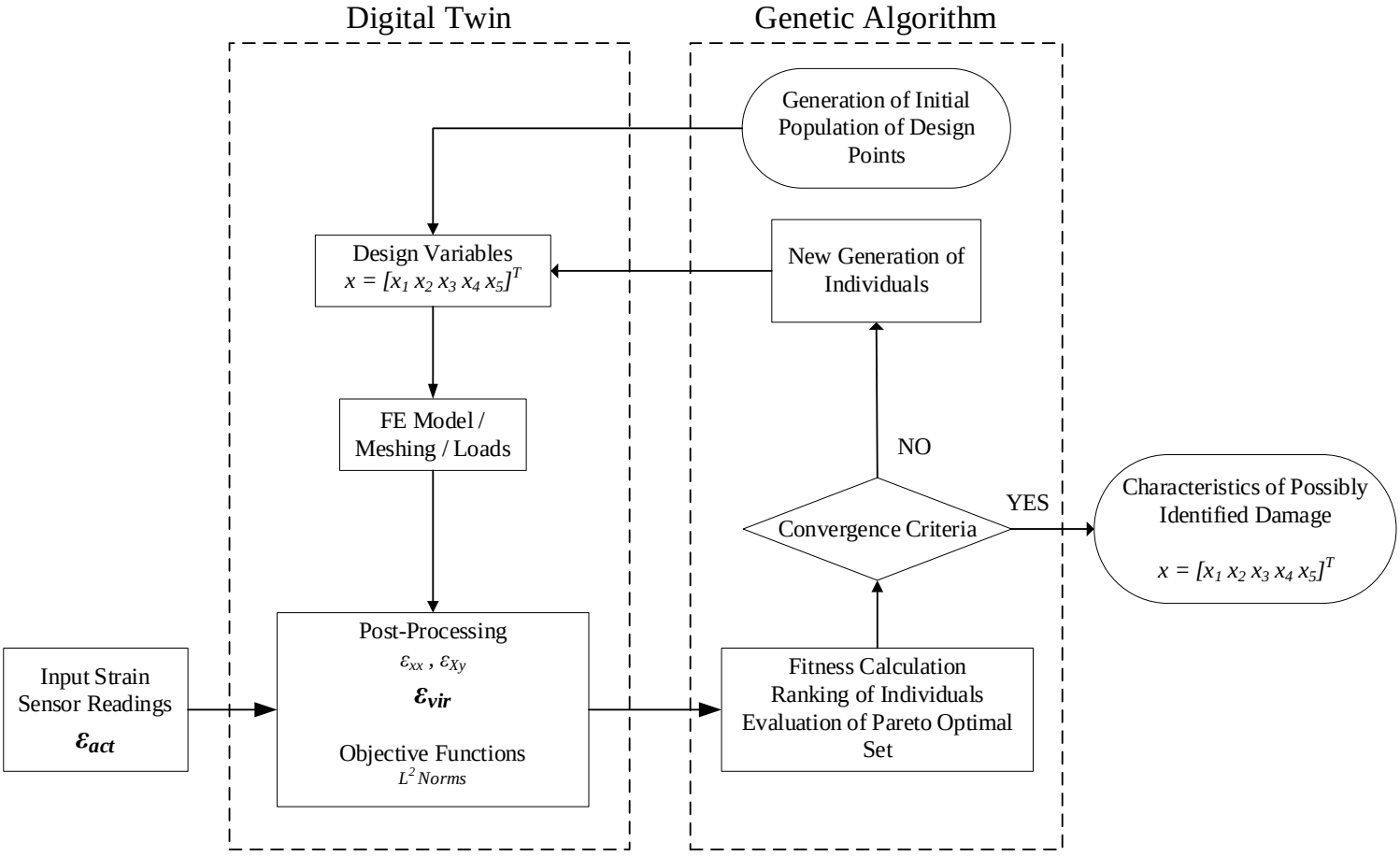


Figure 3.36: Framework of GA-based Damage Identification Approach

3.7 Artificial Neural Network-Based Approach for Damage Identification

The other method developed within the context of this thesis to enable damage identification for a digital twin based SHM system, was realized using Artificial Neural Networks (ANNs). The relevant tools provided by the MATLAB programing environment were used to develop, train, and test the ANNs. In this section, the networks built to address the problem of damage detection are presented and the reasoning behind their choice is explained. Also, the principal features of the datasets used in their training are laid out here. The fundamental concepts concerning ANNs were presented in a detailed way in Chapter 2.4 and generally will not be explained again here, apart from some concepts which were vital to this application.

At first, it should be mentioned that the ANNs considered here follow the paradigm of supervised learning. Therefore, in their respective datasets both input vectors and their corresponding target (label) vectors are included. In our case, the input vector was essentially the strain vector containing the SDIs, i.e. axial and in-plane shear strains monitored from their respective strain monitoring paths specified in Chapter 3.4.3, and obtained from the digital twin, by iteratively solving the model for the various damage scenarios, i.e. the target vector. The target vector essentially contains the identifying features of damage, dealt with in each network. It has been alluded to, that more than one networks were built and trained in the context of the approach followed to achieve damage detection through the use of ANNs. To better understand this, the problem of damage identification has to be viewed through the lens of machine learning.

As theorized in Chapter 3.5.2, all defects considered in this work can be uniquely identified by a set of parameters that contain all the necessary information about their location and magnitude. By virtue of their definition those parameters divide the problem of damage identification to two stages: the first one being the confirmation of the existence of damage, controlled by the span and face identifiers and the second one regarding the quantifying of damage, with respect to its exact location and magnitude. In terms of machine learning, these two stages belong to two inherently different classes of problems: classification, with regard to the first case, and with regard to the second, regression. It is therefore made clear that more than one ANNs must be developed in order to effectively address the complete problem.

In the first stage of detection, what is essentially dealt with are two separate classification problems. The first one is that of the determination of the span where the defect is located, that is its location with respect to the midspan of the girder. In machine learning terms, this is basically a binary classification problem, where the correct classification of a given input vector corresponds to the identification of the span on which the defect is located. The second problem refers to the prediction of the face where the hole is located, which essentially can be treated as a multi-class classification problem, where there are four classes each one corresponding to a face on the girder. In the second stage of detection, the problem dealt with is essentially the prediction of

the location and radius of the SFD. Essentially, this is a regression problem in which the aim is to provide an approximation of a function relating the location and magnitude of damage to the resultant strain-sensor data. At this point, having described the problem of damage identification in the context of machine learning, the methods employed in this work to realize this task are detailed by individually describing each ANN developed.

Span Classification Network

As mentioned above, in the first stage of the process of detection the span on which the defect is located is identified. This was achieved in practice using the MATLAB programming environment and more specifically its "patternnet" function. This function takes as input the number of hidden layers, the type training function and the type of performance function to be used by the network and returns a pattern recognition network, which is essentially a feedforward network that can be trained to be able to classify inputs according to target classes.

The number of hidden layers as well as the size of each layer i.e., the number of neurons in each layer are largely responsible for the performance of the network. There is no standard recipe regarding the appropriate number of layers or neurons for each case, therefore it is often a matter of experience and trial and error. In the case of this network, it was observed that one hidden layer containing 10-12 neurons was the structure providing the most consistent results. Regarding the selection of the training function, the one chosen uses the Levenberg-Marquardt or damped least squares algorithm to tune the weights and biases of the network. This was chosen because MATLAB recommends it as the fastest backpropagation algorithm contained in its toolbox, as well as because it provided stable positive results. As far as the performance function was concerned, the mean squared error between inputs and network outputs was used.

The input vector used to train the network, was essentially comprised of strain results corresponding to the SDIs monitored along their respective strain monitoring paths, as those were described previously and may be denoted as $\boldsymbol{\varepsilon}_{\text{in}}$ going forward. The size of this vector varies with the number of sensors located on each path and will be further expanded upon in Chapter 4 dealing with the implementation and results of this approach. The target vector for this network was a vector comprised of 0s and 1s only, defined in the following way:

$$target = \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & right\ span \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, & left\ span \end{cases} \tag{3.11}$$

The number of input and target vectors used i.e., the size of the dataset, was decided in advance in such a way as to contain as many and as diverse cases of damage as possible in order to build a robust network. The strain values used in the input vector were obtained by solving the digital twin as many times as was needed to cover all the prescribed damage scenarios. Further details on this subject, as well as the splitting of

the dataset to training, validation and testing subsets will be detailed further in Chapter 4.3.

Face Classification Network

As the face detection component of the overall damage identification problem can be considered a multiclass classification problem, the same basic principles were applied here as with the span classification network previously. Namely, "patternnet" was used here as well and also, the same network architecture was found to be ideal for this application. This means that one hidden layer of 10-12 neurons was used, the training was done using the Levenberg-Marquardt backpropagation algorithm and the network performance evaluated using the mean square error between inputs and network outputs. The same input vector $\boldsymbol{\varepsilon_{in}}$ as with the span classification network was used, as essentially the same damaged cases are considered. The target vector, as a multiclass problem is dealt with, was defined again using only 0s and 1s. This time however, the column vectors were four-dimensional, with the 1 located in the row whose index corresponds to the number identifying the face considered. For example, if the defect is located on the left web, its identifier is 2 and the corresponding target (label) vector is $[0\ 1\ 0\ 0]^T$. In this case as well, the percentage of the initial dataset allocated to training, validation and testing as well as performance and other relevant metrics will be presented in the next chapter.

SFD Characteristics Prediction Network

The second and final stage of detection concerns the prediction of the characteristics of the SFD i.e., the circular hole introduced to the digital twin to simulate the effect of damage on the stress and strain fields. Those characteristics essentially refer to the location and magnitude of the hole. The location is expressed in terms of the center's longitudinal distance from the nearest boundary condition and its transverse location with regard to the longitudinal axis of symmetry of the face on which is located. The magnitude of the hole is naturally expressed by its radius. These characteristics constitute the target vector used to train this ANN, while the input vector is, as in all neural networks developed, the vector $\boldsymbol{\varepsilon_{in}}$ of sensor-strain data corresponding to the SDIs and monitoring paths, which have been described in detail.

As mentioned earlier, the problem of predicting the basic characteristics of damage is essentially a problem of function approximation, with this function approximating the relationship between a particular case of damage and the resultant strain distribution on the paths monitored. Therefore, a regression or "fitting" network was chosen using the "fitnet" command provided by MATLAB. Regarding network architecture, again one hidden layer was used containing 9-10 neurons. This number of neurons was found to have the best performance when the network was trained. As far as the backpropagation algorithm used, the Levenberg-Marquardt algorithm for damped least squares was selected for this network as well, as it is the fastest and more efficient one provided by the MATLAB toolbox. Finally, as far as the compartmentalization of the dataset to training, validation and testing sets is concerned, that is dealt with in the following

chapter, where the performance and training of the ANNs using digital twin-obtained virtual strain sensor data.

To summarize this section, an ANN-based method was developed to solve the inverse problem of determining the structural state of a thin-walled girder subjected to three-point bending, given a known structural response. The problem was considered in the context of machine learning as a two-stage problem. In the first stage, the prediction of the face and span where the defect was located were considered as classification problems and addressed by developing two pattern recognition networks. These take in as input strain data obtained from simulations performed by the digital twin and as target vectors appropriately structured column vectors of 0s and 1s. In the second stage, the prediction of the location and magnitude of damage were addressed, in the context of a regression problem and by the development of such a neural network. The outputs produced from the three different networks, constitute the set of identifying parameters, used for damage detection throughout this thesis. In Figure 3.37, a flowchart describing the framework behind this process is provided.

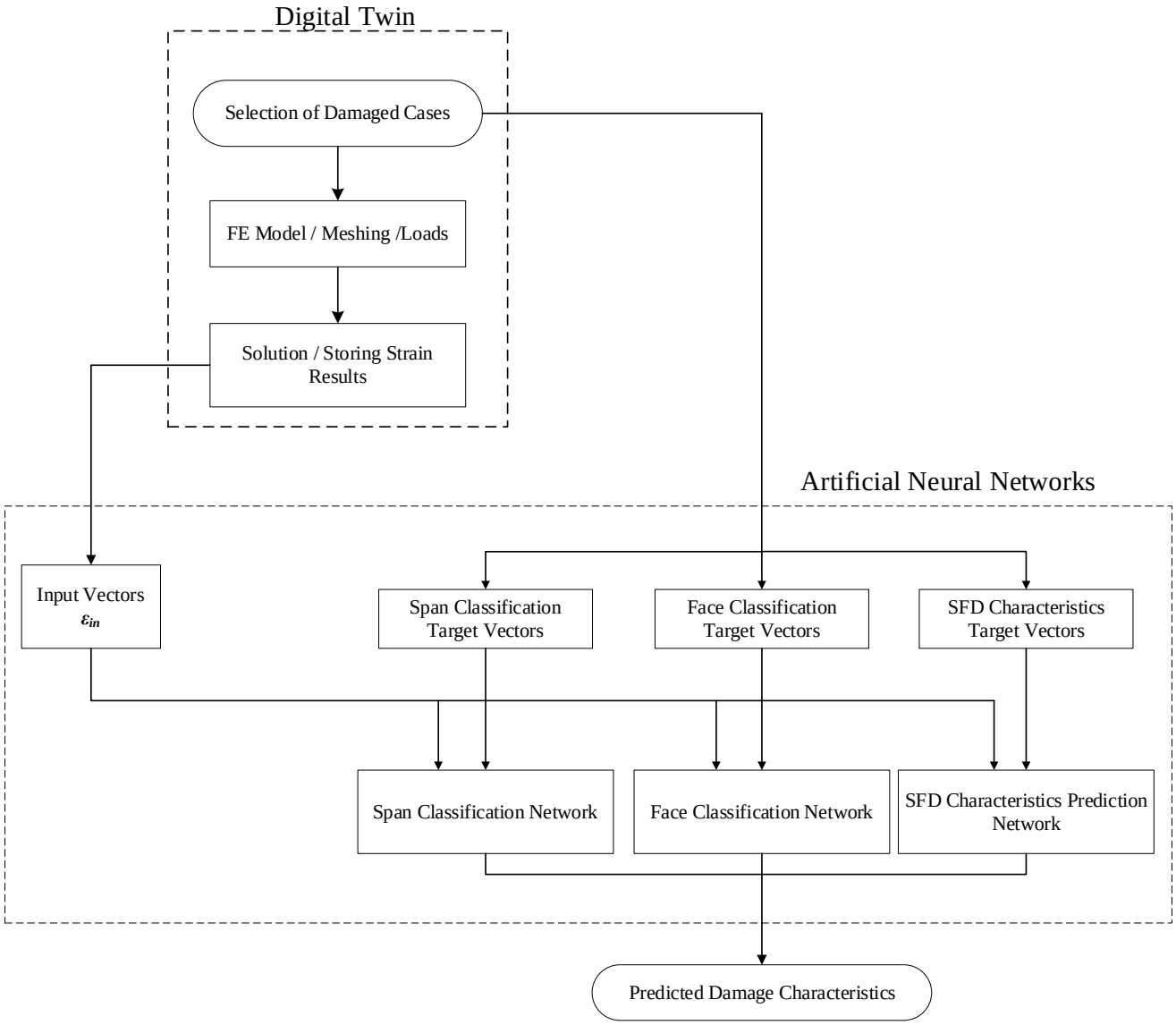


Figure 3.37: Framework of ANN-based Damage Identification Approach

4 Validation in a Virtual Environment

4.1 Introduction

In this Chapter, the two methods previously employed to achieve damage detection within the context of the digital twin based SHM system are tested within a virtual environment. In Chapter 4.2, the GA-based method is tested using input strain measurements obtained from the digital twin and the parameters of the GA are tuned accordingly to increase the method's performance. Furthermore, noise is artificially introduced to the input strains to simulate actual real-world conditions and to test the method's robustness in that regard. Finally, the method's results and the conclusions drawn from them are discussed. In Chapter 4.3, the ANN-based approach is implemented using strain results from the digital twin for training and testing data alike. The proposed training dataset is presented along with the performance results of the relevant networks. Also, their predictive capabilities are investigated using previously unknown input data, obtained yet again from the digital twin and the results are commented on. Finally, in Chapter 4.4 the two methods are compared with regard to their performance, robustness and applicability and the overarching conclusion drawn are discussed.

4.2 Validation of the GA-Based Approach

4.2.1 Numerical Validation Framework

As was mentioned in Chapter 3.6, the GA-based method for damage identification developed in this thesis uses as input strain-sensor vectors corresponding to monitored SDIs along their respective paths. These vectors, denoted as $\boldsymbol{\varepsilon}_{\boldsymbol{vir}}^{(\boldsymbol{i})}$ and defined in Eq. (3.8) are *k*-dimensional column vectors, with *k* being the number of sensors per path. In total, there are as many such vectors as there are paths; therefore, there are eight such vectors. Regarding the number of sensors located in each path i.e., the dimension of those vectors, this was chosen as a parameter of the validation process. Namely, the method was implemented on three configurations containing 16, 10 and 5 sensors per path, located at equidistant intervals. This was done in order to investigate the influence of the monitoring scheme on the system's capability to detect damage, both when "clean" and "noisy" measurements were used.

In this stage of validation, these vectors were obtained from the digital twin by solving the FE model for various damaged cases corresponding to predefined locations, shown in Figure 4.1. The locations of the various defects were chosen in such a way as to provide representative results for all the monitored areas. Namely, half of the cases were located on a flange and the other half on a web, while also covering the left and

right span of the girder equally. Moreover, in terms of their local placement on each face, it was made sure that they were spaced across both its transverse and longitudinal directions. In terms of their magnitude, the same radius was selected for all cases. The damage identifiers for these damaged cases are also presented in table form below. In Table 4.1, the ones corresponding to the flange cases are included while in Table 4.2 the ones corresponding to the web case are shown. It must be noted that: x refers to the longitudinal distance from the nearest boundary condition; y denotes the transverse position with regard to the longitudinal axis and r is naturally the radius of the circular hole. All of the above are expressed in millimeters.

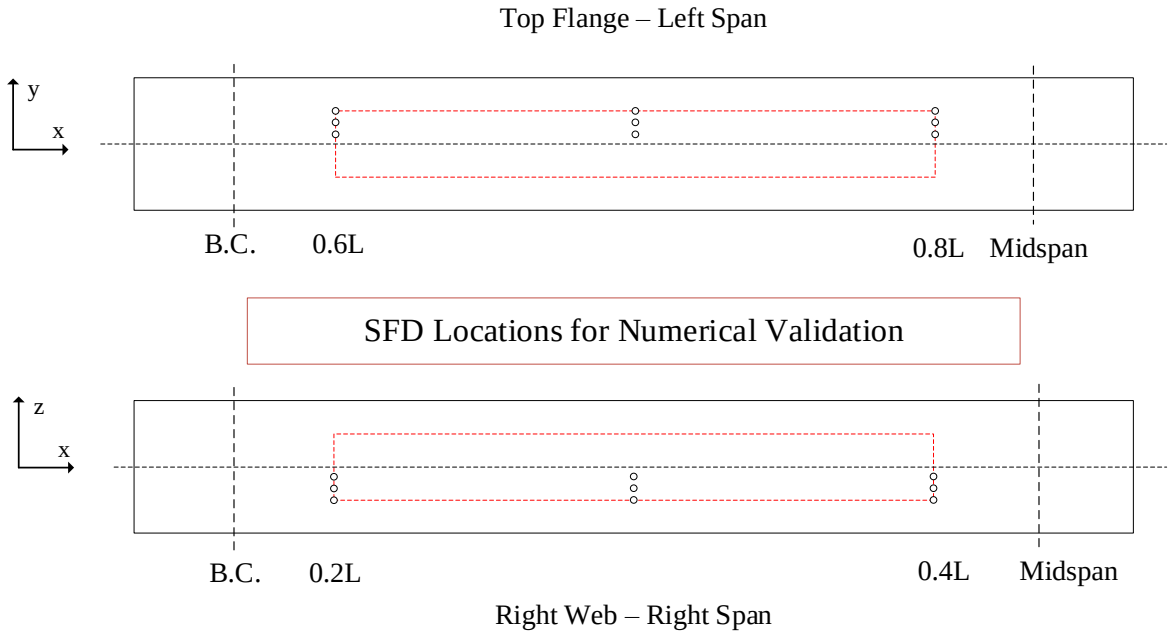


Figure 4.1: SFD Locations used for Validation in the Virtual Environment

Table 4.1: Particulars of Top Flange – Left Span Damage Cases

Top Flange (1) – Left Span (2)										
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
x	[mm]	150	150	150	300	300	300	450	450	450
y	[mm]	4.5	14	23	4.5	14	23	4.5	14	23
r	[mm]	4	4	4	4	4	4	4	4	4

Table 4.2: Particulars of Right Web – Right Span Damage Cases

Right Web (1) – Right Span (1)										
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
x	[mm]	150	150	150	300	300	300	450	450	450
y	[mm]	-4.5	-14	-23	-4.5	-14	-23	-4.5	-14	-23
r	[mm]	4	4	4	4	4	4	4	4	4

As shown from the above, 18 overall cases were considered for the numerical validation of the developed method. For each case, the damage identifiers were set accordingly in the APDL code and the strain results from each path were saved in output files. These were then used as input for the digital twin in the code employed for the GA-based method in the ANSYS Workbench environment, which is shown in Figure 4.2 below. As mentioned earlier, ANSYS DesignXplorer uses MOGA, a hybrid of the NSGA-II algorithm and allows for the principal parameters to be controlled by the user. These include the number of individuals in the initial population, the number of individuals in each subsequent generation, the maximum number of generations as well as the convergence criteria such as the maximum allowable Pareto percentage and convergence stability criterion, which have been explained in detail in Chapter 3.6 and the number of final candidate solutions. It should be noted also that the algorithm supported by this environment uses only one crossover point.

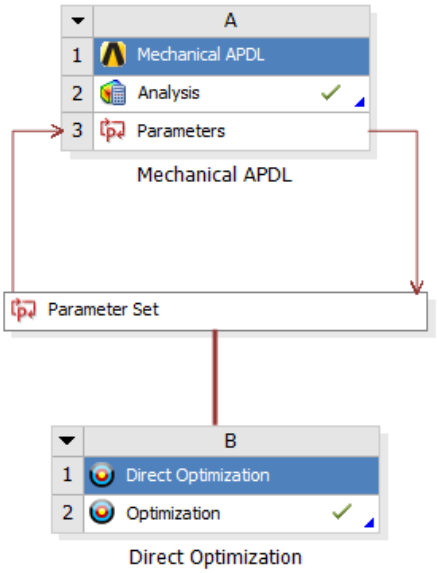


Figure 4.2: Screenshot from the ANSYS Workbench Environment used in the GA-Based Approach

4.2.2 Results without Noise Pollution

It was mentioned earlier in this chapter that the predictive capabilities of this method were tested against results with and without added noise pollution. This was done to gain an insight on the developed method's predictive capabilities in real-world conditions. However, before this was attempted the method was tested on data obtained from digital twin simulations, which although purely numerical and thus idealized -as they do not represent the uncertainties inherent to real-world measured data- can provide a solid baseline on the method's capabilities and limitations. Also, they may be very useful as a means of tuning the method's parameters in order to obtain optimal performance.

With that in mind, in order to find the most successful "recipe" of MOGA parameters, different parameter schemes were used in testing the method on the aforementioned damage cases. These were then compared with regard to their performance, expressed

in terms of the success they achieved in detecting each damaged case. The three such parameter schemes used for the purpose of this thesis are presented in Table 4.3 below.

Table 4.3: Initial MOGA Parameter Schemes

MOGA Parameter	Assigned Value		
	Scheme No. 1	Scheme No. 2	Scheme No.3
No. of Individuals in Initial Population	120	120	140
No. of Individuals in Each Generation	40	50	50
Max. Pareto Percentage (%)	70	70	70
Convergence Stability Percentage (%)	2	2	2
Max. No. of Generation	20	20	20
Max. No. of Candidates	1	1	1
Type of Discrete Crossover	One Point	One Point	One Point

At this point, all of the parameters in the method have been set apart from the number of sensors per path. It was decided, as mentioned previously, to employ three different configurations of 16,10 and 5 sensor per path. At this stage, the three MOGA schemes used to tune the method's parameters were implemented on the 16-sensor configuration, with the aim to test the other configurations using the most effective parameter scheme. Therefore, what follows are the results obtained by performing the requisite simulations, using as input non-polluted data corresponding to the cases mentioned in Chapter 4.2.1 and to the 16 sensors per path configurations.

Regarding the presentation of these results, the following should be noted. The results are provided in the form of percentages of successful detection which is expressed with four separate percentages. The first one deals with what is considered as first stage detection; namely, whether the span and face where damage is located are correctly identified, analogous to the role of the classification neural networks in the ANN approach. Then, the prediction of the particulars of damage is addressed by presenting what percentage of predictions were within certain proximity intervals. Specifically, regarding the distance x from the nearest boundary condition, the interval was set at 20 mm; regarding the transverse location y with regard to the longitudinal axis of symmetry it was set to 4.5 mm and finally with regard to the radius r it was set to 0.2 mm. Those described above are presented in Table 4.4 below.

