

# Journal of Mechanics of Materials and Structures

**EFFICIENCIES OF ALGORITHMS FOR VIBRATION-BASED  
DELAMINATION DETECTION: A COMPARATIVE STUDY**

Obinna K. Ihesiulor, Krishna Shankar, Zhifang Zhang and Tapabrata Ray

Volume 8, No. 5-7

July–September 2013



## EFFICIENCIES OF ALGORITHMS FOR VIBRATION-BASED DELAMINATION DETECTION: A COMPARATIVE STUDY

OBINNA K. IHESIULOR, KRISHNA SHANKAR, ZHIFANG ZHANG AND TAPABRATA RAY

The need for efficient and low-cost techniques adequate for damage detection has become of great interest in engineering applications where structural health monitoring (SHM) is of paramount importance. Promising algorithms for SHM have to deliver results with very low computational and response time requirements and be trustworthy within a certain accuracy. Different algorithms (artificial neural networks (ANN), response surface methodology (RSM), and optimization techniques — gradient-based local search (GBLS) and nondominated sorting genetic algorithms (NSGA-II)) are proposed to fill this research gap. The concept of a surrogate model as a fast-executing model is also introduced. Because the objective of this paper is to concentrate on viable techniques suitable for damage detection using vibration methods with very low computational requirements, surrogates are therefore employed to curtail the computational expense. Particularly of interest among the proposed algorithms is RSM, the principle of which has proved successful in the pharmaceuticals industry over the years. However, RSM has not been so widely used in the field of structural engineering for delamination detection. In this paper, we have demonstrated that a fourth-order polynomial has the capability to detect delaminations in composite structures. In order to reduce the size of training data required to solve the inverse problem by the proposed algorithms, the idea of a suitable design space is brought to the limelight as the combination of all possible simulations that one is concerned about. Since the overall sum of design space is usually prohibitively large, we have used  $K$ -means clustering to effectively achieve this. This research concerns the application of ANN, RSM, and optimization techniques for delamination detection using changes in natural frequencies before and after damage. Efficiencies of algorithms (ANN, GBLS, and NSGA-II) are compared with the developed RSM models in terms of the accuracy of delamination detection and response time requirements. The methods have been shown to compete effectively for delamination detection and are accurate in detecting the size and locations of delaminations at midplanes. RSM has a unique feature in that it produces models with a small training dataset requirement and also generates mathematical models that are easy to interpret and implement. The optimization techniques, when integrated with surrogate models, require small training sets clustered through the entire design space. ANN, however, requires large training datasets to achieve its results. As such, the potential of these algorithms as tools for on-board damage detection when integrated into a SHM system is successfully demonstrated.

### 1. Introduction

The utilization of in-service composite structures can be affected by degradation resulting from prolonged use and exposure to harsh environmental conditions. This is because composite structures are susceptible

---

Obinna K. Ihesiulor is grateful to the University of New South Wales at the Australian Defense Force Academy for provision of tuition fee waiver and living allowance support.

*Keywords:* vibrations, delaminations, ANN,  $K$ -means clustering, surrogates, RSM, optimization techniques.

to hidden or barely visible damage caused by impacts, design errors, overheating, loading abrasion, and fatigue, that if unchecked results in fiber breakage, matrix cracking, and delaminations [Chakraborty 2005]. These factors can cause severe consequences for in-service structures with regards to higher life-cycle costs, low structural reliability, and loss of operational capability causing loss of lives and property [Kessler et al. 2005]. Hence, early and accurate detection and monitoring of structural failures such as delaminations is a principal concern.

Delamination detection has therefore gained much attention from the structural engineering community because unpredicted delamination damage may cause catastrophic failures [Zheng et al. 2011]. Hence, the need to avoid delamination failure by providing a reliable and effective nondestructive damage-identification technique is crucial to maintaining the safety and integrity of structures. Damage detection is of paramount importance because it is the most vital subsystem of structural health monitoring (SHM). Damage detection can be achieved via any of various methods such as visual inspection, nondestructive evaluation methods, and solution of inverse algorithms. In the field of medicine, for example, a doctor regularly monitors a patient's blood pressure to determine the health of the patient by observing deviations using sophisticated equipment. Similarly, engineers monitor the integrity of a structure by measuring changes in the responses of the structure which can lead to structural failure. Taking immediate responses in both scenarios can avoid catastrophic implications [Mufti 2001]. An effective SHM system involves the use of expertise in many disciplines, giving rise to solutions of the multidisciplinary problem involving damage detection modeling systems via finite element analysis, optimization methods, structures and materials, computers, communication and electronics, real-time controllers, intelligent processing, and so on. The aim of SHM is not just to detect structural failure, but also to provide an early indication of damage. The early warning provided by an SHM system can then be used to define remedial strategies before the structural damage leads to failure.