Table 4.4: Damage Detection Accuracy for MOGA Parameter Schemes (16 Sensors per Path)

	Flange Locations			Web Locations		
	Scheme 1	Scheme 2	Scheme 3	Scheme 1	Scheme 2	Scheme 3
Detection (Face & Span)	88.89%	100.00%	100%	88.89%	88.89%	100.00%
x (Within 20mm)	0.00%	22.22%	55.56%	33.33%	22.22%	44.44%
y (Within 4.5 mm)	11.11%	44.44%	22.22%	33.33%	22.22%	66.67%
r (Within 0.2 mm)	11.11%	0.00%	22.22%	22.22%	22.22%	11.11%

Although all three schemes were very successful in first stage detection, it became evident that the first two were lacking when it came to providing a more accurate prediction on defect location. Also, none of the schemes showed consistency in accurately predicting the magnitude of damage, which however is not a primary concern at this point and thus was not considered as a deciding factor in choosing the "recipe" to be used from now on. The most important factor was the ability to achieve first stage detection and then came the ability to predict the longitudinal component of the location. The reason is that by combining these two pieces of information, a very comprehensive understanding of the location of the defect is given, in the scope of this application. Therefore, it is evident from all those reasons that the third scheme was the one displaying the greatest performance and thus will be used for all other simulations going forward.

The success of this particular parameter scheme is attributed to the fact that the large number of initial population individuals was uniquely suited to the particular problem dealt with here. Namely, the diversity in the initial population was what allowed the method to overcome the local nature of damage; that is, by making sure that candidate points would definitely exist in the immediate neighborhood of the target. That way, the algorithm was able to take advantage of this locality and thus converge rapidly to the correct solution without sacrificing speed compared to. Figure 4.3 the other schemes, albeit having a greater population of candidate points and therefore requiring more calls to the solver to compute the requisite objective functions shows a three-dimensional representation of the pareto optimal points for an indicative damage case.

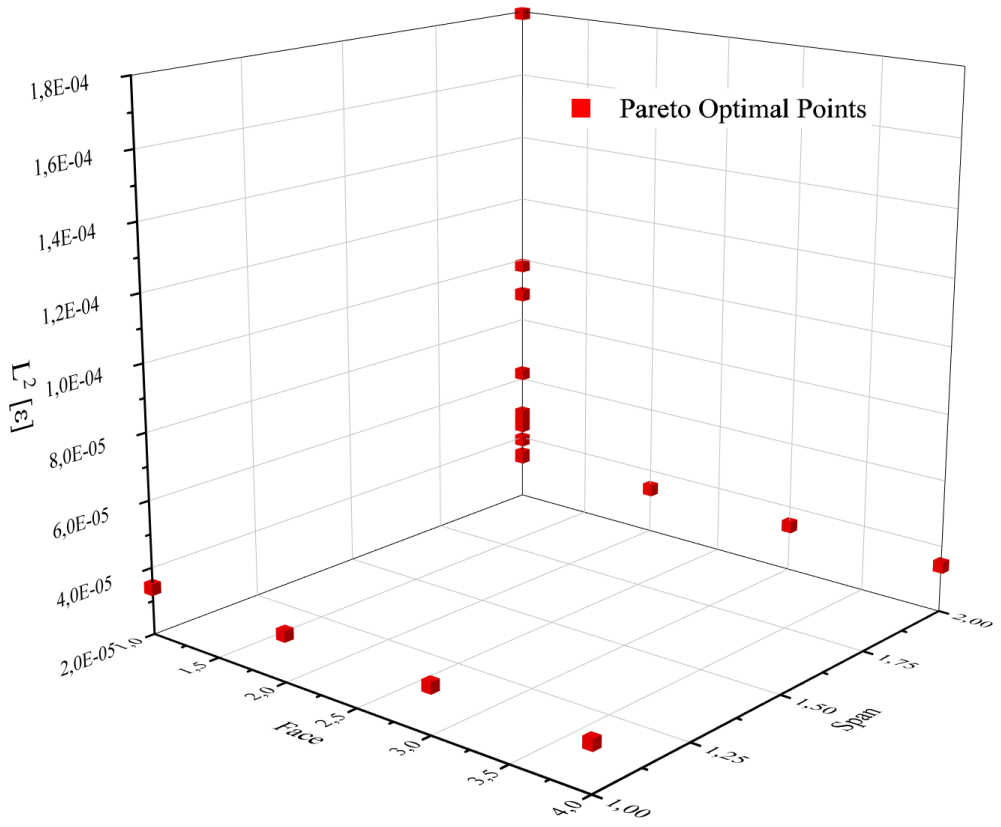


Figure 4.3: Pareto Optimal Points for Indicative Damaged Case with 16 Sensors per Path

The identifiers for span and face are plotted in the horizontal axes x and y, while in the vertical axis the objective function corresponding to the path nearest to the defect location is shown. The particular case considered in this graph corresponds to Case 7 of Table 4.1. It is evident that the majority of Pareto optimal points are found at the combination of face and span corresponding to those of the target i.e., top flange (1) and right span (2). The point found to have the lowest objective function value was indeed the candidate point selected by the algorithm.

Having established the "recipe" of MOGA parameters which yielded the greatest accuracy, the method was then tested on a different set of damaged cases, this time less extensive as the capabilities of the method had been proven, which used as input vectors of lower dimension. That means, that the strain-sensor results simulated by the digital twin were this time obtained from a smaller number of sensors corresponding to each path where SDIs were measured. As mentioned earlier, the configurations that were considered used 10 and 5 sensors per path. Table 4.5 shows the locations of prospective damaged cases used in this stage of the validation process. The notation is the same as the one used in Table 4.1 earlier.

Table 4.5: Particulars of Damaged Cases for Validation of 5 & 10 Sensor Paths

		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Face	-	Bottom Flange (3)	Top Flange (1)	Bottom Flange (3)	Right Web (2)	Right Web (2)	Left Web (4)
Span	-	Left Span (2)	Left Span (2)	Right Span (1)	Right Span (1)	Left Span (2)	Right Span (1)
x	[mm]	200	300	360	150	240	400
y	[mm]	9	14	-14	-4.5	9	18.5
r	[mm]	4	4	4	4	4	4

The results obtained from the MOGA runs for those cases, are presented below in Figure 4.6 for both sensor configurations. The presentation of these results is made in the same way as those earlier in this chapter.

Table 4.6: Damage Detection Accuracy of GA-based Method for Indicative Cases with 5 & 10 Sensors per Path

	5 Sensors per Path	10 Sensors per Path
Detection (Face & Span)	100.00%	100.00%
x (Within 20mm)	66.67%	66.67%
y (Within 4.5 mm)	100.00%	66.67%
r (Within 0.2 mm)	50.00%	16.67%

Observing the results presented above it was made clear that the chosen MOGA parameter scheme continued to yield very good results even when the number of sensors across every path was decreased. Even more, the predictive capabilities of the method regarding location and magnitude were found to be improved overall. Figure 4.4 below shows the Pareto optimal points for Case 3 of Table 4.5, with the face and span identifiers plotted on the horizontal axes and the objective function corresponding to path nearest to the defect plotted on the vertical axis. The majority of Pareto optimal points, i.e. feasible solutions, lie on the vertical corresponding to the bottom flange (3) and right span (1) combination, which showcases the algorithms strong convergence to the correct solution.

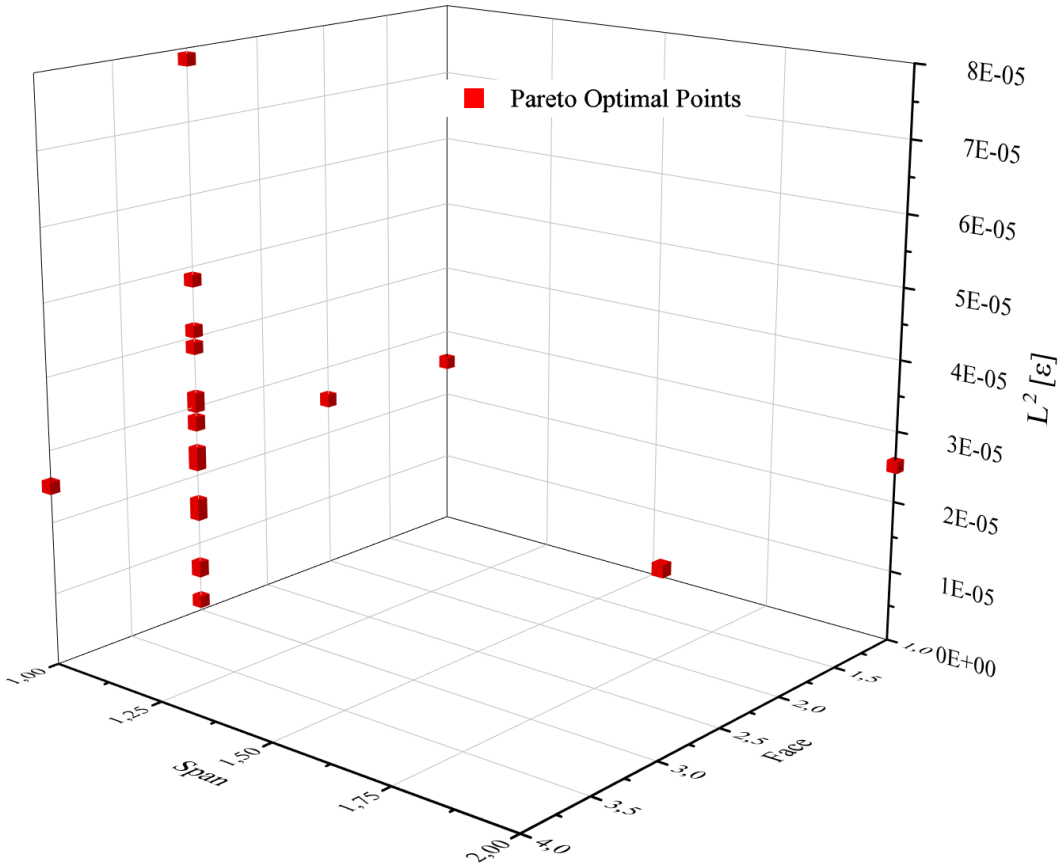


Figure 4.4: Pareto Optimal Points for Indicative Damaged Case with 5 Sensors per Path

Finally, in Figure 4.5 an overview of all those discussed above is presented. Namely, the cumulative percentages achieved by all sensor configurations and for all four parameters of detection are presented. It is shown that the 5-sensor configuration displayed the overall best performance, however all three have produced encouraging results. It should be noted that the decrease in success rate mentioned earlier refers to the location and magnitude predictions, as first stage detection was ensured in all cases and sensor configurations with this parameter scheme. The reason why performance was slightly decreased with the increase in sensor numbers was attributed to the algorithm's fast convergence when the size of the input vector was increased. Specifically, a higher dimension strain vector in conjunction with the locality of the effect of damage on the strain field, enabled the generation of fitter individuals in the

initial population, which in turn led to the algorithm converging faster to its final solution. On the other hand, when less sensors were considered, the algorithm had to generate more design points until the convergence criteria were met, thus increasing the probability of a more accurate solution.

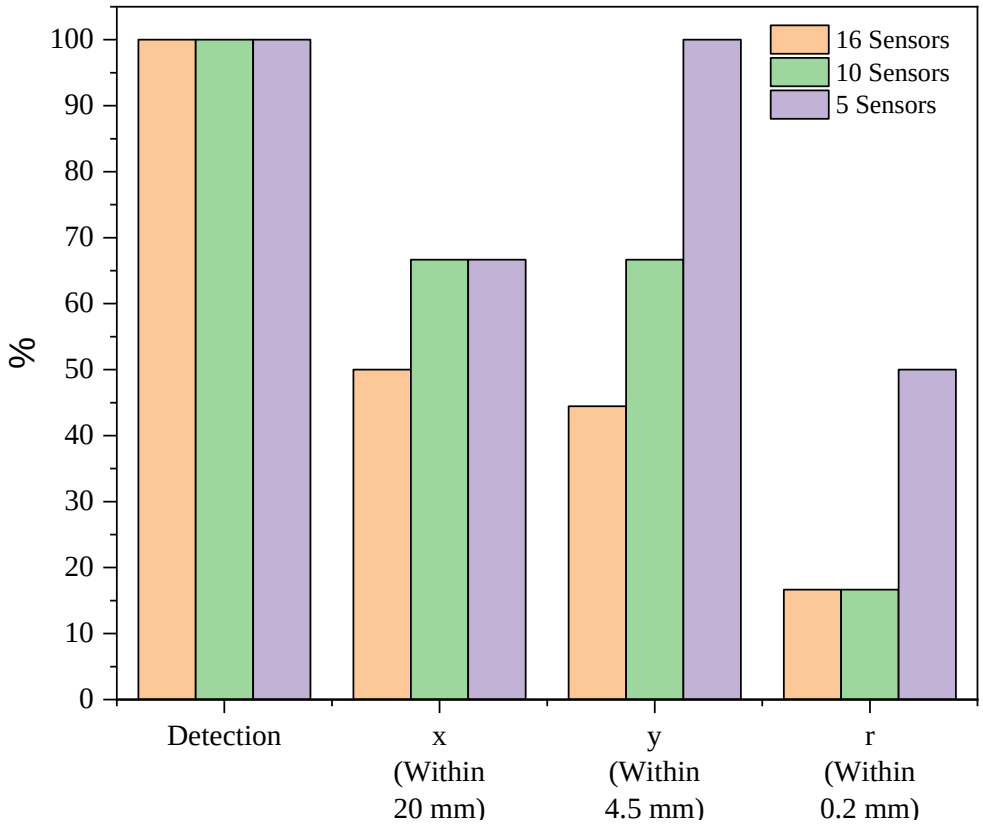


Figure 4.5: Final Detection Accuracy Results for GA-based Method and Non-Polluted Strain Data

4.2.3 Results with Added Noise Pollution

In the previous chapter, the method was implemented using target results obtained by simply solving the FE digital twin and extracting the necessary outputs. However, these measurements are idealized, due to the fact that the FE model, as a numerical simulation, cannot replicate actual measurement errors which are inherent to the use of common measuring devices such as strain gages. So, it was made clear that however encouraging might initial results be, they should not be considered indicative of the absolute capabilities of the method in question. Therefore, the integration of noise pollution in the numerical results had to be first considered before any definitive conclusions on the robustness of the method were drawn.

In order to create noise pollution, pseudo-random noise was inserted on the idealized digital twin strain readings using randomly distributed values over a predefined interval. This was achieved by importing the “clean” strain readings in a MATLAB script, and adding noise values from a uniform distribution appropriately scaled to the desired interval. The extent of the interval was considered as a parameter indicating the sensitivity and robustness of the approach and initially, the error of measurement was

set at an interval of [-3,3] με. Noise was introduced to strain reading corresponding to all sensor configurations presented above, aiming to establish a threshold under which the method's capabilities remain uncompromised. Following are two graphs comparing the idealized versus the realistic strain results for an indicative damage case, namely Case 9 of Table 4.1. The term idealized refers to results obtained directly from FEA, while realistic refers to those to which noise was introduced, as mentioned earlier. Figure 4.6 displays the strain distribution along the two top flange paths where damage is located, while Figure 4.7 that corresponding to the right web paths. In this case, the level of added noise was set at the [-3,3] με interval mentioned earlier.

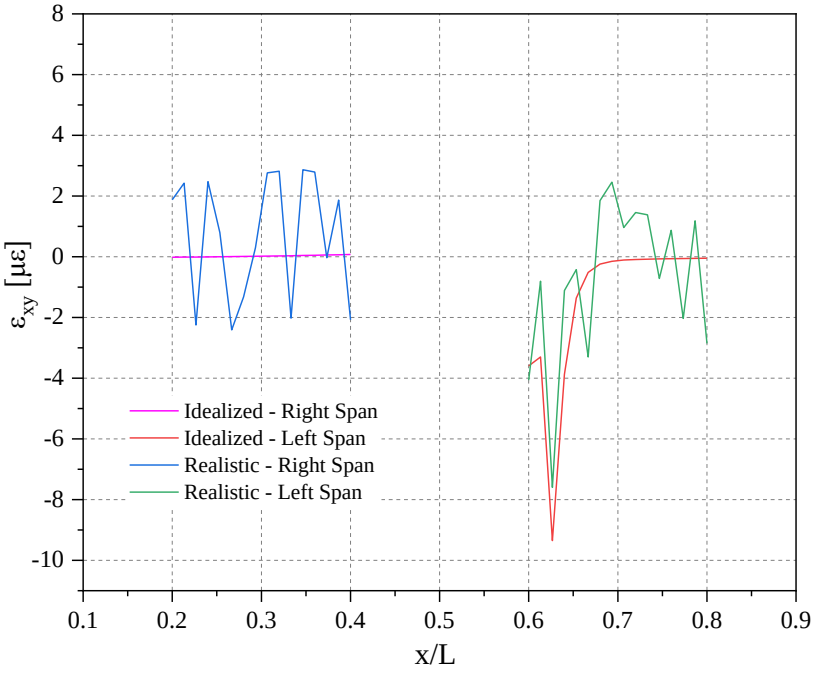


Figure 4.6: Comparison of Idealized and Realistic Strain Distributions for Top Flange Paths

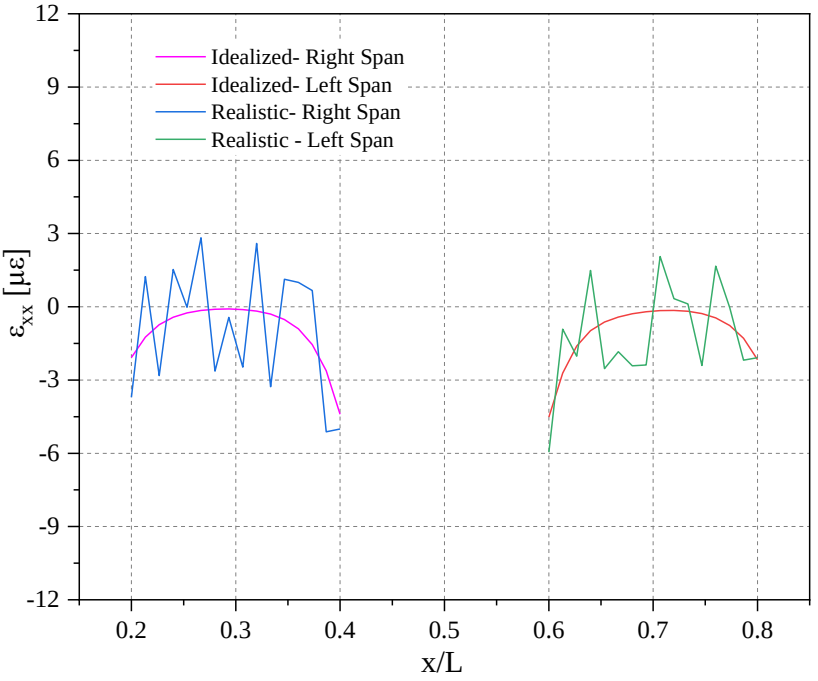


Figure 4.7: Comparison of Idealized and Realistic Strain Distributions for Right Web Paths

It can be observed from the above that especially in the case of in-plane shear strains monitored on the flange paths, the levels of added noise are orders of magnitude greater than the digital twin results. The same was the case for the web paths where damage was not present. In the immediate vicinity of the defect, although the effect of noise was clear, the local effect of damage was still present. This should allow the algorithm to converge to that solution effectively, however it is possible that the noise added to the other sensor paths could hinder its predictive capabilities.

As mentioned earlier, the overarching goal at this point was to establish the noise threshold under which detection is possible for every sensor configuration. In all cases, initially noise was added at the [-3,3] με range and was then adjusted accordingly, with regard to the obtained results. In the 16-sensor configuration, the capability of detection was maintained at this range, for web defect cases in particular. Regarding flange locations, certain limitations arose. It became apparent that at the aforementioned noise levels, detection of flange located defects was hindered. Namely, a tendency was observed that the noise threshold decreased as the defect was placed farther from the monitoring path. In any case, this threshold was consistently within the ± 1.5-2 με interval.

Regarding the 10-sensor configuration, a general decreasing tendency was observed for both web and flange cases. Initially, added noise levels were set at the same values as for the 16-sensor case, however the results that came up were not satisfactory. So, the range was decreased accordingly, and it was observed that for web cases, the effective threshold was reduced to [-2,2] με for the majority of cases. For flange cases, this decrease amounted to half the previous value, meaning that the acceptable range of added noise in this case was found to be within the [-0.5,0.5] με interval. It should be noted that outliers were found where detection was impossible at this level of noise, however these were singular cases that were expected considering the random nature of added noise.

This same tendency was also observed in the 5-sensor configuration, for web damage cases in particular; that is, the interval within which the method maintained its predictive capacity was decreased, this time to the [-1.5,1.5] με range. A singular outlying case was documented here as well; however, this was not indicative of overall performance. Regarding the flange cases, the various “runs” indicated that the effective range of damage detection remained the same as the 10-sensor configuration, i.e. within the [-0.5,0.5] με interval.

Therefore, it is safe to say that a decrease in the number of equally spaced sensors leads to the method becoming more susceptible to noise pollution. This was suspected however, considering the local effect of damage on the strain field. Namely, when an extensive network of sensors is employed the disturbance in the strain field is accurately captured, and even when noise is included remains clearly defined. On the contrary, this is not the case when sensors are more sparsely located along the paths, unless the defect lies in the immediate vicinity of one of the sensors employed.

To complete the presentation of results contained in this chapter, the effects of noise pollution on some indicative cases are presented below. Specifically, one case for the web and one for the flange is shown for each sensor configuration, comparing the results with and without noise to showcase the effect of noise on the method's accuracy in each case. The level of noise pollution is also mentioned to provide context for the parameters used in each simulation.

Table 4.7: Effect of Noise Inclusion on Damage Prediction Accuracy of GA-based Method – 16 Sensor Configuration

		Flange Case			Web Case		
		Added Noise Level: [-2,2] με			Added Noise Level: [-3,3] με		
		Target	Predicted (Non-Polluted)	Predicted (Polluted)	Target	Predicted (Non-Polluted)	Predicted (Polluted)
Face	-	Top Flange (1)	Top Flange (1)	Top Flange (1)	Right Web (2)	Right Web (2)	Right Web (2)
Span	-	Left Span (2)	Left Span (2)	Left Span (2)	Right Span (1)	Right Span (1)	Right Span (1)
x	[mm]	150	155	155	450	450	450
y	[mm]	4.5	14	14	-23	-23	-23
r	[mm]	4	3.8	3.8	4	4.4	4.4

Table 4.8: Effect of Noise Inclusion on Damage Prediction Accuracy of GA-based Method – 10 Sensor Configuration

		Flange Case			Web Case		
		Added Noise Level: [-0.5,0.5] με			Added Noise Level: [-2,2] με		
		Target	Predicted (Non-Polluted)	Predicted (Polluted)	Target	Predicted (Non-Polluted)	Predicted (Polluted)
Face	-	Bottom Flange (3)	Bottom Flange (3)	Bottom Flange (3)	Left Web (4)	Left Web (4)	Left Web (4)
Span	-	Left Span (2)	Left Span (2)	Left Span (2)	Right Span (1)	Right Span (1)	Right Span (1)
x	[mm]	200	175	175	400	405	405
y	[mm]	9	18.5	18.5	18.5	23	23
r	[mm]	4	4.8	4.8	4	4	4

Table 4.9: Effect of Noise Inclusion on Damage Prediction Accuracy of GA-based Method – 5 Sensor Configuration

		Flange Case			Web Case		
		Added Noise Level: [-0.5,0.5] με			Added Noise Level: [-1.5,1.5] με		
		Target	Predicted (Non-Polluted)	Predicted (Polluted)	Target	Predicted (Non-Polluted)	Predicted (Polluted)
Face	-	Top Flange (1)	Top Flange (1)	Top Flange (1)	Right Web (2)	Right Web (2)	Right Web (2)
Span	-	Left Span (2)	Left Span (2)	Left Span (2)	Left Span (2)	Left Span (2)	Left Span (2)
x	[mm]	300	340	175	240	270	260
y	[mm]	14	14	-9	9	9	9
r	[mm]	4	3	3.8	4	4.4	4.6

The results presented above confirm what has been discussed earlier; namely, that the GA-based damage detection method that was developed performs robustly when no added noise is included and also that it maintains this accuracy when noise is added, within certain noise levels. It should be noted that these noise levels represent threshold values under which not even first stage detection i.e., face and span classification, is achieved, indicating that the developed method remains very accurate until it becomes altogether unable to perform its purpose. Finally, the decreasing tendency of the range of added noise where detection is achieved while the number of sensors located on each path is also decreased, is clearly shown in the above as well.

4.3 Validation of ANN-Based Approach

4.3.1 ANN Training and Performance

In Chapter 3.7, the basic principles and tools used in the development of an ANN-based approach for damage detection were explained in detail. The present chapter deals with the particulars of the implementation of this method in a virtual environment. The dataset used to train the neural network is detailed at first; then, the results obtained from the training are given in the form of performance plots and other relevant metrics and finally some indicative predictions are presented, in the same manner as those referring to the alternative method developed for this thesis, as was detailed in Chapters 4.2.2 & 4.2.3 previously.

As mentioned earlier, there were three separate networks developed to address the problem of damage detection: a classification network responsible for the prediction of the span where the defect was located; a classification network responsible for the

classification of the face on which damage was located; and finally, a regression network aiming to predict the characteristics of damage, i.e. its location, expressed in terms of its longitudinal distance from the nearest support and transverse location with regard to the centerline of the corresponding face. The other characteristic of damage i.e., its magnitude, expressed as the radius of the circular hole, was decided to be omitted from the training dataset as initial results indicated that it added significant complexity to the training data which could not be quantified and dealt with in another way. Therefore, for all intents and purposes the radius of the hole was considered to be constant and equal to 4 mm.

The relevant strain data required to create the training dataset were to be obtained, as mentioned before, from the digital twin. Namely, the FE model had to be a solved a number of times in order to provide both the input and target (label) vectors. Therefore, a scheme involving the various damage scenarios had to be considered first. Specifically, the various cases of damage considered had to be decided in advance in terms of their location, and in such a way that the resultant dataset would be both large enough and would provide adequate representation of the possible damage scenarios, so that a robust and accurate neural network could be trained. A schematic of the pattern employed is shown below in Figure 4.8 and discussed afterwards.

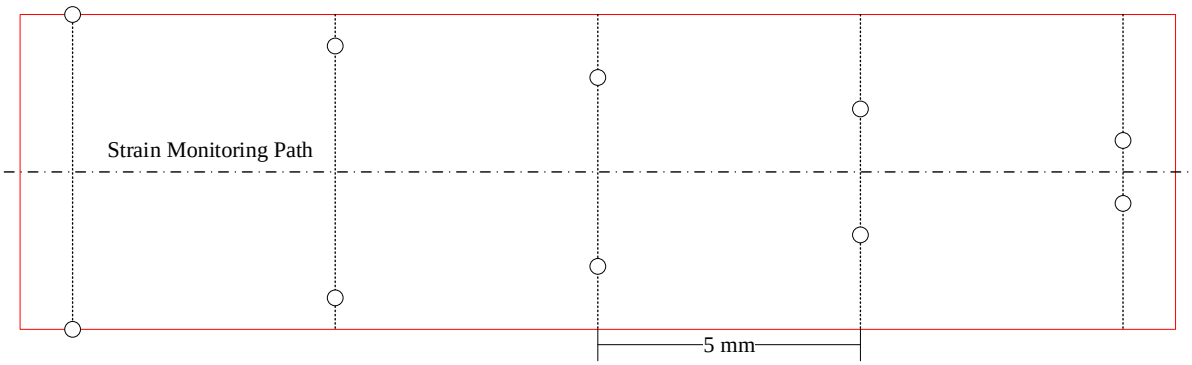


Figure 4.8: Damage Scenario Possible Location Pattern for ANN Training Dataset Generation

The pattern shown above corresponds to a region within the area where damage can be introduced, as explained in Chapter 3.5.1. It is also repeated in the same way in all four faces of the girder. At longitudinal locations located at one element length intervals, i.e. every 5 mm, damage is introduced in locations located symmetrically with regard to the face's longitudinal axis of symmetry. Thus, every 25 mm all possible transverse locations where the hole may be located, as imposed by the limitations detailed in Chapter 3.5.1, have been covered. In this way, at every possible longitudinal location eight total defects are mapped across all four faces. This ultimately yields a total number of 976 different damage scenarios, which constitute the basis on which the training datasets for the three developed neural networks were constructed.

To complete the description of the datasets used in the training of these ANNs, another key factor has to be addressed first, that of the size of the input vectors, which as has been mentioned, are common to all networks and are essentially vectors of strain readings obtained from their respective monitoring paths. In the same way as in the

GA-based approach, there were three sensor configurations considered, again with 5,10 and 16 sensors located at each path. The only difference in this case was that, in order to save valuable computational resources by solving the FE model only once for all cases, the sensors corresponding to the 5-sensor path were not equidistant but were set at specific locations in advance. Following is a presentation of the datasets used for each ANN and the way they were divided to training, validation and testing sets as well as representative performance and other metrics plots which were produced as a result of their training.

Span Classification Network

The basic particulars employed in this neural network, such as the training algorithm, the number of hidden layers as well as that of neurons per layer and the performance function, were detailed in the relevant section of Chapter 3.7 and therefore will not be repeated here. Regarding the input vector, it is a vector of strain readings obtained from all strain monitoring paths located on the girder. Therefore, it may be denoted as $\varepsilon_{in}^{(i)}$ where i=1,2, … ,976 and refers to the possible damage scenarios considered in the dataset. Regarding its dimensions, it is either a 40×1, 80×1 or 128×1 column vector in the case where 5,10 or 16 sensors per path respectively, are considered. As far as the target vector is concerned, that is a binary column vector as defined in Eq. (3.11).

Finally, the dataset was allocated to training, validation and testing as follows: 75% for training; 15% for the validation set; and 10% for the testing set. It was considered useful to mention here that the training set is used to tune the network's weights and biases and to compute its gradient; the validation set is used to monitor its error in order to facilitate early stopping to the network if the validation error begins to increase, thus avoiding overfitting; and finally, the test set which is independent from the other two and is used to estimate the networks' performance. Figure 4.9 shows an indicative performance plot for the network trained with data from the 10-sensor configuration.

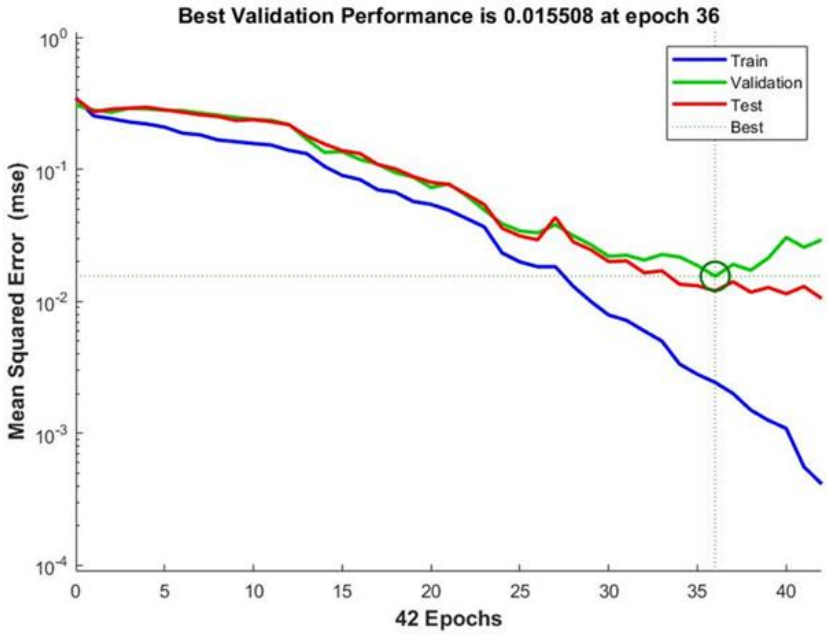
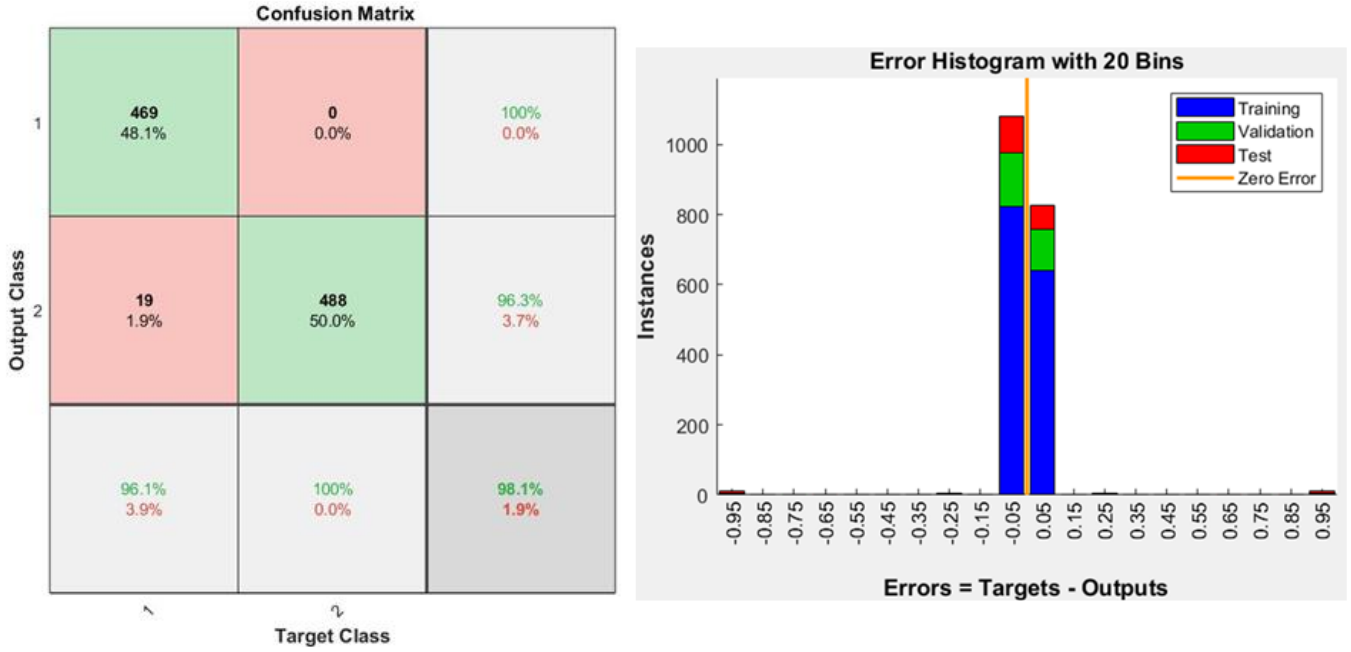


Figure 4.9: MSE Performance of Span Classification ANN – 10 Sensors per Path

The same overall performance was observed for the neural networks developed using the datasets corresponding to the other sensor configurations, so it was decided to not include them for the purpose of conciseness. Regarding the performance plot shown above, it is evident that no significant overfitting occurs before the best validation performance is achieved, and where early stopping takes place, and that all curves follow the same decreasing trajectory. Also, the fact that the testing and validation curves reach their minima concurrently, is an indicator that the choice of dataset division was a successful one. Finally, the confusion matrix obtained from the same neural network training instance, as well the error histogram are provided in Figure 4.10 (a) and (b) respectively, to further illustrate the classification network's performance.



(a) (b)

Figure 4.10: (a) Confusion Matrix & (b) Error Histogram of Span Classification ANN – 10 Sensors per Path

Face Classification Network

Again, the basic particulars employed in the development of the ANNs which were employed to address the face classification problem, have been detailed in Chapter 3.7 and will not be iterated here. The same holds for the description of the input vector, which was described again in the previous section and as it is the same for each network will not be described again hereafter. However, as the problem of correctly identifying the face on which damage was located has been treated as a multiclass classification problem, some more information on the label vector were considered useful. Namely, that was again defined as a vector of 0s and 1s, but this time was a 4×1 column, with the row containing the 1 being that whose number is the same as the identifier used for a specific face. In this case also, three different ANNs were trained using input vectors corresponding to the three sensor configurations. Regarding the splitting of the dataset, the same paradigm was followed here as well; that is 75% for training, 15% for validation and 10% for testing. Following in Figure 4.11 is an indicative performance

plot obtained when training this neural network using the 16-sensor dataset. Here as well, no overfitting occurs before the best validation performance is achieved and all curves follow similar decreasing trajectories. Also, in Figure 4.12 (a) the confusion matrix obtained from the same training instance for this neural network is provided and the error histogram is given in Figure 4.12 (b).

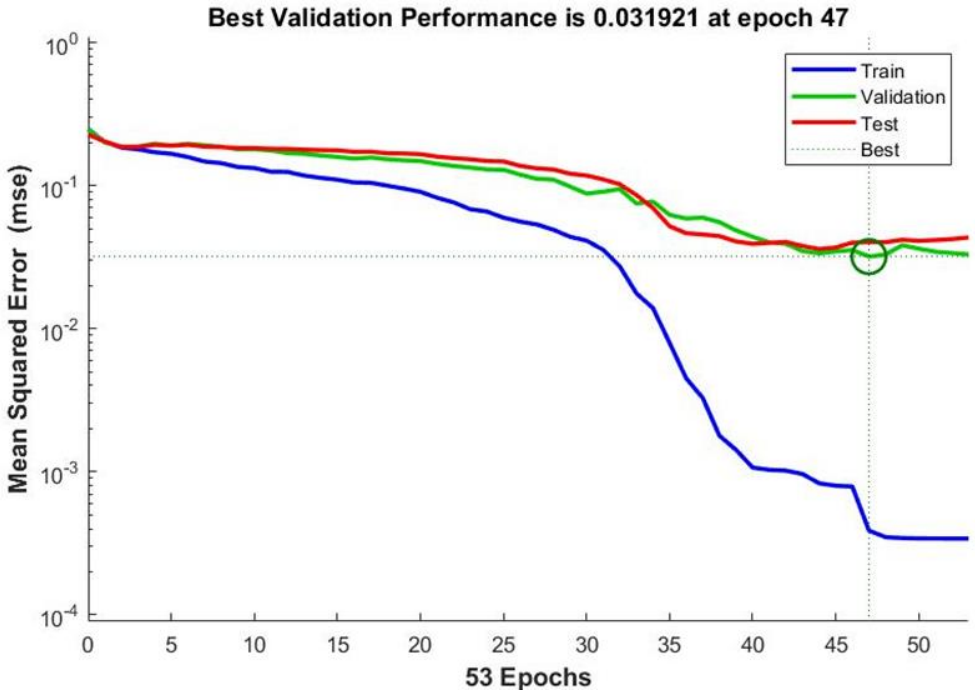


Figure 4.11: MSE Performance of Face Classification ANN – 16 Sensors per Path

(a) (b)

Figure 4.12: (a) Confusion Matrix & (b) Error Histogram of Face Classification ANN – 16 Sensors per Path

Finally, it is worth mentioning that in this case as well, all networks had similar performance characteristics and therefore the rest were not included.

SFD Characteristics Prediction Network

The final group of neural networks developed to handle the problem of damage detection, were those dealing with the prediction of the particular characteristics of damage scenarios. As mentioned before, it was elected to omit the radius of the hole from the target vector as it introduced complexity which led to difficulties in the network's ability to generalize. Therefore, the target vector used was essentially a two-dimensional column vector, defined as $[x\ y]^T$ where x and y are defined as explained in the beginning of the present chapter. Regarding input vectors, here as well the same strain vectors were used to train three different fitting ANNs, one for each configuration of sensors. It should be noted that neither the input nor the target data were normalized before being fed to the neural network, as this is done internally for the fitnet command by MATLAB and no significant improvement in performance was observed when this was attempted. As far as the particulars of the network architecture are concerned, these were detailed in Chapter 3.7 and thus shall not be repeated at this point. The division of the dataset used to train the fitting networks was chosen to be the same in all three cases; namely, 75% of the dataset was used in training, 15% was used for the validation set and the remaining 10% for the test set. Figure 4.13 below contains the mean squared error performance of the fitting network trained for the case where 5 sensors per path are considered.

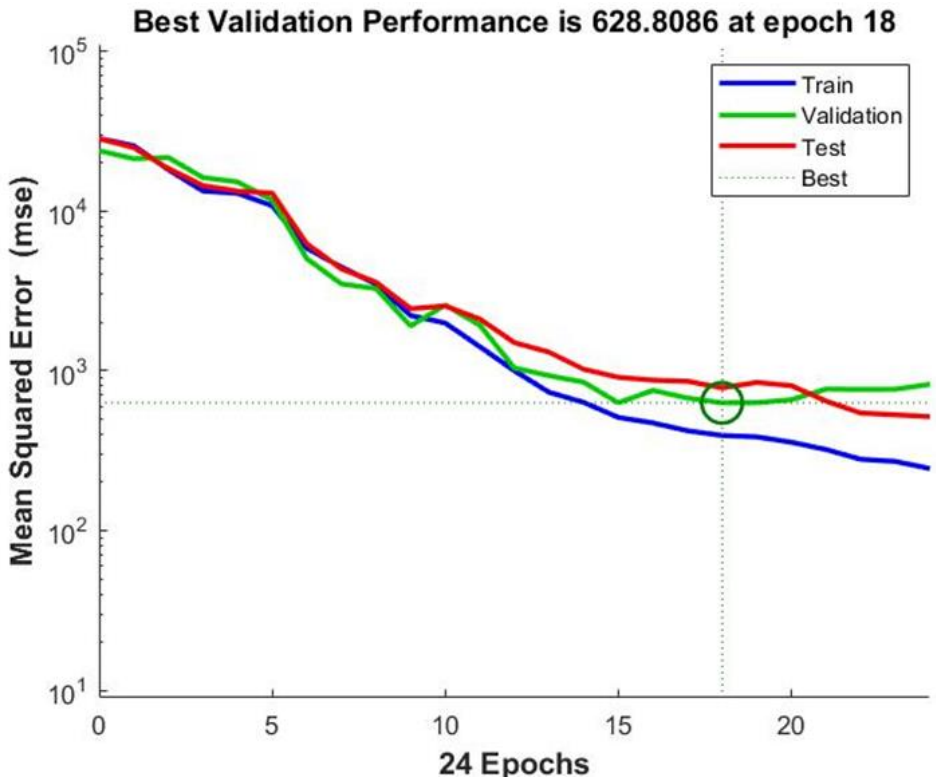


Figure 4.13: MSE Performance of SFD Characteristics Prediction ANN – 5 Sensors per Path

In this case as well, no overfitting occurred until epoch 18, when the best validation performance was achieved and for which the network parameters were set. It should be noted at this point that in MATLAB early stopping is defaulted to six after which the validation performance does not decrease anymore. Also, the high final performance

value is due to the fact that the data were not normalized before input, as this is done internally for the fitnet command which generates function approximation and non-linear regression neural networks in MATLAB, and which as mentioned in Chapter 3.7 was the tool used for this application.

In addition to that, a plot showing the overall regression for the entire dataset, including training, validation and testing, is shown in Figure 4.14 below, corresponding to a dataset obtained from paths containing 5 sensors each. It is evident, that the data fitted lie mainly only the 45° line, which is the line indicating complete equivalency between output and target data. Therefore, this plot indicates that the fitting of data in this case is very satisfactory. Also, it should be noted that there are two clusters of data, corresponding to the x and y data, the former spread in the range between 150 and 450, as expected with some outliers existing beyond this interval, and the latter at the bottom part of the graph, again as expected as these correspond to a range between -23 and 23 με.

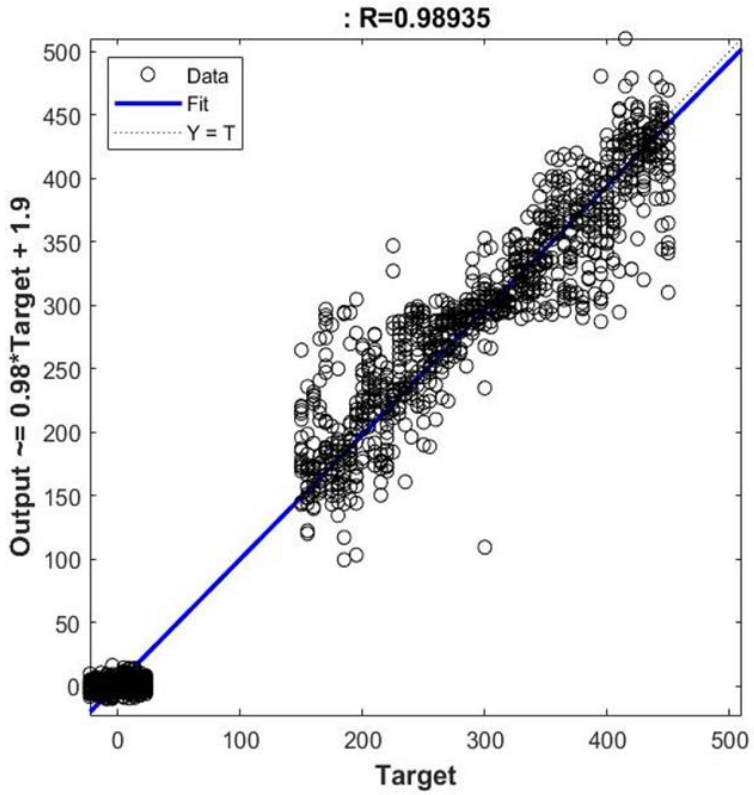


Figure 4.14: Overall Regression of SFD Characteristics Prediction ANN – 5 Sensors per Path

Finally, it should be noted that for the remaining sensor configurations the corresponding plots were very similar and therefore were not included. Having described the particulars of training the ANNs developed to facilitate damage detection, some indicative predictions obtained from them for various damage scenarios are presented in the following sections.

4.3.2 Results without Noise Pollution

As opposed to the GA-based approach, the process of tuning the various network parameters was not validated by predicted results; rather, this was done through the various performance plots and relevant metrics detailed in the previous section, which in large part ensure the predictive capabilities of the neural networks. Of course, it should be noted that these changed each time the networks were trained but in general they are indicative of a baseline that was reached during the training process. However, to provide a comprehensive outlook on the present method's predictive capabilities, it was deemed necessary to provide some indicative results showing its accuracy.

In the same way as the previous method, this one was tested against the same array of damage scenarios. Initially, the input vectors of strain readings consisted of data obtained solely from the digital twin i.e., idealized numerical strain values. This allowed for a baseline to be established on the ANN-based approach's capabilities to identify damage. First, it was tested against inputs corresponding to the damage scenarios described in Figure 4.1 as well as Table 4.1 and Table 4.2 of the previous section. The presentation of the results follows the same principle now as it did then, with the exception that a prediction for the radius is not included, as it was not used as a variable in this case, as has been mentioned previously. The complete results for the cases considered for the 16-sensor configuration are presented in Table 4.10.

Table 4.10: Damage Detection Accuracy of ANN-based Method for Indicative Cases with 16 Sensors per Path

	Flange Locations	Web Locations	Total
Detection (Face & Span)	100%	77.78%	88.89%
x (Within 20mm)	22.22%	66.67%	44.44%
y (Within 4.5 mm)	22.22%	22.22%	22.22%

As mentioned earlier, the misidentification of either face or span constitutes unsuccessful detection in the context of this thesis, so in this particular method if either of the classification networks produces an erroneous prediction, then it is considered that detection is impossible, regardless of how accurate the predictions from the other two ANNs were. With that taken into consideration, the above indicated that for the given trained state of the neural networks, when the method was implemented, flange damage scenarios were classified without error while the regression network performed less satisfactorily in the task of approximating damage location. In damage scenarios concerning web location, the performance of the classifiers was decreased, however that of the fitting network was more promising.

Additional tests were carried out to investigate the predictive capabilities of the method when using input from paths with reduced sensor numbers. Specifically, the same indicative scenarios of damage were used in this case as well, as with the GA-based approach. These are less in number than in the 16-sensor case, as the baseline for damage detection has already been established for this method. The particular characteristics of these scenarios were presented in Table 4.5 earlier and the reader is encouraged to seek them there if interested to acquire the relevant context. In Table 4.11 below, the accuracy of the ANN-based method's outputs is presented for the two different numbers of sensors located per monitoring path and in Figure 4.15 the complete accuracy results for all three cases are presented in a comparative manner.

Table 4.11: Damage Detection Accuracy of ANN-based Method for Indicative Cases with 5 & 10 Sensors per Path

	5 Sensors per Path	10 Sensors per Path
Detection (Face & Span)	100.00%	100.00%
x (Within 20mm)	33.33%	33.33%
y (Within 4.5 mm)	16.67%	33.33%

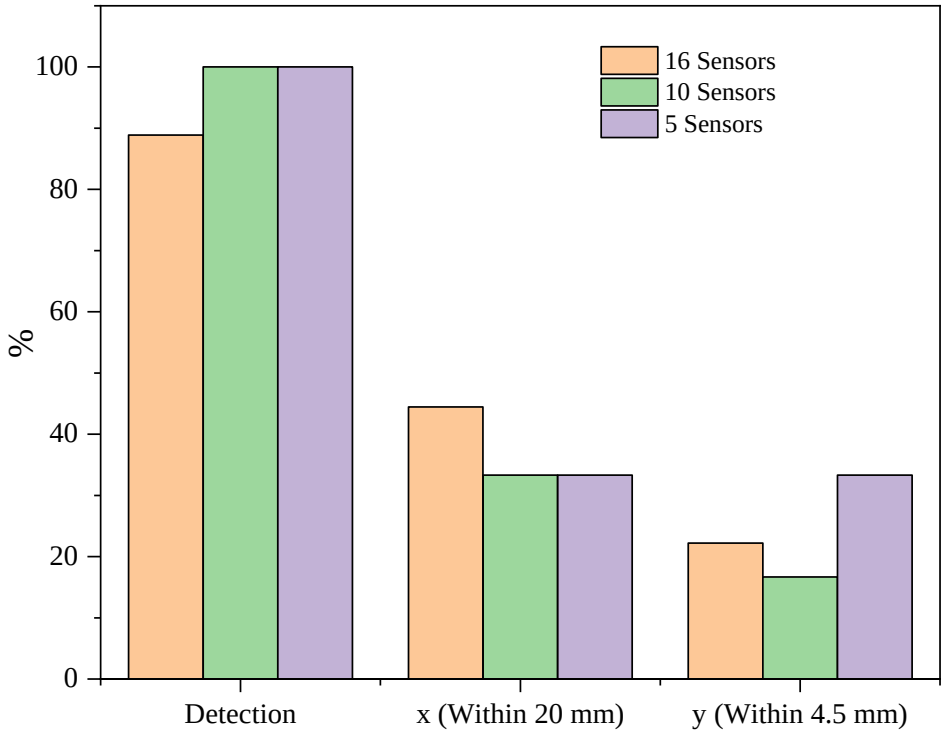


Figure 4.15: Final Detection Accuracy Results for ANN-based Method and Non-Polluted Strain Data

What becomes evident when the above are considered, is that the classification networks enabling first-stage detection were able maintain their high level of accuracy even when trained using an input vector of lower dimensions. The accuracy in the

prediction of damage characteristics does show a decreasing tendency overall, in contrast to what was the case in the GA-based method. This can, in this case, be attributed to the decreased dimension of the input vector used in the training dataset, which "weakens" the neural networks capability to generalize as well. This is a fundamental difference of the two methods, in that the GA-based approach is more case-driven and thus may maintain high accuracy regardless of the number of sensors; however, this type of analysis is reserved for the concluding section of this chapter.

4.3.3 Results with Added Noise Pollution

In the previous section, the predictive capabilities of the ANN-based damage identification approach were established at a baseline level, by testing it using purely numerical digital twin-obtained strain data. So, naturally the next step was to investigate its ability to detect damage in simulated real-world conditions, that is when measurement noise was included in the test data. The inclusion of noise was done in the same way as in the GA-based approach by adding random noise following a uniform distribution and scaled to a desired interval. This interval was set initially at [-3,3] με and then adjusted according to the results. It should be noted that the noise was added only to the strain vectors used to test the neural networks, and not to the training dataset. The ANNs were trained using virtual data obtained by the digital twin. More specifically, while obtaining the results presented in this section it was made sure that the ANNs were tested in the same training state to provide comparable results.

As was the case in the GA-based approach, the reasoning behind the tests that were performed was to establish a noise threshold under which detection is always possible. To obtain results with confidence, the method was tested using various different strain vectors, corresponding to both flange and web cases, to which noise was introduced in the manner described previously. Also, it was made sure that the method would be repeatedly tested for a certain damage scenario in a specific noise range to ensure that accuracy was maintained for this interval. The random nature of the added noise, as well as the way it was built into the MATLAB code used to perform this task ensured that each time a different strain vector was used by the three neural networks.

With that being said, the conclusions drawn were in the same vein as those obtained from the corresponding tests performed on the alternative method, albeit displaying some differences. First of all, scenarios corresponding to damage on the flange were found to be more susceptible to noise pollution; namely, the threshold below which detection was possible was considerably lower for damage cases on flanges than ones located on the webs. The reason behind this happening may again be attributed to the fact that noise is orders of magnitude greater than measured strain across flange paths. On average, for the 16-sensor configuration it was found to be within the [-2.5,2.5] με interval for cases where damage was located on webs, while for flange cases this was at [-1.5,1.5] με. This is considerably less than the corresponding interval for the alternative method that was developed. In similar fashion, for the case where 10 sensors per path were considered, this interval was decreased to [-1,1] με for web cases and

decreased to [-0.5,0.5] με when flange cases were considered. Of course, as with the GA-based method some outliers were found for which detection was impossible even within these limits. They were however not representative of the overall results to which the aforementioned intervals refer. Finally, for the configuration consisting of the least number of sensors the intervals were diminished to the lowest levels out of all cases considered in both methods; namely, to [-1,1] με for web cases and [-0.25,0.25] με for the flange ones. To provide a complete framework on which the comparison between the methods can be made, the same indicative cases used in Chapter 4.2.3 are presented here in the same manner. The aim is to illustrate the ANN-based approach's accuracy using polluted data while also providing the framework to compare the two methods effectively.

Table 4.12: Effect of Noise Inclusion on Damage Prediction Accuracy of ANN-based Method – 16 Sensor Configuration

		Flange Case			Web Case		
		Added Noise Level: [-1.5,1.5] με			Added Noise Level: [-2.5,2.5] με		
		Target	Predicted (Non-Polluted)	Predicted (Polluted)	Target	Predicted (Non-Polluted)	Predicted (Polluted)
Face	-	Top Flange (1)	Top Flange (1)	Top Flange (1)	Right Web (2)	Right Web (2)	Right Web (2)
Span	-	Left Span (2)	Left Span (2)	Left Span (2)	Right Span (1)	Right Span (1)	Right Span (1)
x	[mm]	150	202	209	450	408	388
y	[mm]	4.5	-6.7	-6	-23	-20.5	-4.5

Table 4.13: Effect of Noise Inclusion on Damage Prediction Accuracy of ANN-based Method – 10 Sensor Configuration

		Flange Case			Web Case		
		Added Noise Level: [-0.5,0.5] με			Added Noise Level: [-1.5,1.5] με		
		Target	Predicted (Non-Polluted)	Predicted (Polluted)	Target	Predicted (Non-Polluted)	Predicted (Polluted)
Face	-	Bottom Flange (3)	Bottom Flange (3)	Bottom Flange (3)	Left Web (4)	Left Web (4)	Left Web (4)
Span	-	Left Span (2)	Left Span (2)	Left Span (2)	Right Span (1)	Right Span (1)	Right Span (1)
x	[mm]	200	176	261	400	397	174
y	[mm]	9	6	10	18.5	1	6

Table 4.14: Effect of Noise Inclusion on Damage Prediction Accuracy of ANN-based Method – 5 Sensor Configuration

		Flange Case			Web Case		
		Added Noise Level: [-0.25,0.25] με			Added Noise Level: [-1,1] με		
		Target	Predicted (Non-Polluted)	Predicted (Polluted)	Target	Predicted (Non-Polluted)	Predicted (Polluted)
Face	-	Top Flange (1)	Top Flange (1)	Top Flange (1)	Right Web (2)	Right Web (2)	Right Web (2)
Span	-	Left Span (2)	Left Span (2)	Left Span (2)	Left Span (2)	Left Span (2)	Left Span (2)
x	[mm]	300	307	286	240	305	417
y	[mm]	14	-1	-2	9	0	2.5

What can be immediately inferred from the results shown above, is that the accuracy of the method is definitely decreased when the number of sensors across each path is decreased as well. Also, it can be observed that the predicted coordinates of the hole's location, expressed by x and y, also indicate decreased accuracy even in the non-polluted cases. This has to do with the nature of the fitting neural network used to predict defect characteristics. Namely, the possible longitudinal and transverse locations for the damage scenarios considered in this thesis, are by design discrete due to certain limitations posed by the requirements of the FE mesh, which have been given in detail in Chapter 3.5.1 earlier. However, it is not within the capabilities of the MATLAB tool used in this work to define that the approximated function should also be discrete, and that is the reason why the predicted values that come up are not necessarily part of the discrete domain, specified for these variables in the digital twin's initial formulation and ultimately for the decreased accuracy. Apart from that, the performance of the classification networks remains robust providing consistently accurate predictions, within the specified noise intervals.

4.4 Concluding Remarks

In this chapter, the results from the validation of both methods developed within the scope of this thesis, a GA-based and an ANN-based damage identification approach, were presented. Their validation was enabled from data produced by the digital twin model detailed earlier and were used both in their "clean" form, that is as they were obtained from the FE model and in a polluted form where noise was added artificially to simulate real-world sensor measurements.

What both methods showed was an ability to robustly achieve what was called first-stage damage detection, which means they were able to correctly predict the face and

span of the girder on which the defect was located. When it came to the prediction of particular defect characteristics such as location and magnitude, the GA-based approach proved to be more accurate. This was due to its formulation as this method calls for the repeated dynamic solution of the FE model, as dictated by GA parameters, which by in large ensures that it will be able to capture the intricacies and particular characteristics of the digital twin in a better way, thus yielding overall more accurate results. On the other hand, the ANN-based approach, due to limitations imposed by the programming tools used, was not able to capture those particular characteristics as well; namely, the discrete nature of the variables describing the location components of the defect. Also, the parameter of defect magnitude, as expressed by the radius of the hole which simulated the effects of damage, was omitted as it introduced significant computational complexity. This, coupled with the fact this parameter was not of primary importance in the process of damage identification, provided the justification for this simplification.

When noise was added to the digital twin-obtained strain readings both methods showed similar trends. Namely, a decrease in the levels of noise under which detection was possible with the simultaneous decrease in number of sensors per path was documented. Also, a greater susceptibility to noise pollution leading to misidentification was observed for cases where damage was located on a flange, which was attributed to the baseline of the in-plane shear SDI being significantly lower than the level of added noise. Of the two methods, the ANN-based one proved to be overall more susceptible to noise pollution, with its damage characteristics prediction component being more compromised, as was documented for the “clean” case as well.

Overall, the results provided by both methods were promising showing that both displayed consistency and generally good, even at times exceptional accuracy in predicting the characteristics of an existing damage scenario. Also, valuable insights were drawn at this stage regarding the effects of measurement noise on the predictive capabilities of both methods, which will prove valuable when they are tested using actual data.

5 Experimental Validation

5.1 Introduction

It was mentioned earlier in this thesis, that the case chosen for the development of the digital twin and the damage identification schemes based on it, was selected for two main reasons. That it would be an appropriate simplification of hull structural response under still water loading and also that it would create the opportunity to perform actual experiments and use the recorded data to test the efficacy of the developed damage detection schemes. This is the goal of this present chapter, to describe those tests as well as the steps taken to use the results obtained from them in the damage detection framework that was developed.

The chapter is organized as follows. In Chapter 5.2 the particulars of the tests that were performed are described; in Chapter 5.3 the recorded data from the experiments are presented and discussed; in Chapter 5.4, the implementation of the damage detection methods on the recorded data is detailed; finally, in Chapter 5.5 the main conclusions from this process are discussed.

5.2 Description of the Laboratory Tests

The case considered in this thesis, as has been mentioned numerous times, is that of thin-walled square girder subjected to three-point bending. Therefore, the laboratory test in question was essentially a three-point bending test performed on the hydraulic MTS universal testing machine of the Shipbuilding Technology Laboratory (S.T.L.). A schematic of the three-point bending test configuration is provided in Figure 5.1 below.

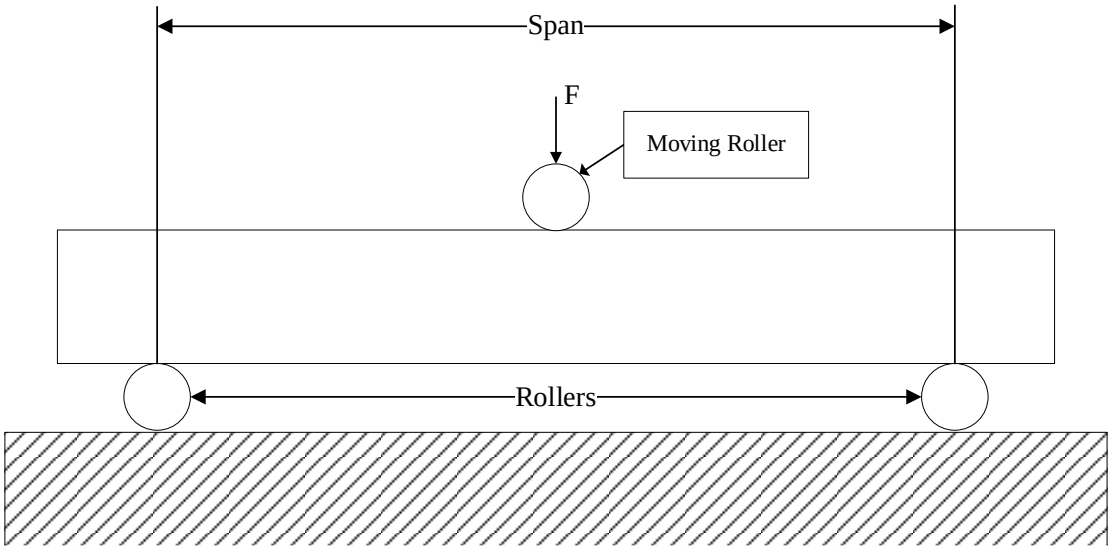


Figure 5.1: Three-Point Bending Experiment Set-Up

Regarding the test configuration, the girder, which will be referred to as the test specimen, is placed on two cylindrical blocks known as rollers which act as the supports. A third cylindrical block, referred to as the moving roller, administers the applied force from the machine to the girder, subjecting the specimen to bending. As far as the specimen itself is considered, a square hollow beam was procured from a commercial vendor and cut to pieces of equal lengths, which were the used as the test specimens. A schematic detailing their dimensions is provided in Figure 5.2 below.

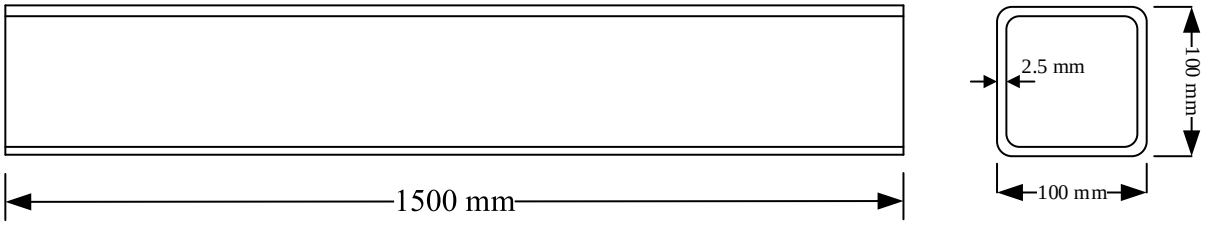


Figure 5.2: Test Specimen Dimensions

The cross-sectional dimensions selected were, nominally, the same as those used in the digital twin model. Regarding the length of the test specimen, this was dictated by the capabilities of the universal testing machine at the disposal of S.T.L. Specifically, it was decided in advance that the maximum span within the capabilities of the facilities was to be used, which is 1200 mm. Therefore, an overall length of 1500 mm for the entire specimen was deemed as an adequate choice, as it satisfies the span requirement while also providing additional practical benefits, related to the procurement of the specimens.

The next step in preparing the specimens for the experiments was deciding the configuration of strain sensors employed during the tests. The sensors used were strain gages, namely rosette strain gages capable of measuring displacement in two directions, having a 90-degree angle between them. These are known in the literature as tee rosettes and may be referred to as such in this chapter. Regarding their number, eight in total were used; four were mounted on the top flange and the other four were mounted on the left web, which was the web where the defect i.e., the hole, was introduced. Their number was chosen based on practical considerations regarding the capabilities of the data acquisition system used in the tests. Finally, the span of the girder on which they were located, corresponds to the left span in the digital twin formulation.

Regarding strain gage placement, a decision was made, according to the theoretical understanding of the process as well as the numerical results produced by the digital twin, to arrange the sensors in the immediate vicinity of the proposed damage locations. Therefore, the following pattern, identical for both flange and web sensors, was implemented: Strain gages were to be mounted on the theoretical location of the neutral axis on the webs i.e., mid-height, and on the zero in-plane shear strain direction of the flanges i.e., mid-width at equal distances of 40 mm from one another while spanning a region from 150 mm to 270 mm away from the point of load application.

Apart from the reason mentioned earlier, that is the ability to take advantage of the locality of the effects of damage, this configuration was chosen for an additional reason as well; namely, the fact that strain gages, and therefore possible damage locations, were positioned near the point of load application, was decided because monitoring an area that displays higher strain levels was expected to contribute to the limiting of the influence of noise on actual measurements. Finally, to provide better understanding of the pattern of sensors employed, Figure 5.3 is given below showing a schematic of it while in Figure 5.4, its implementation on the actual test specimen is shown.

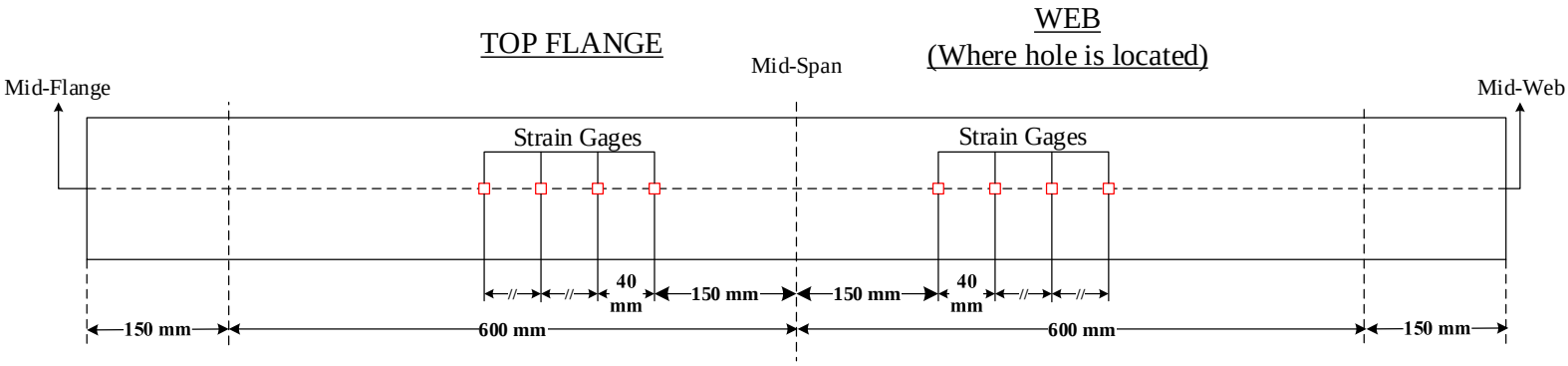


Figure 5.3: Schematic of Strain Gage Placement

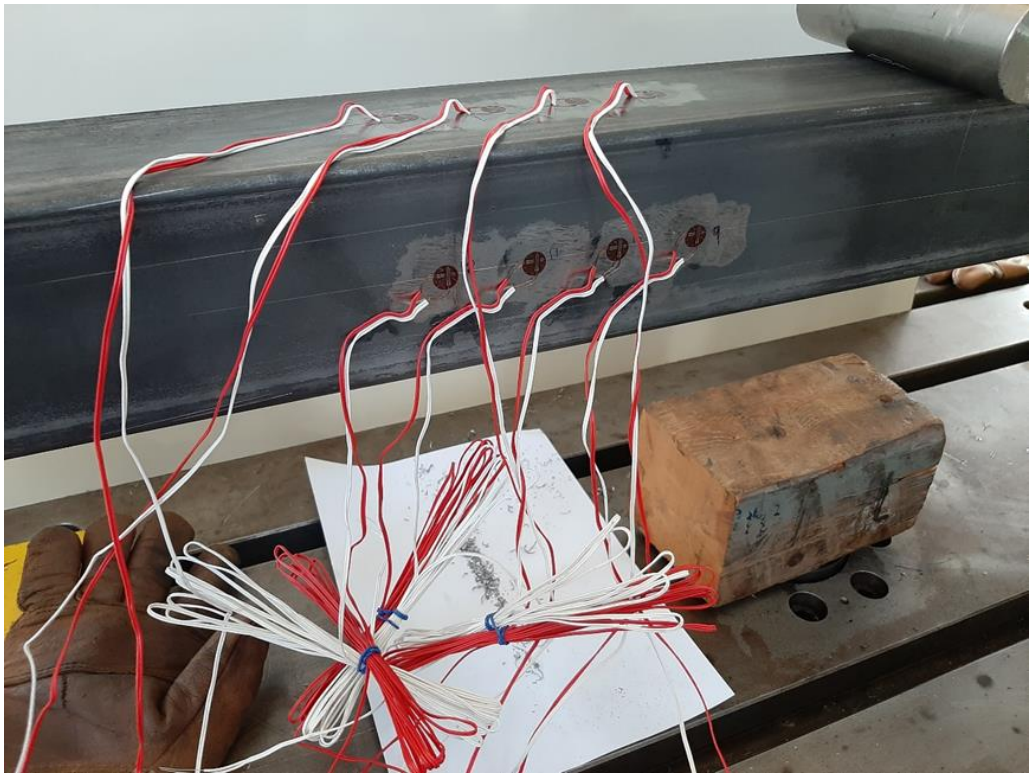


Figure 5.4: Strain Gage Placement on Actual Test Specimen

As was mentioned earlier, the strain gages used were rosettes with a 90-degree angle between the two gages. Therefore, their alignment when placed on the specimen had to be considered to make sure that the correct strains were measured. In the case of the webs, where axial strain must be monitored, this can be easily accomplished by simply

making sure that one of the two strain gages on the rosette is aligned with the longitudinal axis. In the case of the flanges however, where in-plane shear has to be monitored along the zero-shear strain direction, the rosettes have to be mounted along this axis with their two gages located at an angle to it. In this case, they were located at a 45-degree angle, as it leads to the easiest possible calculation of shear strain from the recorded strains; namely, in this case shear strain is calculated merely by subtracting the recorded strains ε_1 and ε_2. In Figure 5.5 below, this alignment is shown on the actual specimen used in the laboratory tests.

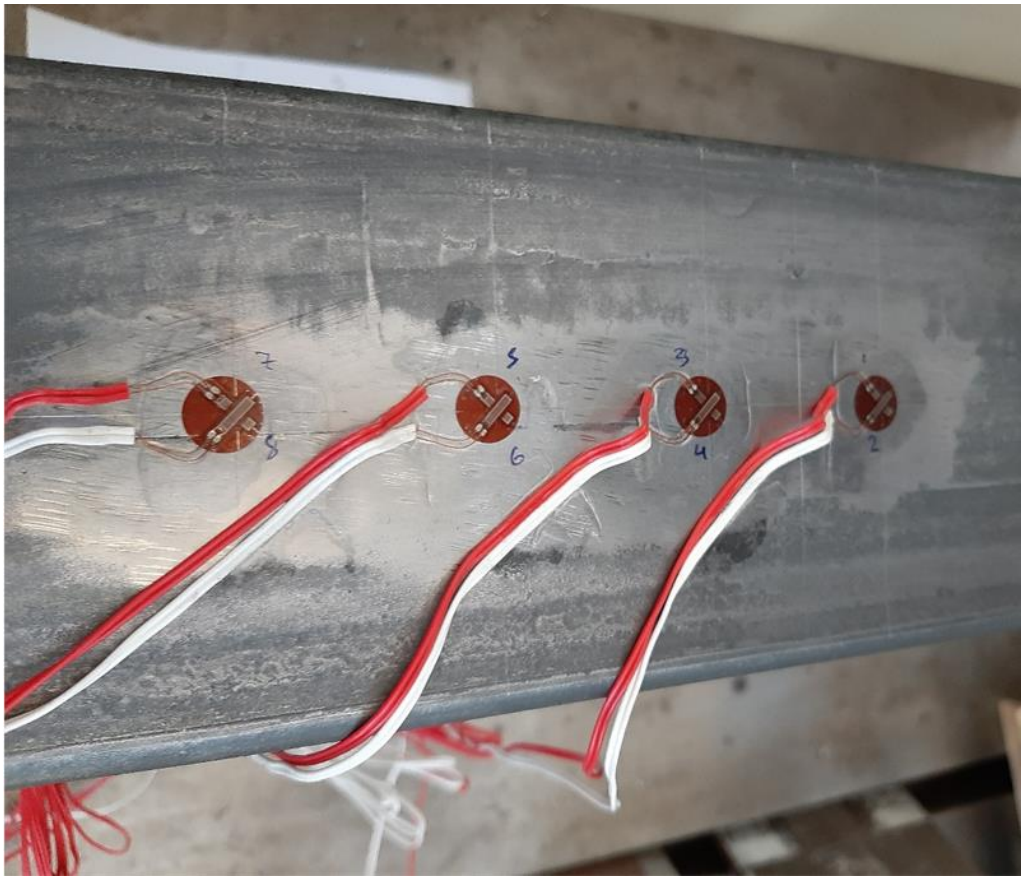


Figure 5.5: Detail of Strain Gage Alignment for Rosettes on Top Flange

At this point, having described the experimental set-up, the execution of the tests shall be described. Firstly, the girders were marked on the locations where the rollers would come in contact, as well as where the strain gages would be mounted. Then, the surface of the girder on those locations was smoothed and the strain gages were mounted. After the requisite amount of time elapsed for them to properly adhere, the girder was placed on the set-up of the universal testing machine, making sure that the distance between the rollers i.e., the span, was measured at 1200 mm, while the moving roller was mounted at precisely the middle of the span. Then, force was gradually applied until it reached 19 kN, a value decided in advance after the relevant calculations were performed to ensure that no local buckling might occur on the webs or flanges, and it was held there for the duration of the test. The specimen was held in that state for an

appropriate time interval, so as to obtain stable strain readings representative of the pristine specimen.

Then, the hole drilling procedure began. Its possible locations where limited in advance to the region covered by the strain gages, in order to maintain consistency with the digital twin. Its drilling was not done all at once at a final specified diameter, rather it was gradually enlarged from a diameter of 3.5 mm to a final diameter of 10 mm. After the drilling of each hole, an adequate amount of time was allowed to elapse for the vibration created by the drilling to subside and then the process was repeated. In total two test specimens were used and in each specimen two tests were carried out. In each test, a hole was opened in a different location, one on a flange and the other on a web, and specifically the web on which the strain gages were mounted. In Table 5.1 below, an overview of the locations of the various damage scenarios is provided for reference.

Table 5.1: Damage Locations for Laboratory Tests

		Test Specimen 1		Test Specimen 2	
		Case 1	Case 2	Case 1	Case 2
Face		Left Web (4)	Top Flange (1)	Top Flange (1)	Left Web (4)
Span		Left Span (2)	Left Span (2)	Left Span (2)	Left Span (2)
Longitudinal Location (from Mid-Span)	[mm]	170	250	170	250
Transverse Location (from Mid-Height)	[mm]	23	-23	-23	23

The locations of damage scenarios were selected in such a way that for each specimen the two holes were located at different faces and as far as possible in terms of longitudinal location. The reason behind this was to make sure that the locality of the effects of damage on the strain field was taken into full advantage and the existence of one defect does not affect the local disturbance of the strain field caused by the presence of the other. This allowed for more tests to be performed on a given number of specimens. Also, for each case the hole was drilled at an equal distance from two neighboring strain gages in order to maximize their ability to capture the disturbance caused on the strain field by the presence of damage. Regarding the terminology used, the test specimens are denoted as Test Specimen 1 & 2 and for each specimen there are Cases 1 & 2 corresponding to each different hole drilled. Finally, additional schematics showing the various damage scenarios in a detailed way are provided in the next section when the strain measurements for each case are presented.

5.3 Laboratory Test Results

5.3.1 Results for Test Specimen 1

Case 1

The first test carried out was on Test Specimen 1 for the damage scenario of Case 1 presented in Table 5.1 and also shown in Figure 5.6 below to provide better reference. Also, Figure 5.7 is provided showing the hole drilled on the actual specimen, as photographed during the execution of the laboratory tests.

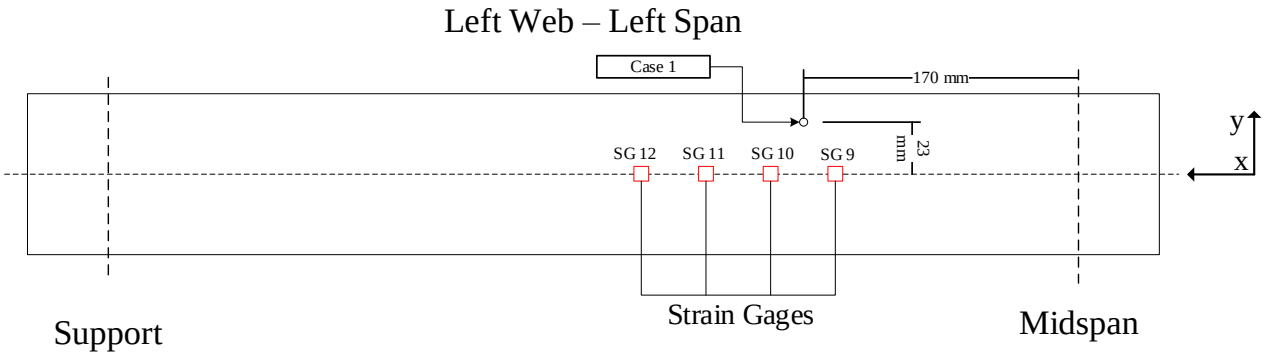


Figure 5.6: Test Specimen 1 / Case 1 Schematic

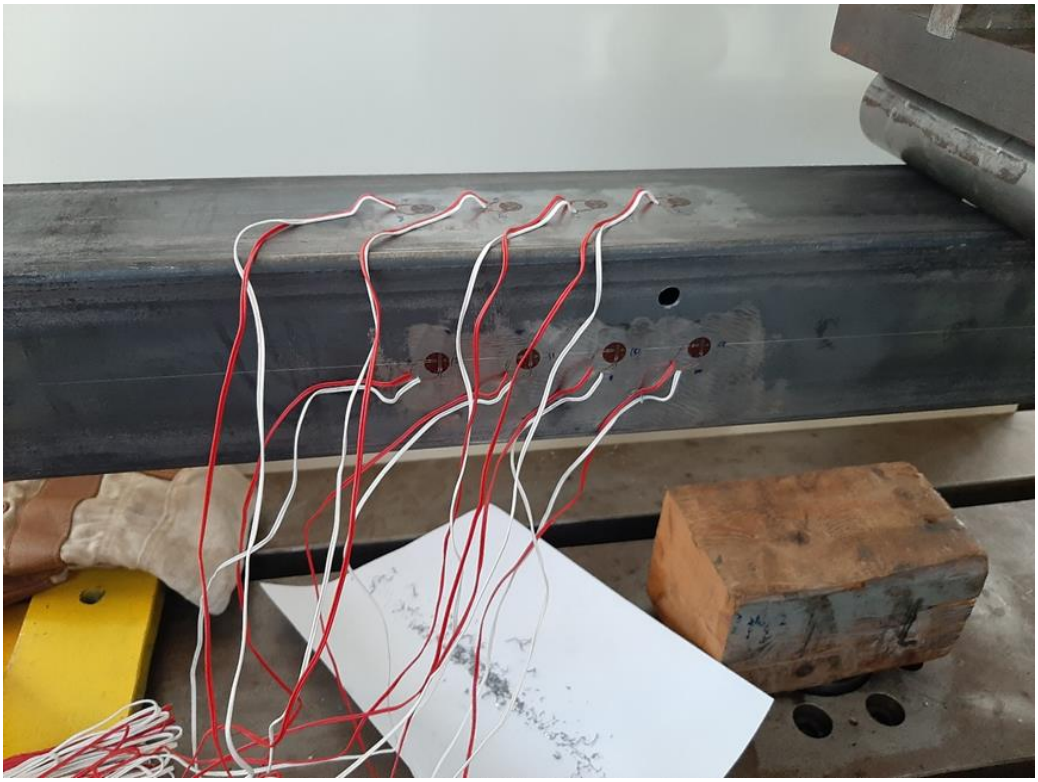


Figure 5.7: Test Specimen 1 / Case 1 in Actual Test Setting

For this case, strain distribution as a function of time elapsed during the experiment is presented as measured in both web and flange strain gages. Specifically, regarding those measurements from the top flange, the ones shown correspond to shear strain and have been produced by subtracting strains measured from the two gages in each rosette,

as was previously described. The top flange in-plane shear strain results are shown in Figure 5.8 while axial strains are shown in Figure 5.9.

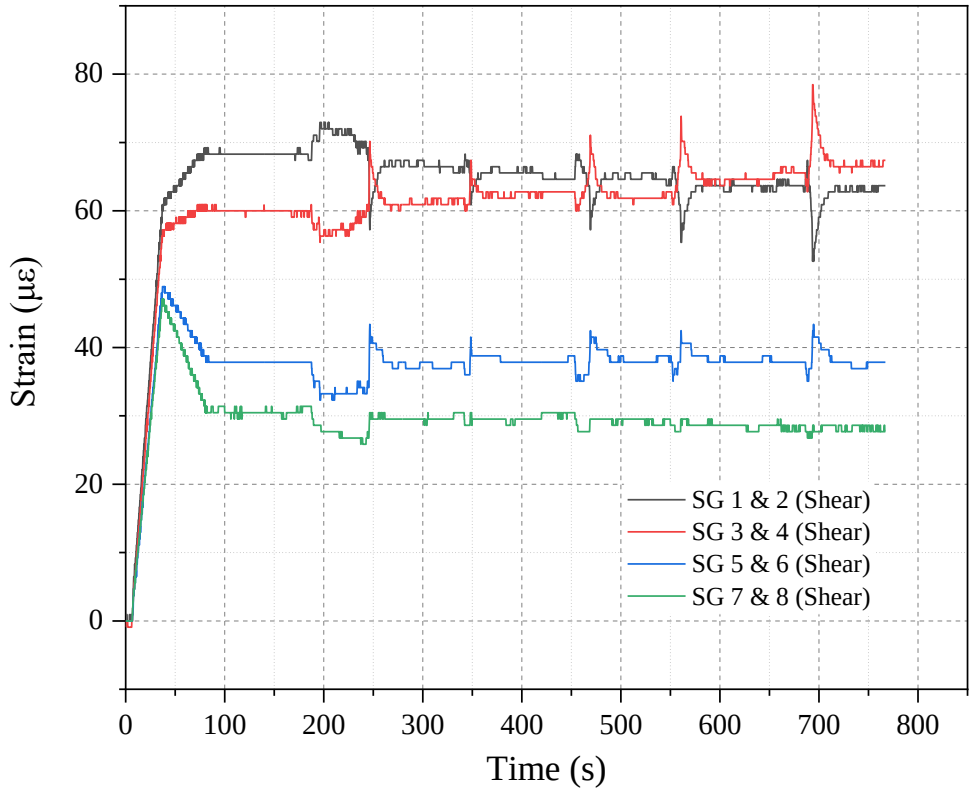


Figure 5.8: Test Specimen 1 / Case 1 - Top Flange Shear Strain Readings

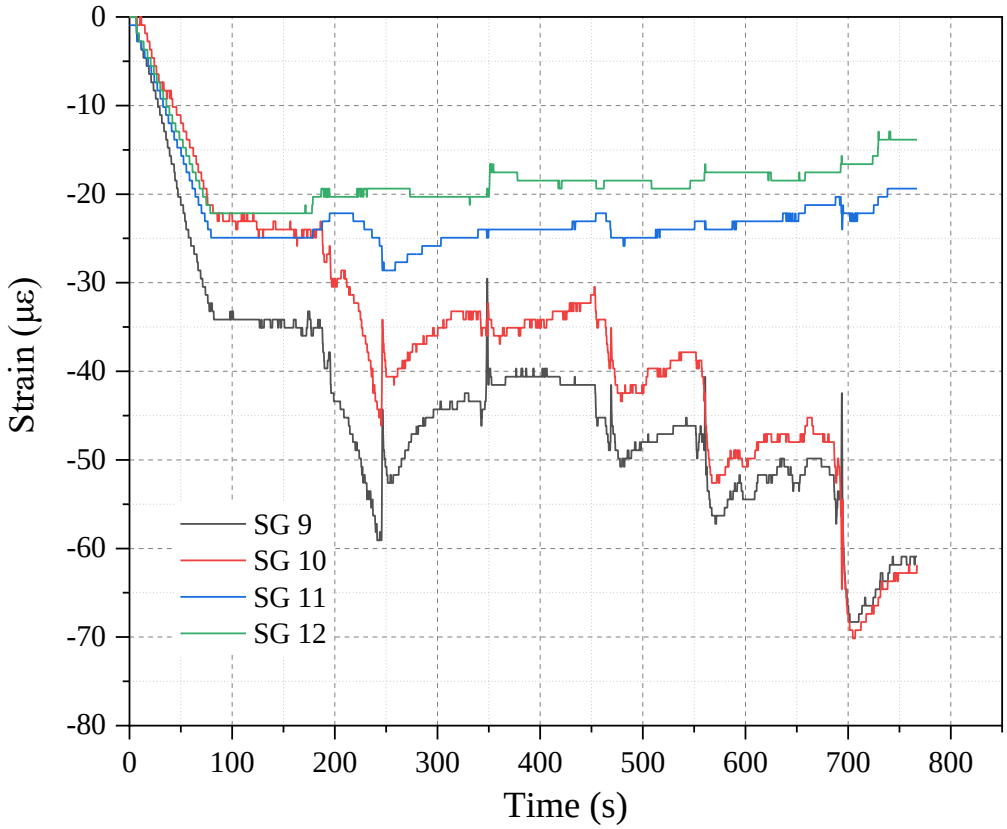


Figure 5.9: Test Specimen 1 / Case 1 – Web Axial Strain Readings

It is evident from Figure 5.8 that, although theoretically it was expected that shear strain would be zero along this direction, it actually was not. However, this was expected to be the case and can be attributed to possible misalignment of sensors as well as a host of other environmental factors. It is also shown that the vibrations caused by the drilling of the hole are "sensed" by the sensors nearest to it. This effect, however, did not carry on when the vibrations subsided, so the actual presence of the defect did not appear to have an effect on shear strain on the flange. In contrast, it is clearly shown in Figure 5.9 that the presence of the hole was established by the axial strain measured by both neighboring strain gages (SG 9 & SG 10).

This was further substantiated by the statistical results, presented in the form of a bar chart with error bars in Figure 5.10 below, where mean axial strain absolute values are presented factoring in an error of $\pm 3\sigma$, where σ is the standard deviation of strain measurements. Both the mean and standard deviation shown for the various hole diameters were calculated over the time intervals where the added noise due to the vibrations caused by drilling had subsided, which are clearly evident in the distributions presented above. The mean strains represent actual strain values for each particular case while the error bars offer an initial estimation of the random measurement noise, which may be used to calibrate the digital twin in order to use the tools developed for damage identification.

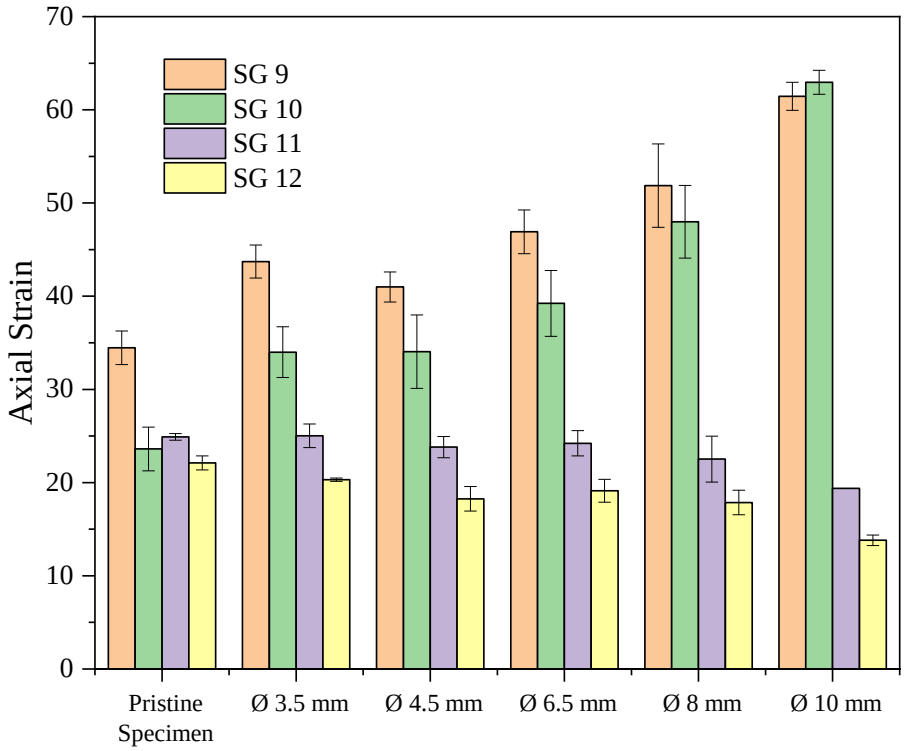


Figure 5.10: Test Specimen 1 / Case 1 - Web Strain Gage Axial Strain Results (Absolute Values)

Apart from the ability of the neighboring strain gages (SG 9 & 10) to accurately capture damage, which is further substantiated by this graph, some more observations can be made. Namely, that baseline strains for the strain gages i.e., strain readings corresponding to the pristine specimen, were similar for all strain gages apart from one

(SG 9). This had to be taken into account later when the data was modified in order to be used as input for the damage detection schemes. Also, the strain gages farthest from the hole remain relatively stable and within their baseline level values all along the duration of the test, indicating that they are unaffected by the presence of the defect and thus lending credence to the existing theoretical estimation about the local nature of damage.

Finally, another thing that can be observed was that significant disturbance in the strain field, as captured by the strain gages, began to appear for radii greater than 6.5 mm. This also proved what was expected from theory, that is that the magnitude of the defect would affect strain distribution proportionately. Finally, these results corroborate the claim made following observations on the numerical results obtained from the digital twin in Chapter 3.5.3, which indicated that disturbance in the strain field due to damage is limited to a narrow region in the immediate vicinity of the defect. Actually, experimental results showed that this region is even smaller than the one predicted based on the numerical models.

Case 2

The second test performed on Test Specimen 1 consisted of the drilling of a hole on a flange location. In Figure 5.11 below the relevant schematic is provided.

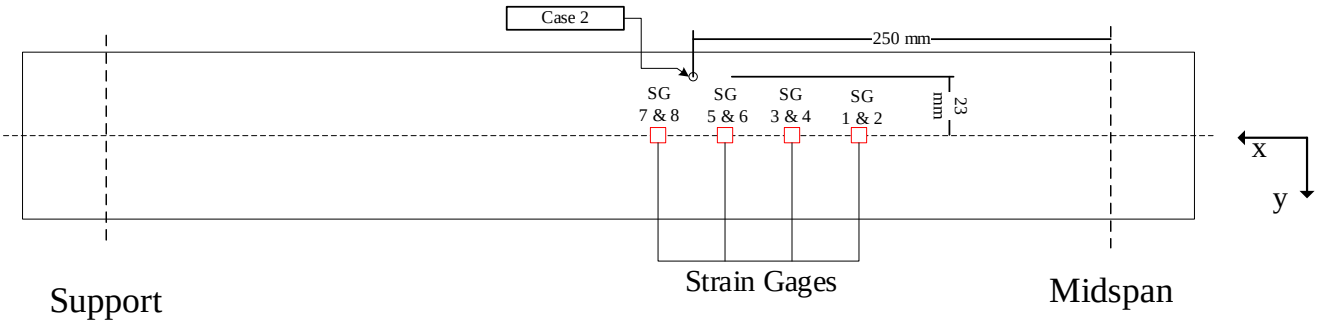


Figure 5.11: Test Specimen 1 / Case 2 Schematic

Also, the results obtained from the strain sensors are presented; in Figure 5.12 top flange in-plane shear strain readings are shown as a function of time elapsed during the test execution, obtained by subtracting measurements from the two gages in the rosette; in Figure 5.13 axial strain results are presented obtained from the strain gages mounted on the webs. It is evident that the presence of damage was detected only by the top flange sensors as expected and specifically, by the ones nearest to the hole (SGs 5-8). Also as expected, the existing defect from the previous test did not affect the distribution of axial strain on the web where it was located, as the measured strains remained constant at the levels reached during the previous test.

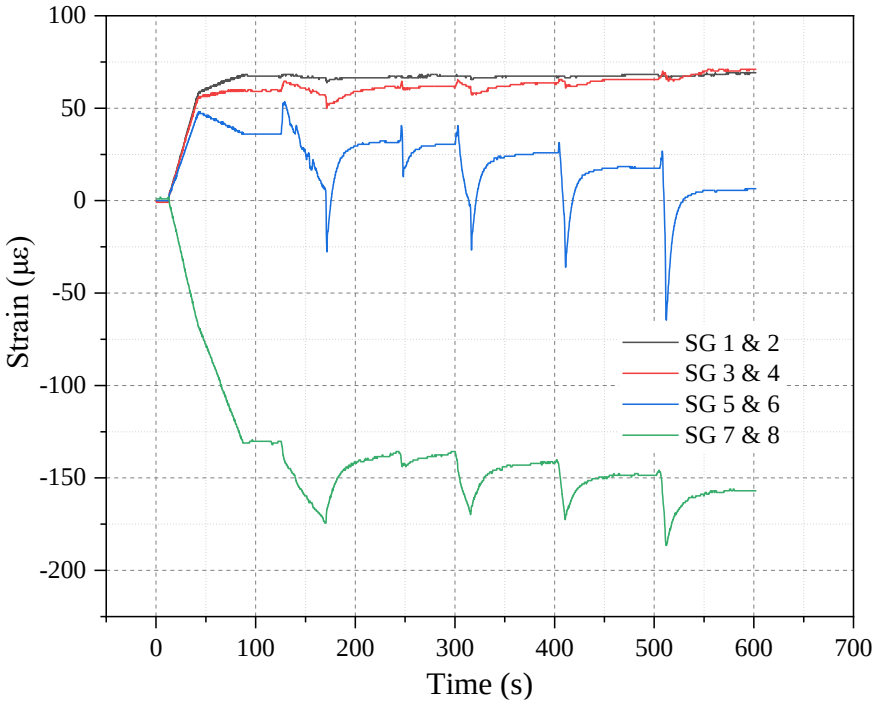


Figure 5.12: Test Specimen 1 / Case 2 - Top Flange Shear Strain

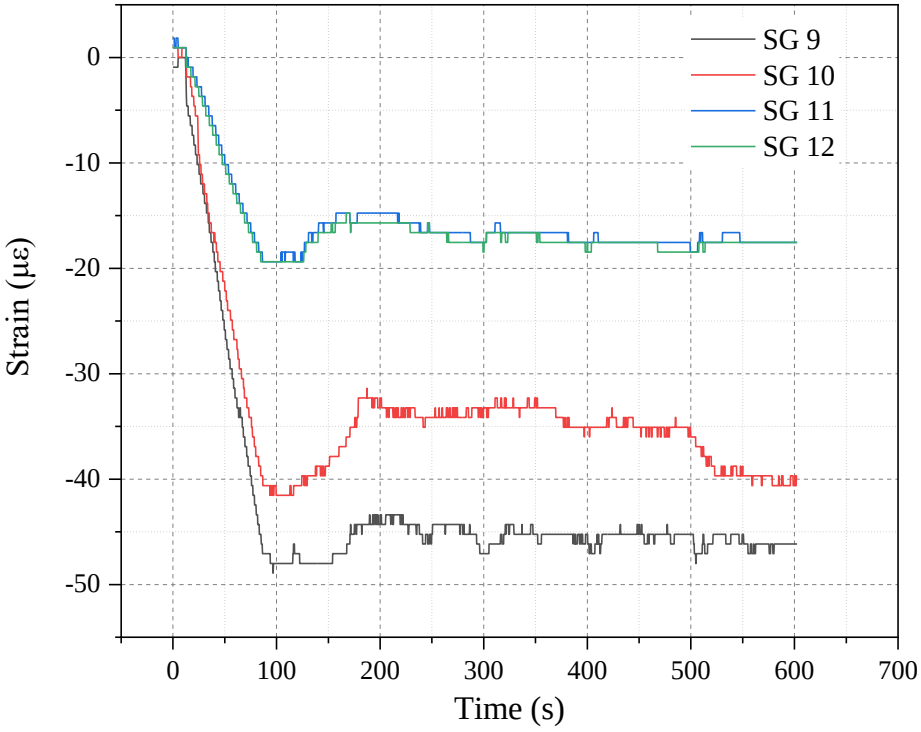


Figure 5.13: Test Specimen 1 / Case 2 - Web Axial Strain Readings

Analyzing the results from Case 2, several remarks can be made. It was again proven that a network of sensors more "tightly packed", may capture the presence of damage in more than one of its sensors, thus increasing the probability of successful detection. Also, the local effect of the defect in strain distribution was once again made clear, as the physics of the problem would dictate. Also, monitoring shear strain along the zero-strain direction on the top flange yielded promising results regarding its use as a damage indicator by clearly capturing the presence of damage, by way of the deviation in

measured strain. This is further substantiated in Figure 5.14 below, where a statistical representation of the results is presented, following the same paradigm as the one used in Case 1 previously. In-plane shear strain readings from strain gages mounted on the top flange are what is shown below, as these sensors capture the presence of damage successfully. The locality of the effect of damage on the strain field is displayed here even more clearly, as the strain gages farthest from the hole (i.e., SG 1-4) remain constant. On the contrary, in the results from the strain gages located in its immediate vicinity, a clear decreasing tendency is shown in strain results. At this point, it is important to note that the baseline strain of SGs 7&8 was different in this case, namely it was negative as evidenced by showing it in absolute value, compared to Case 1 while all others remain constant. This indicates that this sensor was polluted with noise between the two tests, whose source was not clear but may be attributed to a host of unforeseen factors. In any, case this was noted and taken into account later when these results were processed in order to be used as input in the damage detection schemes.

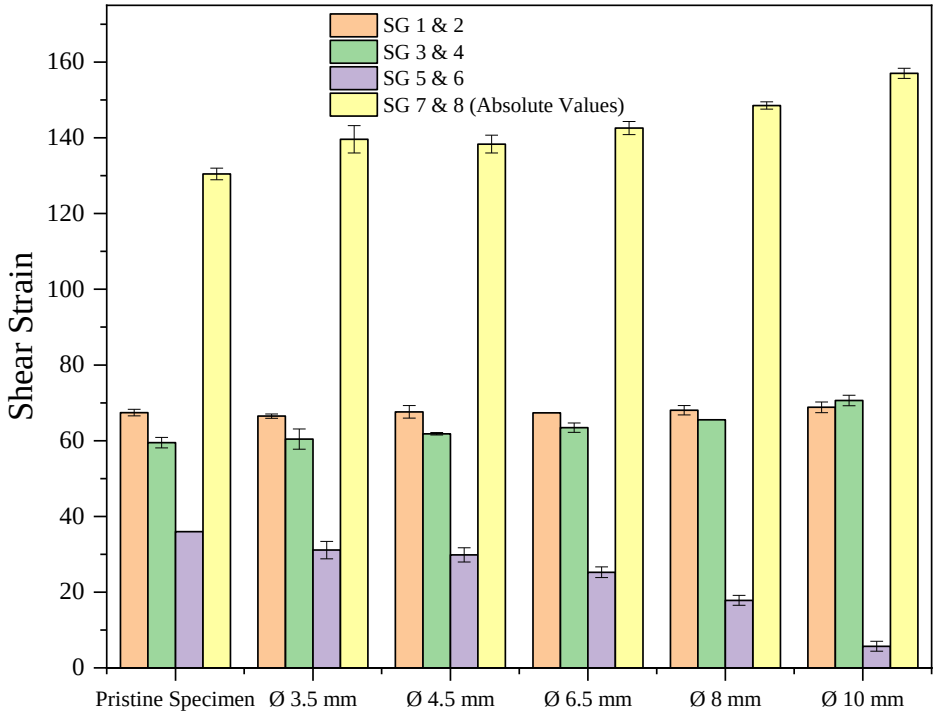


Figure 5.14: Test Specimen 1/ Case 2 – Top Flange Strain Gage Axial Strain Results

5.3.2 Results for Test Specimen 2

Case 1

After the tests on Test Specimen 1 were concluded, it was removed from the universal testing machine and Test Specimen 2 was mounted in its place. The same procedure was followed in order to set up the experiment. The specimen was aligned to the rollers using a laser and it was made sure that the stationary and moving rollers were all positioned at their specified locations. After it was made sure that the set-up was in order, the experiment began following the exact same steps; that is, force control was used in the machine to slowly apply the predefined force of 19 kN to the specimen,

while at the same time monitoring strain gage measurements in real time to make sure that no irregularities appear. After the target force was reached, the machine was held there for some time until static phenomena were clearly established. Then the drilling process began, again increasing the radius of the hole gradually and allowing for the necessary duration to elapse between openings, to ensure that strains estimated at each level would be free of the effects induced by the vibrations caused by the drill. A schematic of the hole drilled for Case 1 of Test Specimen 2, as referenced in Table 5.1, is provided in Figure 5.15 below while the actual specimen from the laboratory test is shown in Figure 5.16.

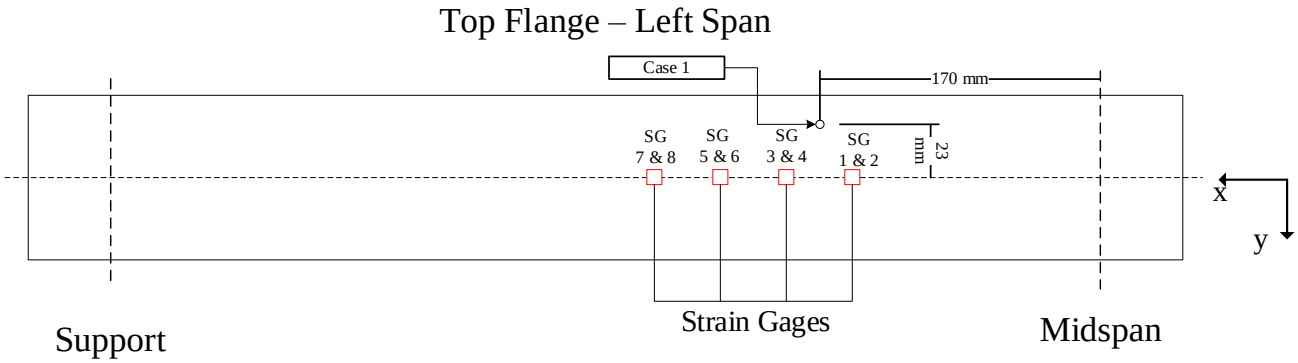


Figure 5.15: Test Specimen 2 / Case 1 Schematic

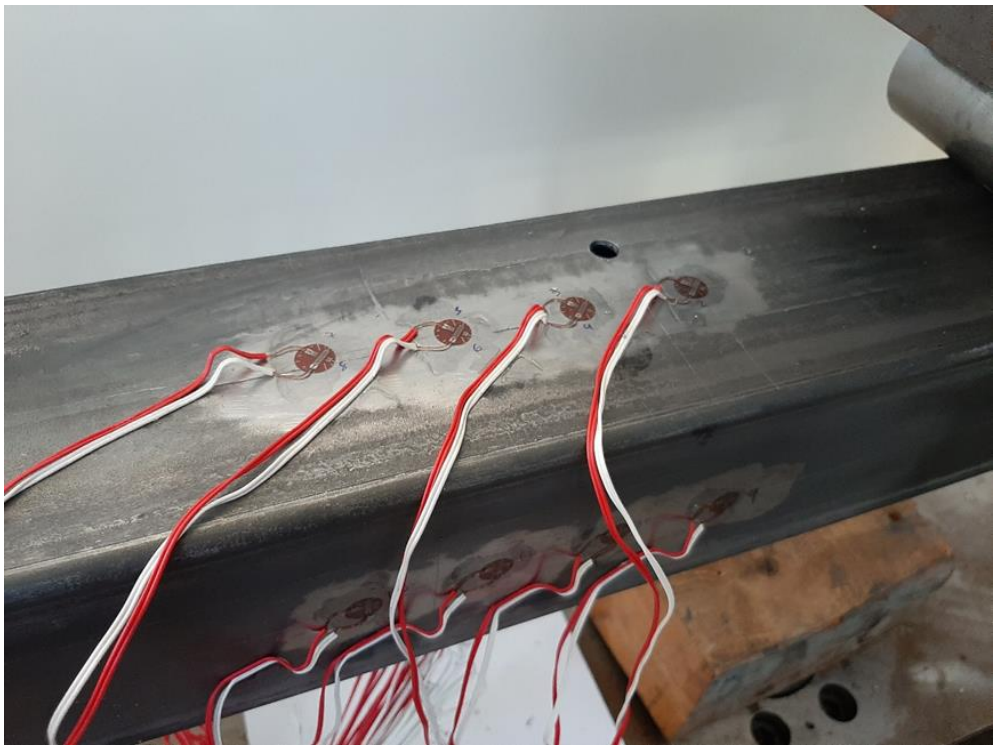


Figure 5.16: Test Specimen 2 / Case 1 in Actual Test Setting

In the following page, the strain distributions over time, captured by the mounted strain gages on the girder are presented in Figure 5.17 and Figure 5.18; Figure 5.19 contains a statistical representation of results from the top flange strain gages, where the hole was located.

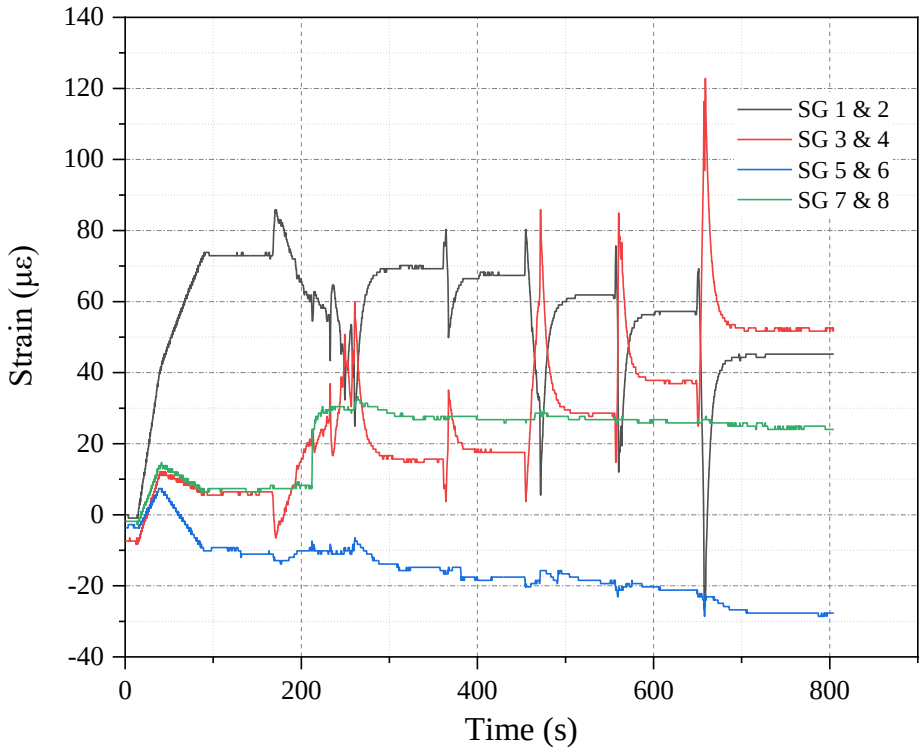


Figure 5.17: Test Specimen 2 / Case 1 - Top Flange Shear Strain

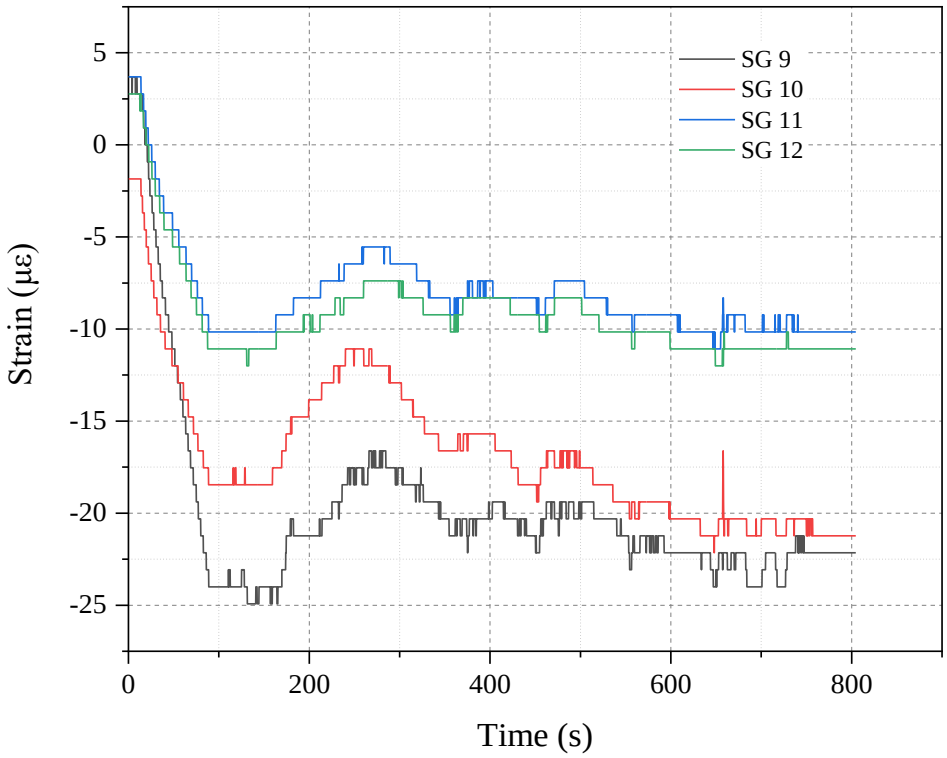


Figure 5.18: Test Specimen 2 / Case 1 - Web Axial Strain Readings

As expected, the strain gages located on the web measuring axial strain remain virtually unaffected by the presence of the hole, in the same manner as in the flange cases examined for the other specimen (see Figure 5.13). On the other hand, it is evident from Figure 5.17 that the strains measured at the nearest sensors to the hole i.e., SG 1 & 2 and SG 3 & 4, were clearly affected by its opening, as evidenced by the peaks indicating

the moments when it was drilled as well as the subsequent plateaus, all at different strain levels. Variation of strain was also recorded at the other two sensors, with the one recorded from SG 5 & 6, the nearest of the two, appearing to follow a pattern consistent to the timeline of the drilling. So, this may well be attributed to the presence of damage. However, it did have a different baseline from the other sensors, being negative and therefore shown in absolute value below, which had to be taken into account as an indicator of noise in the measurements. The farthest strain gage appears to be affected only by the initial drilling, as its readings remained consistent afterwards. This led to the assumption that it may have possibly picked up noise, especially considering that the initial drilling of the hole lasted for a significant amount of time. These observations made on the in-plane shear strain measurements, are substantiated further by the statistical representation of results shown in Figure 5.19 below, following the established error bar method of presentation detailed in the previous section.

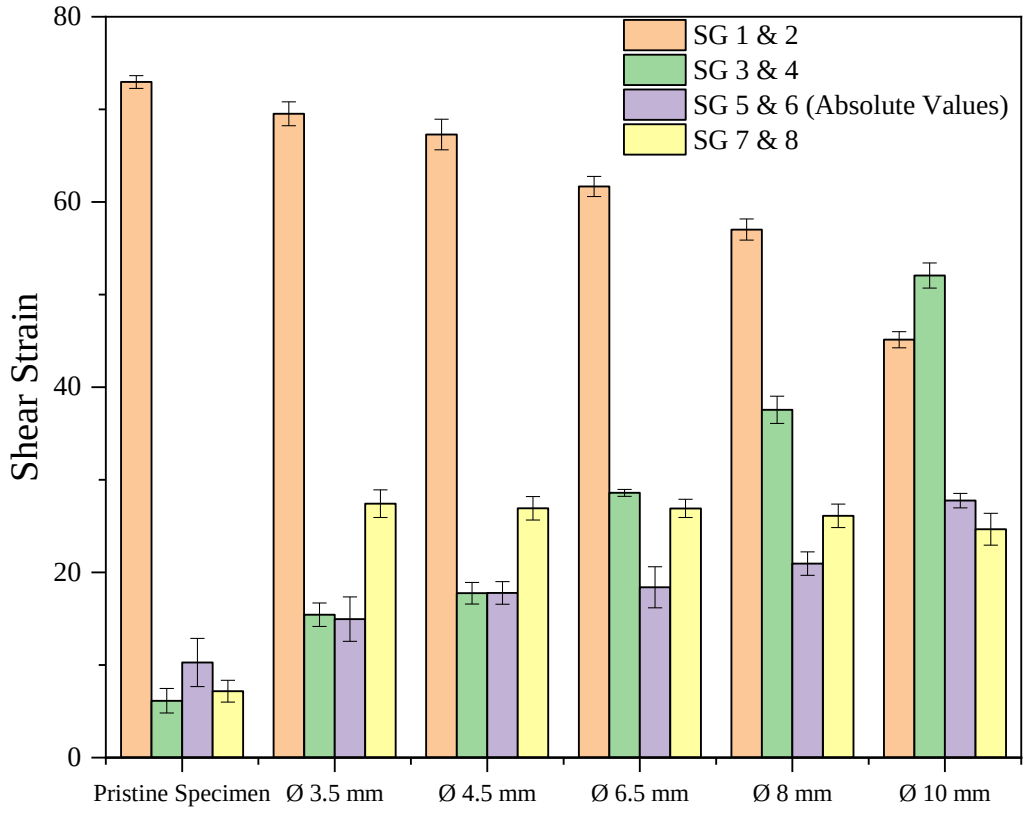


Figure 5.19: Test Specimen 2 / Case 1 – Top Flange Strain Gage Axial Strain Results

Case 2

Finally, the results obtained from the final experiment performed during the course of the laboratory tests carried out for the purposes of this thesis are presented here. Initially, a schematic of the damage scenario considered for this case is provided in Figure 5.20. Afterwards, the recorded strain values from the strain gages are presented: first for the top flange sensors in Figure 5.21 and then for the web-mounted strain gages measuring axial strain in Figure 5.22, which are expected to capture the damage present as the defect considered in this scenario is located on that web.

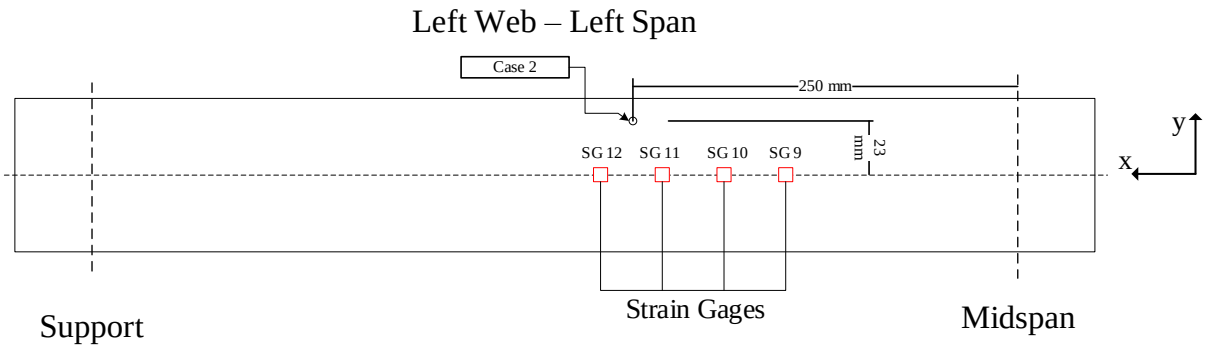


Figure 5.20: Test Specimen 2 / Case 2 Schematic

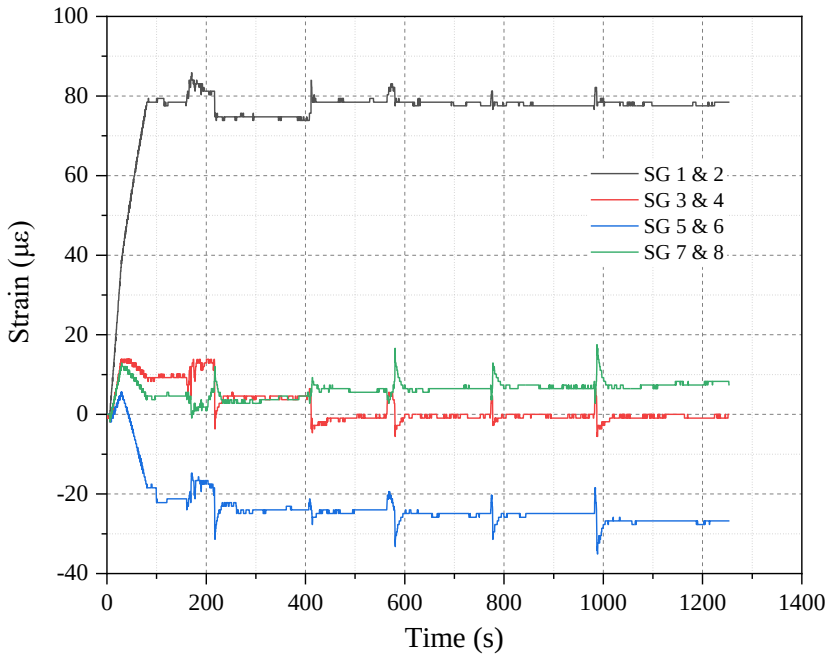


Figure 5.21: Test Specimen 2 / Case 2 - Top Flange Shear Strain

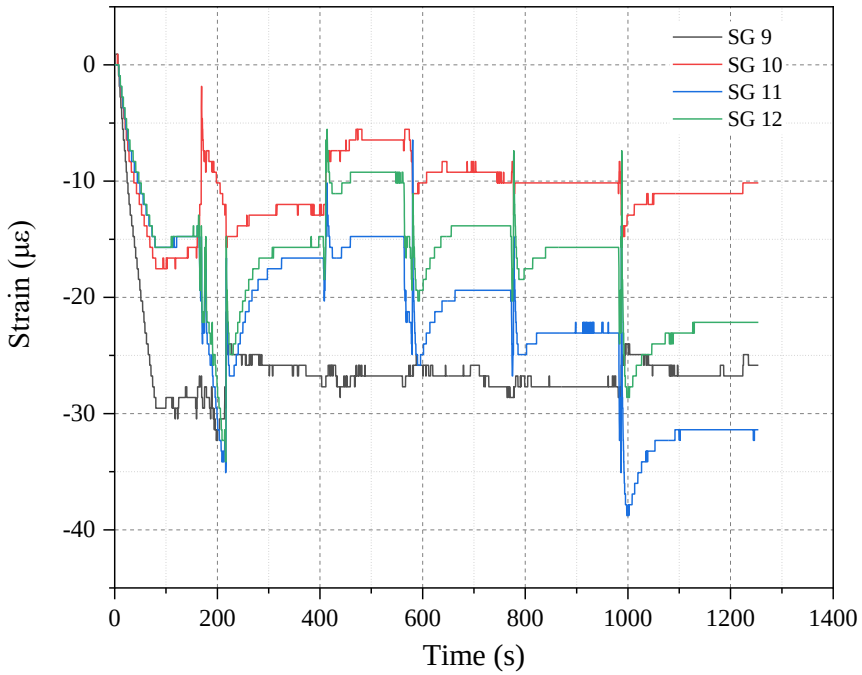


Figure 5.22: Test Specimen 2 / Case 2 - Web Axial Strain Readings

The assumption made earlier, that the results shown in Figure 5.22 representing axial strains would capture damage effectively was proven to be correct and is also showcased in the statistical representation of results presented in Figure 5.23, in the form of a column graph with error bars showing median strains (absolute values) with an error equal to $\pm 3\sigma$, where σ is the standard deviation. As mentioned earlier, these were calculated over time intervals corresponding to the different hole diameters where vibrations have subsided. Also, in-plane shear strains captured on the top flange shown in Figure 5.21 display certain peaks at the moments in time when the hole was drilled. However, the influence of the defect is limited to that in this distribution.

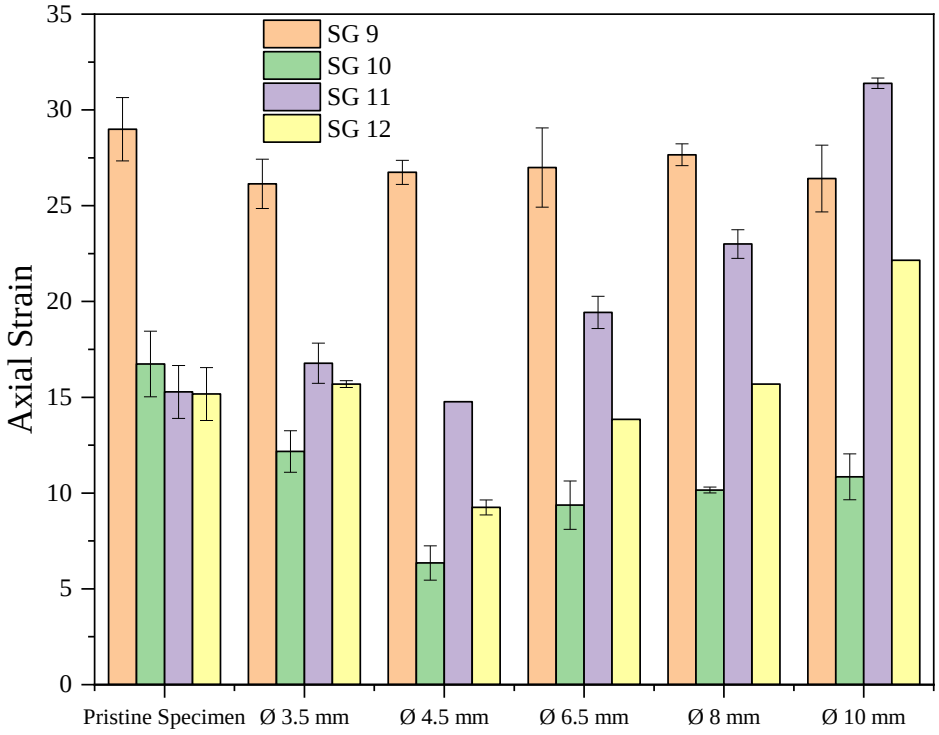


Figure 5.23: Test Specimen 2 / Case 2 - Web Strain Gage Axial Strain Results (Absolute

Once again, the hole appears to definitely affect strain distribution for the two strain gages nearest to it (SG 11 & SG 12). Also, as has been previously observed in the results of previous test cases, the third nearest strain gage (SG 10) was also affected, albeit in a diminished fashion. Moreover, it was again observed that diameters greater than 6.5 mm led to more significant variations of strain.

5.3.3 Concluding Remarks on Laboratory Test Results

At this point, the key insights gained from the laboratory tests carried out are summarily presented. First, it was immediately made clear that zero strain directions practically do not exist in a real-life testing scenario. Misalignment of sensors, environmental conditions, residual stresses due to the fabrication of the specimens as well as a host of other factors affect the readings, making necessary the establishment of a baseline for each sensor in each test setting, to be able to use the actual results in the numerical

environment. This highlights the need to calibrate the numerical model according to the physical measurements, if a truly data-enabled digital twin is to be created.

With regards to the physical meaning of the resulting measurements, it was established that the effect of a hole on the measured strain is locally confined. Specifically, it is confined to the face where the hole is located and also, to its immediate vicinity. That necessitates the placement of a sensor, or a network thereof, in the close "neighborhood" of the defect in order to more probably detect it. Also, the location of the defect -and therefore the sensor network- in high stress areas is a significant contributor to an increased probability of detection, because it leads to more accentuated variations of strain that both facilitate easier detection and "drown" out any added noise.

5.4 Validation of Damage Identification Schemes with Experimental Data

5.4.1 Implementation Framework

At this point, the experimental results discussed in the previous sections were processed accordingly in order to be "fed" to the damage identification methods, detailed in Chapters 3 & 4 previously. It became evident, as was expected based on the author's understanding of the nature of experimental measurements, that even though theoretically strain was monitored across zero-strain directions some strain was still measured. However, this was consistently at very low levels and thus is attributed to noise pollution and other factors that affect measurements when strain gages are used to that effect. Therefore, these levels were considered as the sensor's baseline and were subtracted from the measurements obtained for a specific hole radius. In this case, this was selected to be 4 mm, as at this level it was evident that the effects of damage were clearly pronounced.

Also, it was decided to follow two separate approaches regarding the implementation of the damage detection schemes on the experimental data. Namely, in the first one it was decided to use a variation of the digital twin detailed in Chapter 3 earlier. Specifically, it was modified to exactly simulate the physical characteristics of the test specimen and the conditions of the test and also, the post-processing section of the APDL code was amended so as to replicate the experimental data acquisition scheme; that way, an exact virtual copy of the test specimen was produced. For this approach only the GA-based method was used as the limited number of sensors and their localized placement were not deemed suitable to produce a dataset capable of successfully training the neural networks used in the ANN-based method developed. The scope of this initial approach was to obtain an initial estimate of the strongest test "candidates", i.e. the specific tests where the method produced successful results, which would then be used in the second, more holistic approach.

In this approach both methods will be utilized, specifically in their 16-sensor configurations which allow for the use of the experimental data, as the 16-sensor paths include the locations of the strain gages used in the actual experiments. The remaining (virtual) sensor values will be substituted in the target vectors with data obtained from the digital twin of the test specimen, with noise added to them in the same levels as those established from the experimental measurements. These levels were determined using basic tools of statistics which will be expanded upon in the relevant section.

Additionally, the FE model at the core of the digital twin used in this application was redesigned to match the test specimens used, with regard to both physical and material properties as well as applied loads. Namely, the imposed boundary condition was a displacement which caused an equivalent structural force to the load applied by the universal testing machine i.e., 19 kN. The APDL code producing this model was used in the GA-based approach and also, that same model was used to retrain the relevant neural networks featured in ANN-based approach using exactly the same procedure and parameters mentioned in detail in Chapters 3 & 4. Finally, it should be noted that the reason behind implementing this approach was that it was believed that it constitutes a truly data-enabled digital twin and is indicative of the formulation that would be used in a real-world application of a structural digital twin.

5.4.2 Implementation on Reduced Sensor Digital Twin

In this section, the results are provided that were obtained using the reduced sensor digital twin GA-based approach, directly simulating the experimental set-up. For every case, a table showing the actual versus the predicted characteristics of damage is given and afterwards the outcomes are discussed. It should be noted that the terms used to describe each case are the same used earlier in this chapter (see Table 5.1).

Table 5.2: Damage Detection Results for Test Specimen 1 / Case 1

		Actual	Predicted
Face		Left Web (4)	Left Web (4)
Span		Left Span (2)	Left Span (2)
Longitudinal Location (from Mid-Span)	mm	170	175
Transverse Location (from Mid-Height)	mm	23	23

Table 5.3: Damage Detection Results for Test Specimen 1 / Case 2

		Actual	Predicted
Face		Top Flange (1)	Bottom Flange (3)
Span		Left Span (2)	Left Span (2)
Longitudinal Location (from Mid-Span)	mm	250	160
Transverse Location (from Mid-Height)	mm	-23	23

Table 5.4: Damage Detection Results for Test Specimen 2 / Case 1

		Actual	Predicted
Face		Top Flange (1)	Top Flange (1)
Span		Left Span (2)	Left Span (2)
Longitudinal Location (from Mid-Span)	mm	170	275
Transverse Location (from Mid-Height)	mm	-23	-23

Table 5.5: Damage Detection Results for Test Specimen 2 / Case 2

		Actual	Predicted
Face		Left Web (4)	Top Flange (1)
Span		Left Span (2)	Left Span (2)
Longitudinal Location (from Mid-Span)	mm	250	215
Transverse Location (from Mid-Height)	mm	23	-23

As evidenced by the tables presented above, successful detection was achieved in two of the four cases, namely Case 1 for both test specimens. In the other two cases, while the girder's span was successfully predicted, the same was not the case for the face on which damage was present. However, to consider detection successful in the context of this work, at least both span and face have to be classified correctly. Thus, they were considered to be instances where detection was not achieved.

The two successful cases represent damages located in both the web and the top flange. What is common between them is that in both cases the holes were drilled at the same longitudinal location. Also, of the cases examined these were closest to the point of load application, and thus subjected to higher stress and strain levels. Their successful detection lends credence to the hypothesis made initially, that defects located at high stress areas were more likely to be detected, as measurement noise is "drowned".

Bar charts are also provided in Figure 5.24 which summarize the table results. It should also be mentioned that in each bar chart, the results for a specific identifier are presented. In the case of face and span, their numerical identifiers are used for illustrative purposes. These have been mentioned before repeatedly and are also provided in the tables below for reference.

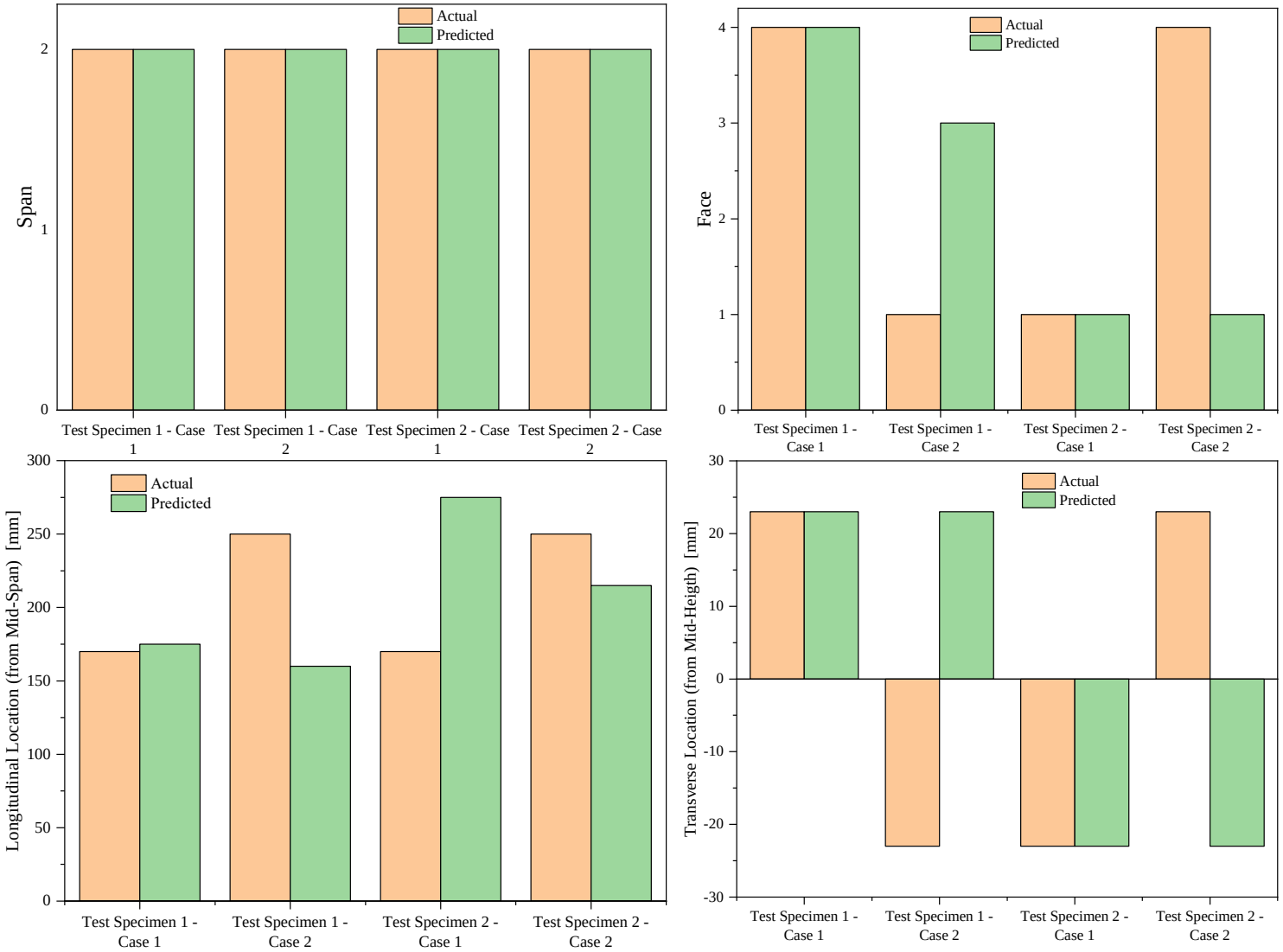


Figure 5.24: Summary of Damage Detection Results for Reduced Sensor Digital Twin

5.4.3 Implementation on Full-Sensor Digital Twin

At this point, the two successfully detected cases were further investigated. Specifically, the actual strain data from these cases were assimilated in the global damage detection schemes developed, specifically those that contain eight separate paths with sixteen sensors each. In order to assimilate the data realistically, first they were included in the appropriate sensor locations, replacing the data obtained from the digital twin in those same locations. Also, the rest of the digital twin strain data were polluted with noise to simulate the real-world conditions.

However, before considering the level of noise that should be added to the digital twin-obtained strain readings in order to use each case in the framework of the global methods, it was considered important to provide some information on how experimental results compare to the ones obtained in the virtual environment. The graphs provided correspond to the candidate tests and display the actual sensor readings compared to the numerical results plotted across the full strain monitoring path. The

reason why these are provided is to show what the deviation is between actual and virtual measurements and thus to provide grounds for discussion on the matter. Figure 5.25 and Figure 5.26 given below refer to Case 1 of Test Specimen 1.

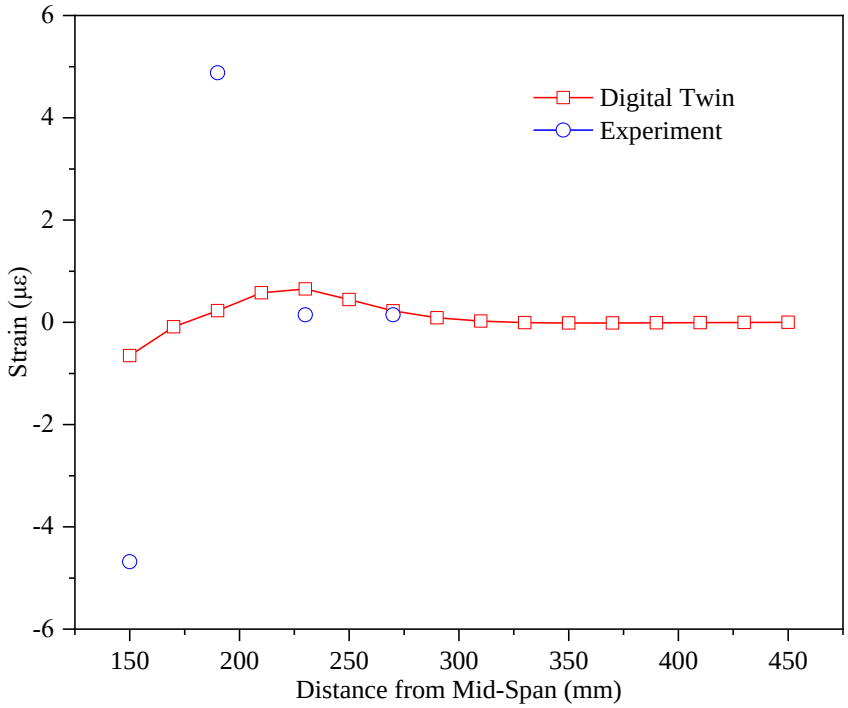


Figure 5.25: In-Plane Shear Strain Distribution Along Top Flange Sensors - Test Specimen 1 / Case 1

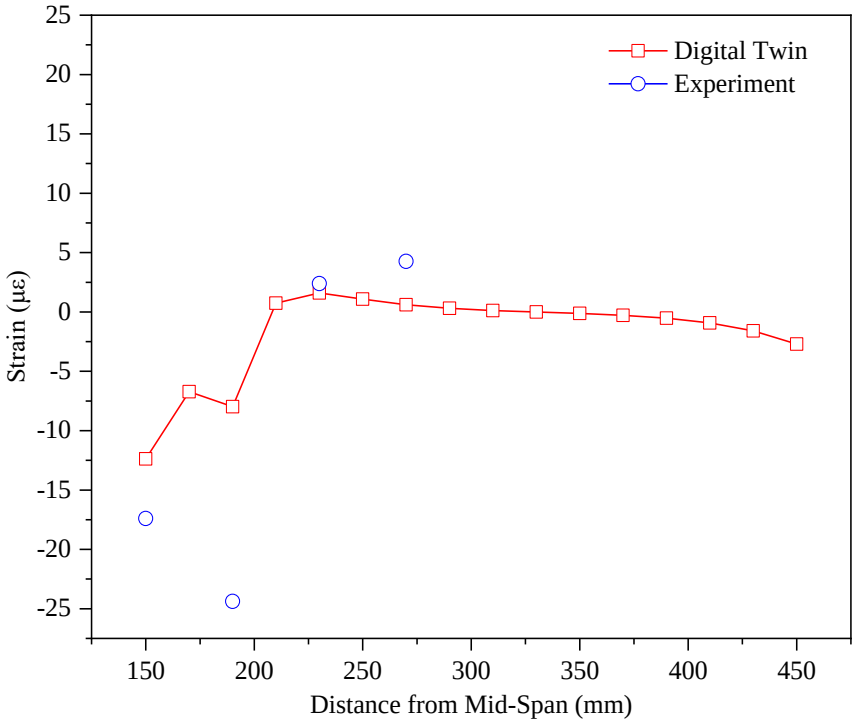


Figure 5.26: Axial Strain Distribution along Web Sensors - Test Specimen 1 / Case 1

Also, the same comparative graphs are given below, corresponding to Case 1 of Test Specimen 2. Specifically, in Figure 5.27 in-plane shear strain readings are compared while in Figure 5.28 what is compared is axial strains from web sensors.

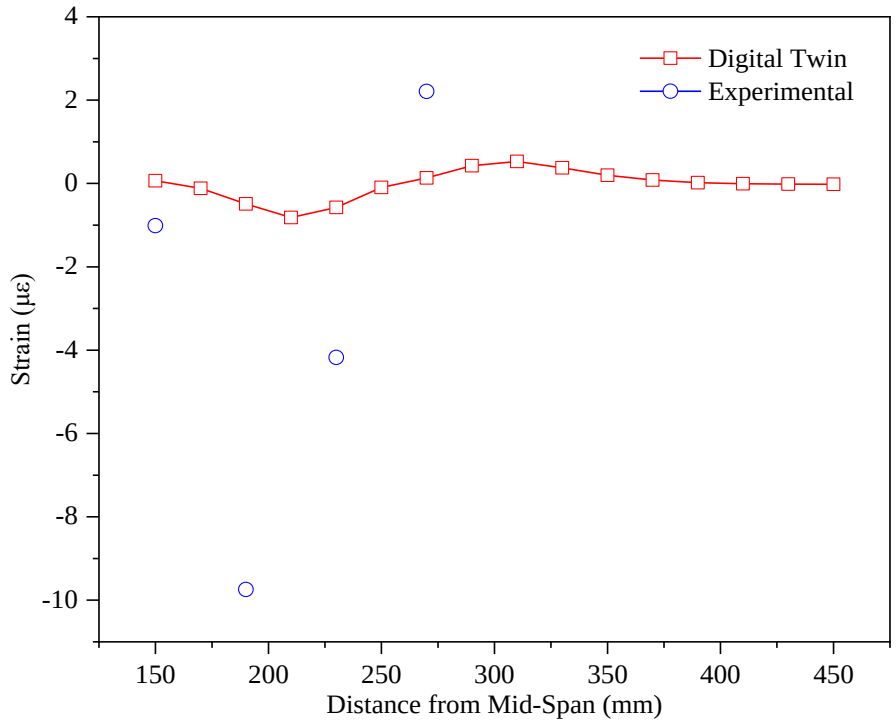


Figure 5.27: In-Plane Shear Strain Distribution along Top Flange Sensors - Test Specimen 2 / Case 1

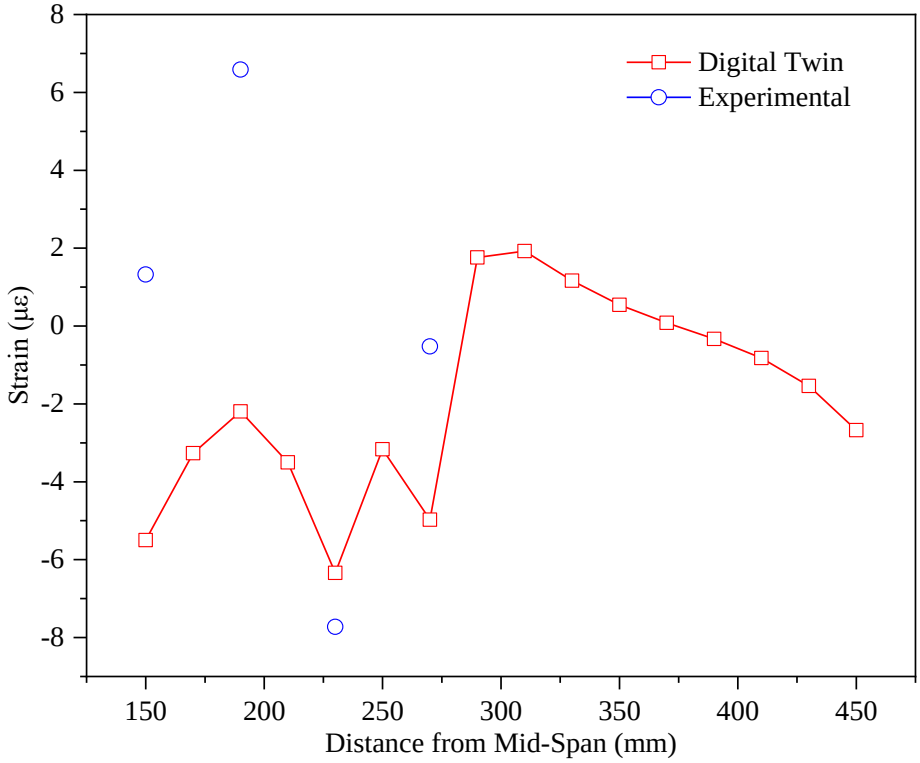


Figure 5.28: Axial Strain Distribution along Web Sensors - Test Specimen 2 / Case 1

The results presented above indicated significant deviation between the digital twin-obtained FE results and the actual measurements. This was significantly more pronounced when it came to the results obtained from the top flange strain gages where in-plane shear strain was monitored. These large deviations were mainly attributed to two reasons. First of all, the possibility that the rosettes located on the top flange were not aligned exactly at a 45° angle with regard to the flange's longitudinal axis of symmetry. To corroborate this claim the following two tables are provided: Table 5.6 containing individual strain readings from all top flange channels, with no subtraction of results to obtain shear having taken place, showing actual results measured on site compared to digital-twin obtained ones; and Table 5.7 comparing actual strain results to results obtained from the digital twin, after 1° of misalignment has been artificially added to the results. Both tables refer to Case 1 of Test Specimen 1 and no additional information is provided on the other candidate as the obtained results followed a similar pattern.

Table 5.6: Comparison between Actual and Virtual Individual Top Flange Strain Gage Results

	Strain (με)							
	SG 1 & 2		SG 3 & 4		SG 5 & 6		SG 7 & 8	
	+ 45°	- 45°	+ 45°	- 45°	+ 45°	- 45°	+ 45°	- 45°
Measured	-185.52	-249.17	-163.50	-228.36	-161.34	-199.33	-150.90	-179.33
Digital Twin	-204.61	-205.26	-186.71	-186.48	-168.68	-168.03	-150.13	-149.90
% Difference	10.29%	17.62%	14.19%	18.34%	4.55%	15.70%	0.51%	16.41%

Table 5.7: Comparison between Actual and Misaligned Virtual Individual Top Flange Strain Gage Results

	Digital Twin Results with +1° Divergence in Rosette Alignment							
	Strain (με)							
	SG 1 & 2		SG 3 & 4		SG 5 & 6		SG 7 & 8	
	+ 44°	- 46°	+ 44°	- 46°	+ 44°	- 46°	+ 44°	- 46°
Measured	-185.52	-249.17	-163.50	-228.36	-161.34	-199.33	-150.90	-179.33
Digital Twin	-191.56	-218.31	-174.69	-198.5	-157.79	-178.92	-140.4	-159.63
% Difference	3.26%	12.39%	6.84%	13.07%	2.20%	10.24%	6.96%	10.98%

What became apparent from the above was that results between numerical and actual cases converged when misalignment of 1° is factored in. Such a small level was possible to have occurred and escaped our attention in the testing phase. Now, the second main reason to which these deviations were attributed, which is also valid for the axial strain measurements had to do with the baseline values of each sensor. It was explained earlier that although theoretically on zero-strain directions, the strain gages measure some strain even when the specimen is in its undamaged state. This level of strain constitutes the sensor's baseline strain. As shown in the results provided in Chapter 5.3, these baseline values vary in an unpredictable manner. On the other hand, the change in

measured strain captured in each sensor due to the opening of the hole follows a standard pattern. Also, the magnitude of this change is relatively small in the case considered. These two factors coupled with the inconsistency of baseline values explain why the disturbance in the strain field due to the presence of damage displays this inconsistency between the actual and numerical experiments.

Although the above established that there is some deviation between actual and numerical data, which falls beyond the levels of random noise and is rather attributed to the reasons detailed earlier, the addition of noise to the digital twin data was decided to adhere to the criteria already mentioned. Therefore, to ascertain what a representative level of noise was for each case, histograms showing the statistical distribution of strain measurements for both the initial and the damaged specimens were produced. They were obtained over the range of data corresponding to plateaus in strain measurement, which were also used to calculate the mean values for each case as detailed in Chapter 5.3 before. That way, the range around the mean in which strain is distributed is illustrated and that was considered to constitute the effective range of noise, which was then added to the digital twin strain measurements. The following graphs show the aforementioned histograms, produced from data by sensors located on the face where the hole was located in both cases.

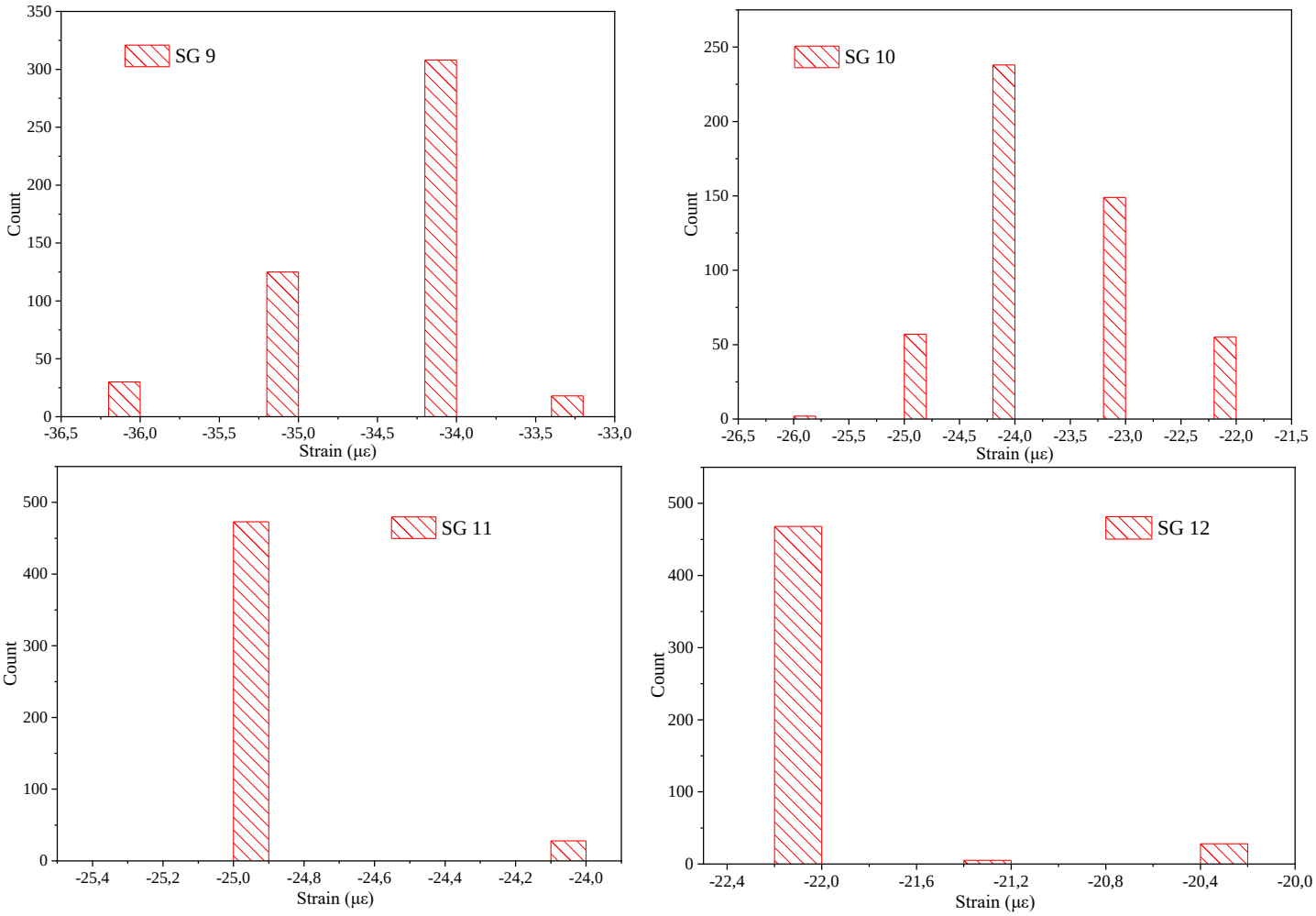


Figure 5.29: Test Specimen 1 / Case 1 – Web Axial Strain Distribution (Initial Specimen)

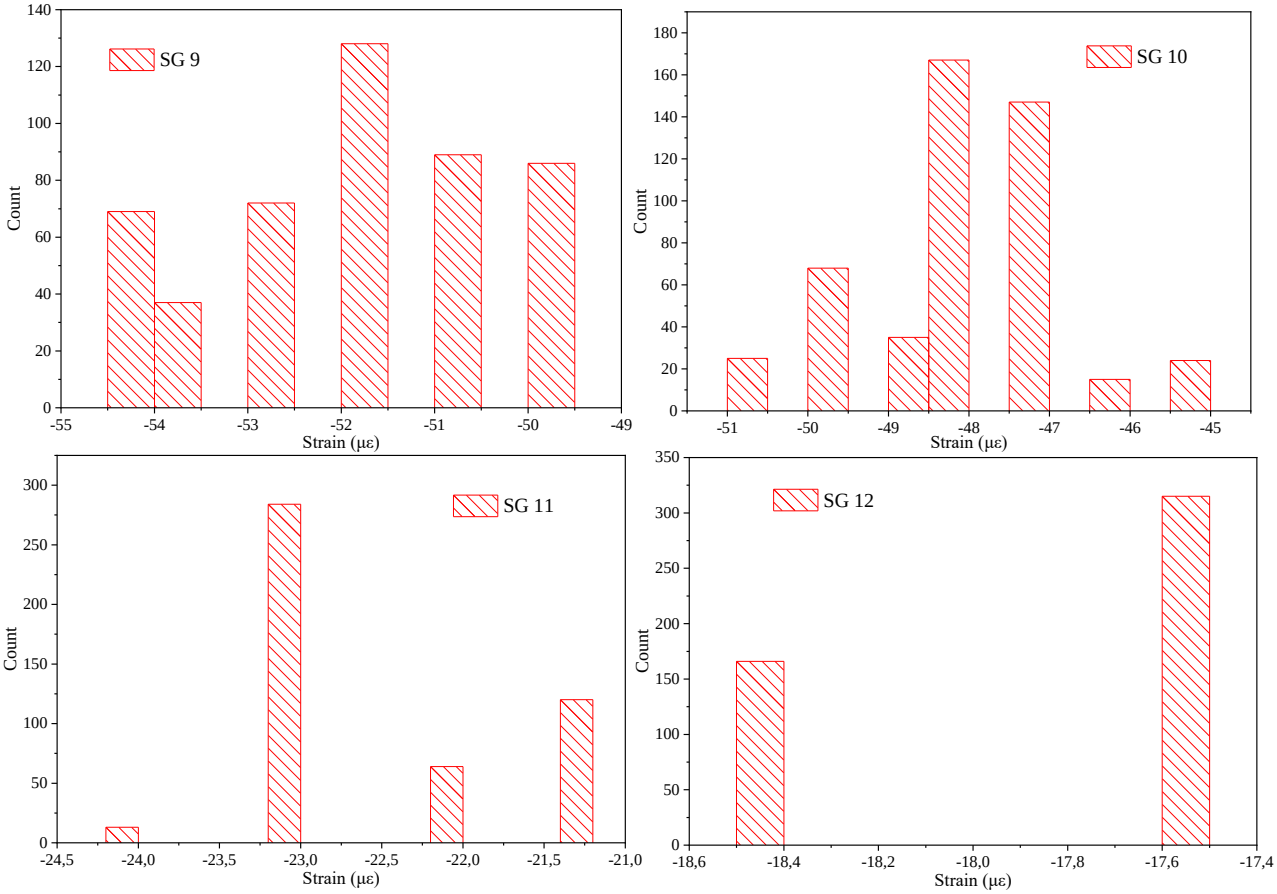


Figure 5.30: Test Specimen 1 / Case 1 – Web Axial Strain Distribution (Damaged Specimen)

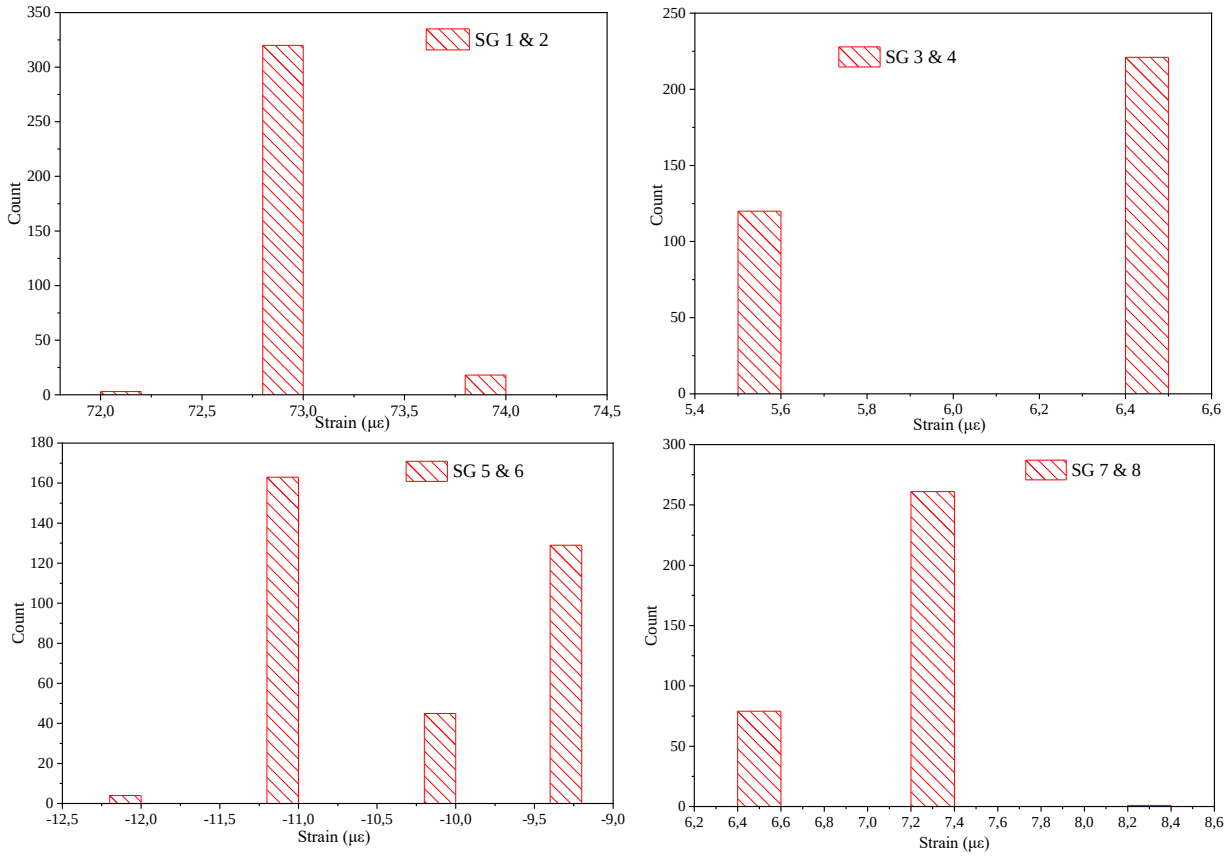


Figure 5.31: Test Specimen 2 / Case 1 - Top Flange Shear Strain Distribution (Initial Specimen)

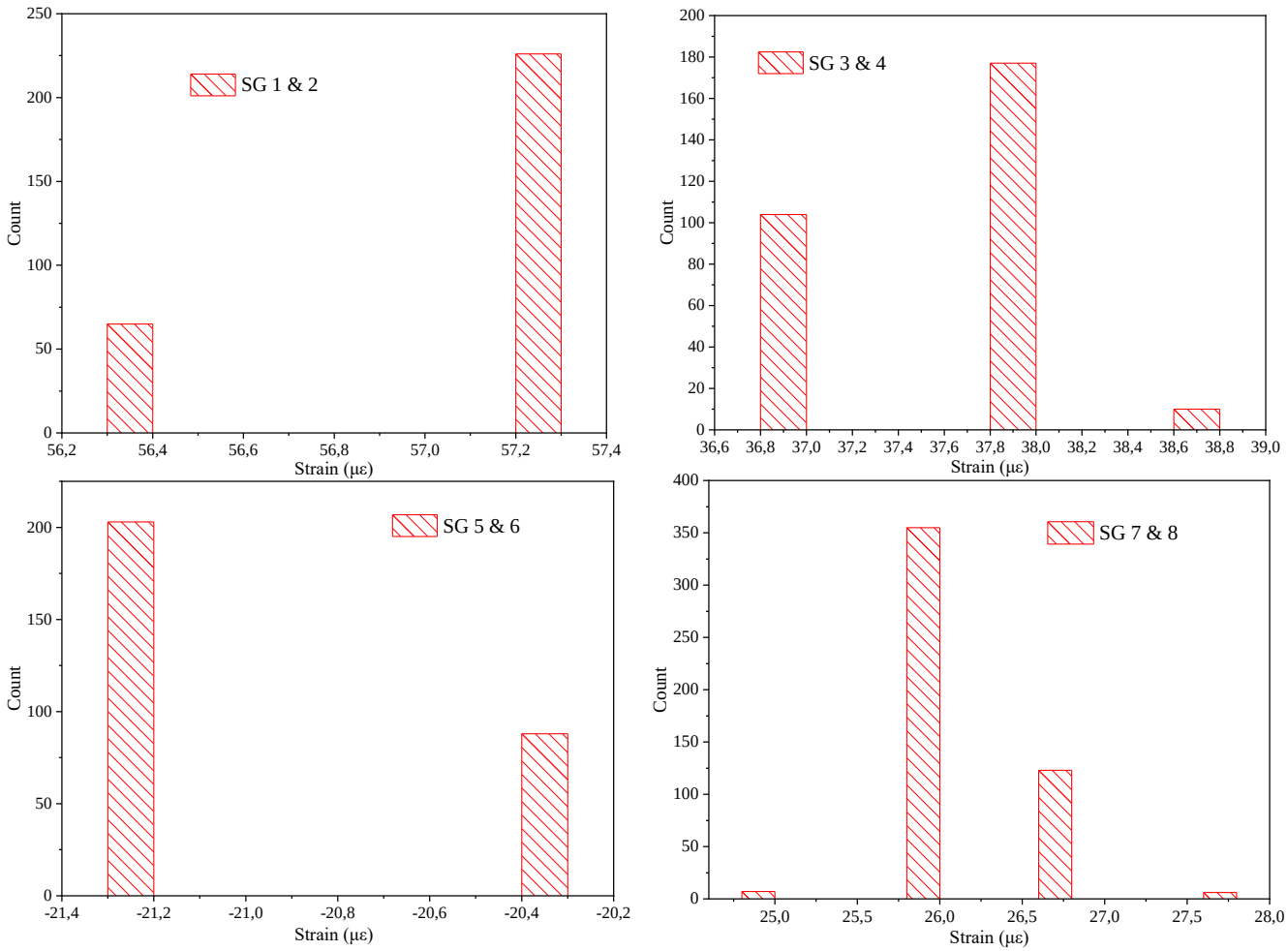


Figure 5.32: Test Specimen 2 / Case 1 - Top Flange Shear Strain Distribution (Initial Specimen)

From the histograms shown in the previous pages, it is made clear that measured strains are generally distributed along few discrete values. The estimated level of added noise was considered to be the maximum range along which strain data were distributed in each separate case. That means, in Case 1 of Test Specimen 1, added noise was considered to be at ±3 με and in Case 1 of Test Specimen 2 at ±1.5 με. Both estimations are within the initial assumptions made during the course of the validation of the developed methods using virtual strain data, as detailed in Chapter 4 previously.

As mentioned earlier, the addition of artificial noise was used to “pollute” strain data obtained from the digital twin in all other sensor positions, apart from those corresponding to those of the strain gages used in the laboratory tests. This was done in order to use the complete formulation of the damage detection schemes developed and test the methods on a scenario resembling a realistic application as closely as possible. Following are the predictions that resulted from the implementation of both the GA-based method as well as the ANN-based one.

Table 5.8: Test Specimen 1 / Case 1 - Comparative Damage Detection Results

Damage Detection Results				
		Actual	Predicted	
			ANN	GA
Face		Left Web (4)	Left Web (4)	Top Flange (1)
Span		Left Span (2)	Left Span (2)	Right Span (1)
Longitudinal Location (from Mid-Span)	mm	170	196	450
Transverse Location (from Mid-Height)	mm	23	1	-23

Table 5.9: : Test Specimen 2 / Case 1 - Comparative Damage Detection Results

Damage Detection Results				
		Actual	Predicted	
			ANN	GA
Face		Top Flange (1)	Top Flange (1)	Bottom Flange (3)
Span		Left Span (2)	Left Span (2)	Right Span (1)
Longitudinal Location (from Mid-Span)	mm	170	215	440
Transverse Location (from Mid-Height)	mm	-23	11	23

As evidenced by the results in Table 5.8 and Table 5.9 above, the GA-based method, which was able to detect damage when using only localized data, now failed to detect the same cases accurately. This means that the added noise pollution across all paths is detrimental to the method's accuracy, at least in the levels of added noise considered in this thesis. On the other hand, the ANN-based method succeeded at correctly classifying damage, as far face and span were concerned while it exhibited reduced accuracy in the prediction of the location of damage, and more specifically its transverse component, which was a trait that it exhibited in the tests using virtual strain measurements as well.

The successful results produced from the ANN-based approach are presented in Figure 5.33 in the following page for further reference. They are shown in the form of bar charts comparing actual versus predicted results. Each bar chart refers to a specific identifier and all identifiers are expressed in the terms that have been described multiple times earlier in this thesis (e.g. in Chapter 3.5.2).

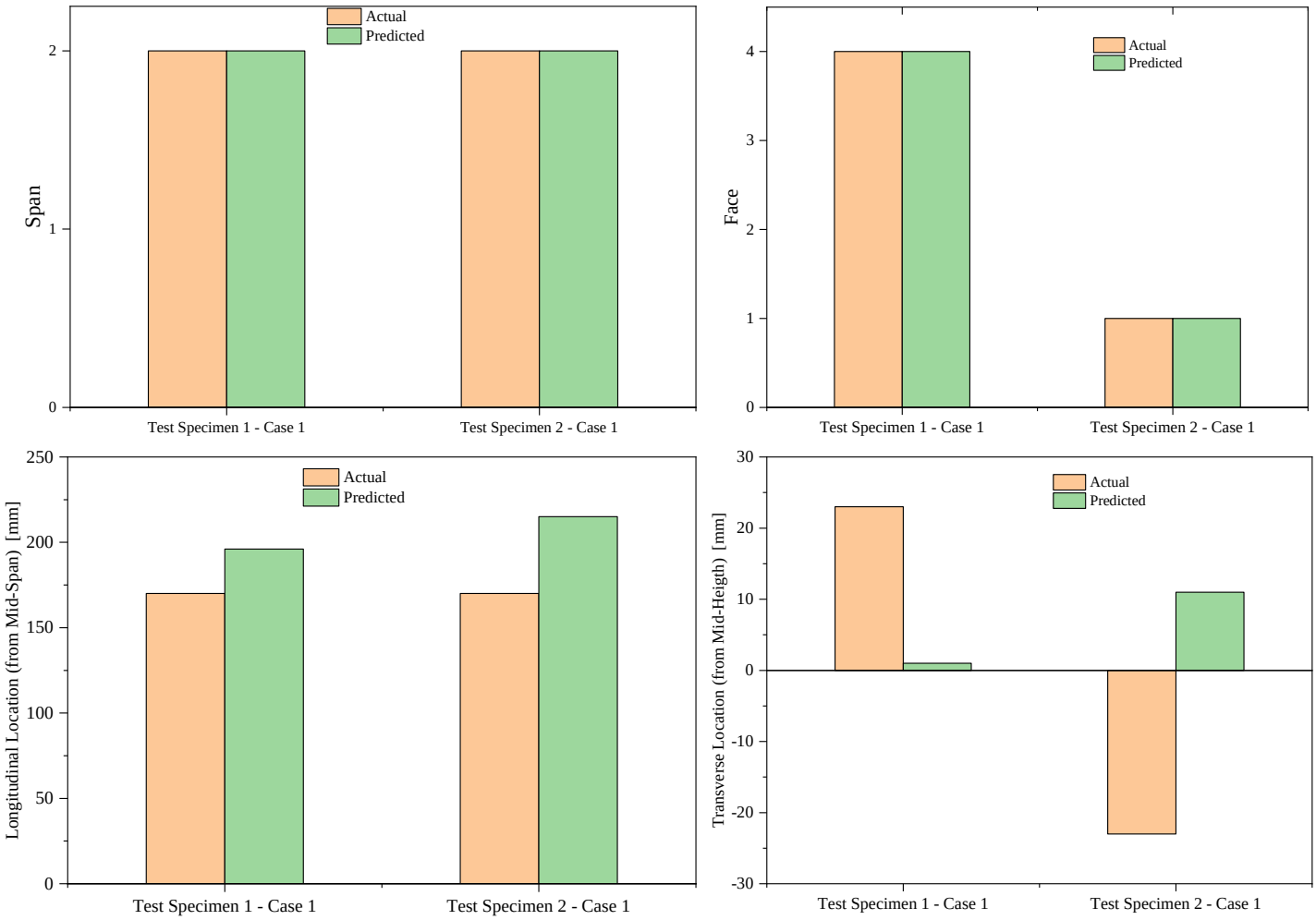


Figure 5.33: Summary of Successful Damage Detection Results for Full-Sensor Digital Twin

5.5 Concluding Remarks

The results obtained from the laboratory tests provided the opportunity to test the damage detection methods that were developed on real-world strain data. Before putting the methods to the test, the data themselves were analyzed and provided key insights on the nature of actual sensors strain readings. Specifically, it was shown that even in theoretically zero-strain directions, some strain will always be measured. That baseline strain, uncertain in nature, proved to be a very important factor in the ability of the damage detection schemes to predict the location of damage accurately. The local nature of the effect of damage, as expected by the physics of the problem and predicted in the numerical analysis, was also proven during the tests beyond doubt.

In terms of the implementation of the two methods, two different approaches were followed. In the first, only the GA-based method was implemented on a modified version of the digital twin exactly matching the test set-up. Two of the four cases considered were successfully classified using this method; namely, those located

closest to the load application point for both flange and web cases, where the higher levels of strain can "drown" out measurement noise more effectively.

Then, a more complete approach using both methods was implemented on the test cases that produced successful results for the first method. In this, the complete digital twin formulation was used in order to simulate the state on an actual asset, where the majority of sensors would be unaffected by damage and introduce only noise to the strain monitoring system. To achieve that, the digital twin was appropriately adopted to match the test specimen characteristics and results were obtained for conditions matching those of the laboratory test. In these results, the actual strain data were assimilated by substituting with them the virtual ones corresponding to the same locations, while for the remaining sensor locations, noise was added to the numerical data to simulate actual measurement conditions. The level noise was selected based on a statistical estimation of the levels of random added noise found on the actual measured strain data.

The implementation of the damage detection methods led to mixed results. The GA-based approach failed to predict damage locations correctly indicating that the induced noise levels proved detrimental to the method's capabilities. On the other hand, the ANN-based approach was successful, with its two classification neural networks correctly predicting the span and face where damage was locating and with its fitting neural network producing relatively accurate predictions regarding longitudinal location, while less accurate ones for the transverse components. However, overall the ANN-based approach's performance was highly satisfactory and its results very promising, especially with regard to possible future applications.

6 Conclusions and Future Work

6.1 Conclusions

In this work, a damage-identification framework for thin-walled girders subjected to three-point bending was developed. This model was chosen as it provided an adequate simplification of the structural response of ship hulls under still water loading. The methods employed to facilitate damage identification used machine learning tools, i.e. Artificial Neural Networks and optimization tools, i.e. Genetic Algorithms.

At first, the problem was modeled using FEA, specifically the ANSYS software. A simple model of a thin-walled girder subjected to three-point bending was developed and validated using estimations from traditional beam theory. A strain monitoring scheme was proposed to be used as the basis of the SHM system, which suggested that zero-strain directions along the girder should be monitored. The extent of this scheme was determined using a complementary, more detailed FE model in order to avoid factoring in areas where local effects were present. This was found to longitudinally span from 0.2L-0.4L and 0.6L-0.8L.

A disturbance in the strain field was then added to simulate the effect of damage. For this thesis, a circular hole was selected to represent this feature, which was referred to as a Strain Field Disturber. Subsequently, it was investigated whether this feature would introduce a disturbance in the strain field which could be captured by the strain monitoring scheme; this was confirmed by the results, showing that indeed a deviation was evident in the strain distribution. It was also observed that the effect of damage on the strain field was highly localized, being constrained to a narrow area on either side of the defect. This proved to be a valuable insight for the future. With the strain monitoring scheme in place, the two damage detection methods were developed and then put to the test.

In the GA-based method, the problem of damage detection was treated as a multi-objective optimization problem, minimizing error functions between target strains and strains obtained by iteratively solving the FE model. The optimization module of ANSYS Workbench was used for the implementation of this method, using the MOGA multi-objective genetic algorithm. For the validation of this method, a number of damage scenarios were considered, and the corresponding strain results were obtained by solving the digital twin which were then used as input for the GA-based approach. The method was tested, using a combination of initial MOGA parameters maximizing accuracy, on input data corresponding to three sensor configurations: one having 16 sensors per path; another having 10 sensors per path and the last one with 5 sensors per path. At first, the input data used were directly obtained from the digital twin and results were very promising, indicating consistent accuracy in all configurations. Then, the same input measurements were polluted with artificial noise to simulate actual on-site measurements and the method was tested on them. The results indicated that a decrease in the number of sensors contained per path led to a decrease in the noise threshold,

below which accurate detection was possible. Also, it was made clear that in the cases where damage was located on webs the method was less susceptible to noise than in the case of flange-located damage.

In the ANN-based method, three neural networks were employed to achieve damage detection: the first two were classification networks used to predict the span and face of the web where damage is located and the second a fitting network aiming to approximate damage characteristics, such as location and magnitude. They were trained using digital twin-obtained strain data corresponding to an array of damage scenarios, selected to provide an adequate training dataset for the three neural networks. In terms of their validation, the same paradigm as before was implemented. ANNs were trained and then tested, for three different sensor configurations; the tests were performed for both "clean" and "polluted" strain data. Regarding the results obtained from this method, in the "clean" case they were similar to the ones from the GA-based method, albeit showing diminished accuracy in their predictions of damage characteristics. When noise was included, results from this method indicated that it was more susceptible than the alternative one. Overall, results from both methods were very encouraging and showed promise for future considerations.

Finally, a series of experimental tests were carried out to assess the applicability of the aforementioned in real-world conditions. When implementing the methods described above using actual strain data, mixed results were obtained. The GA-based method showed good performance when applied to conditions exactly simulating the experiment while the ANN-based approach was very accurate when actual data were assimilated to digital-twin obtained measurements and a complete formulation was used. Apart from the obtained results, the experimental tests provided a wealth of knowledge regarding the nature of on-site measurements and provided insights on the importance of data assimilation techniques for data-driven physical models.

6.2 Suggestions for Future Works

In this work, a simplified thin-walled girder model was used to simulate hull structural response. In future works, a more detailed box girder model, such as Reckling No. 23 could be used as the digital twin. More realistic modes of structural degradation could be added, such as corrosion or buckling of stiffeners included in the model. Also, another direction that could be followed with regard to the damage identification framework, is the development of a mixed method based on those developed in this thesis, utilizing ANNs to feed a GA in order to produce a less computationally demanding approach, more suitable for complex structures. Finally, it is recommended to carry out more detailed experiments with the aim to obtain valuable experimental data required in the development of data-driven physical mode

7 References

ABS Global Engineering and Technology Overview. (2018, August).

Bathe, K.-J. (2014). *Finite Element Procedures* (Second Edition ed.). Watertown, MA: K.J. Bathe.

Bischoff, M., Bletzinger, K.-U., Wall, W., & Ramm, E. (2004). Models and Finite Elements for Thin Walled Structures. In E. Stein, R. Borst, & T. Hughes (Eds.), *Encyclopedia of Computational Mechanics.* John Wiley & Sons, Ltd.

Chowdhury, M., Nuruzzaman, D., & Rahaman, M. (2014). Frictional Characteristics of Steel Materials Sliding against Mild Steel. *Advanced Material Research, 903*, pp. 33-38.

Coello Coello, C., Lamont, G., & Van Veldhuizen, D. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (2nd ed.). Springer.

Cook, R. D. (1995). *Finite Element Modeling for Stress Analysis.* John Wiley & Sons, Inc.

Farrar, C. R., Doebling, S. W., & Nix, D. A. (2001, January 15). Vibration-based structural damage identification. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 359*(1778), pp. 131-149.

Fonseca, C., & Fleming, P. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest (Ed.), *Genetic Algorithms: Proceedings of the Fifth International Conference* (pp. 416-423). San Mateo, CA: Morgan Kaufmann.

Fujikubo, M. (2019, September 23). Digital Twin for Ship Structures - A Research Project in Japan [Conference Presentation Abstract]. *The 14th International Symposium on Practical Design of Ships and Other Floating Structures*. Yokohama, Japan.

Fujikubo, M., & Yao, T. (2016). *Buckling and Ultimate Strength of Ship and Ship-Like Floating Structures.* Oxford: Butterworth-Heinemann.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation* (2nd ed.). Prentice Hall.

Hughes, O., & Paik, J. (2010). *Ship Structural Analysis and Design.* Jersey City, NJ: The Society of Naval Architects and Marine Engineers.

Kefal, A., Olgun, H., & Oterkus, E. (2015). A smart system to determine hull sensor locations for structural health monitoring of ship structures .

Keller, J., Liu, D., & Fogel, D. (2016). *Fundamentals of Computational Intelligence: Neural Networks, Fuzzy Systems and Evolutionary Computation.* John Wiley & Sons, Inc.

Mansour, A., & Liu, D. (2008). *The Principles of Naval Architecture Series: Strength of Ship And Ocean Structures.* Jesey City, NJ: Society of Naval Architecture and Marine Engineering.

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill International Editions.

Paik, J., & Frieze, P. (2001). Ship Structural Safety and Reliability. *Progress in Structural Engineering and Materials, 3*(2).

Preisler, A., Schröder, K.-U., & Schagerl, M. (2018). Intrinsic Damage Assessment of Beam Structures Based on Structural Damage Indicators. *American Journal of Engineering Research, 7*(6), pp. 56-70.

Rao, S. (2020). *Engineering Optimization: Theory and Practice* (5th ed.). Wiley.

Rytter, A. (1993). *Vibration Based Inspection of Civil Engineering Structures.* Ph.D. Thesis, University of Aalborg, Aalborg, Denmark.

Saathesh Babu, P., Mathiazhagan, A., & Nandakumar, C. (2014). Corrosion Health Monitoring System for Steel Ship Structures. *International Journal of Environmental Science and Development, 5*(5).

Sharma, P., Knezevic, D., Huynh, P., & Malinowski, G. (2018). RB-FEA Based Digital Twin for Structural Integrity Assessment of Offshore Structures.

Smogeli, O. (2016). *Digital Twins at Work in Maritime and Energy.* DNV GL.

Stull, C. J., Earls, C. J., & Koutsourelakis, P.-S. (2011, February). Model-based Structural Health Monitoring of Naval Ship Hulls. *Computational Methods in Applied Mechanics and Engineering, 200*(9-12), pp. 1137-1149.

Zienkiewicz, O., & Taylor, R. (2006). *The Finite Element Method for Solid and Structural Mechanics* (6th ed.). Oxford: Butterworth Heinemann.