SHM is therefore the key to securing confidence in the utilization of fiber reinforced polymers (FRP) in engineering applications. The unique advantages of FRP composite materials have made them preferred in SHM applications. Previously, structures were monitored by carrying measuring devices to the site each time a set of readings was required. Nowadays, by using vibration measurements for structural health monitoring, structures can be monitored from time to time to ensure that they are in good condition by obtaining an extensive amount of processed data off-site. In this work, delamination detection in composite laminates is evaluated in the context of structural health monitoring, which essentially is a reliable system with the ability to detect and interpret adverse changes in a structure as a result of damage. The motivation driving SHM is that knowing the integrity of in-service structures on a continuous real-time basis is essential to structural engineers. Monitoring and evaluating the integrity, in-situ behavior, and health condition of a structure accurately and efficiently while it is in service optimizes resources for repair/replacement, reduces downtime while increasing productivity, and ensures the safety of lives and property. As a result, SHM enables avoidance of catastrophic failure through early detection of problems, optimal use of structures, prevention of regular shutdowns of in-service structures, and inspection of hard-to-reach places. SHM also replaces periodic maintenance with long-term maintenance schedules. This reduces maintenance labor and minimizes human involvement, thus improving safety and reliability [Kessler et al. 2005].

Most nondestructive testing techniques such as ultrasonic inspection, thermography, optical holography, and mechanical impedance for delamination identification in composite laminates [Buynak et al.

1989; Cantwell and Morton 1992; Broughton et al. 2000; Rao 2007] cannot be used for real-time and online damage detection [Doebling et al. 1998]. Moreover, most of these techniques are mainly applicable for the inspection of limited areas of a structure locally and hence are labor-intensive, time-consuming, and cost-ineffective when considering large structures [Dackermann 2010]. Vibration-based monitoring is a viable method to overcome these limitations. This is because of the reliability of its measurements, ease of implementation, and relative cost competitiveness [Cawley and Adams 1979; Kim and Yiu 2004]. In addition, utilization of a vibration-based monitoring tool can provide fast in-situ and real-time monitoring [Doebling et al. 1998]. SHM exploiting vibration measurements are global methods based on the principle that degradation due to damage in a structure changes its vibration parameters, namely, its natural frequencies, mode shapes, and damping characteristics. It is hence feasible to use any one measured vibration quantity to characterize and identify the presence of damage via an inverse modeling. The choice of the natural frequencies as one of the commonly used vibration parameters is attractive because the natural frequencies can be conveniently measured and determined easily from just a few accessible points on the structure and are usually less contaminated by experimental noise [Fang and Tang 2005]. Natural frequency-based methods use the natural frequency changes before and after damage as the basic feature for damage identification in solving the inverse problem.

Reviews of vibration-based health monitoring methods utilizing artificial neural networks (ANN) and optimization algorithms appear in previous works [Islam and Craig 1994; Okafor et al. 1996; Doebling et al. 1998; Valoor and Chandrashekhara 2000; Harrison and Butler 2001; Nag et al. 2002; Watkins et al. 2002; Chen et al. 2004; Su et al. 2005; Addin et al. 2006; Zheng et al. 2011]. Although previous works have demonstrated the feasibility of ANN and optimization algorithms for delamination damage detection, some limitations are worth highlighting. Most of the damage detection methods that have been reviewed attempt to identify delamination by solving an inverse problem, which often requires the construction of numerical models. Numerical simulations have therefore become increasingly useful for studying the performance of engineering structures. However, because of the large scale of many damage-identification problems, it is often not feasible to conduct many experiments to explore all damage scenarios. Tremendous amounts of computing time are required to run these simulations and oftentimes computing power is simply not available to conduct such complex simulations due to budget and time considerations. This dependency on firsthand numerical models, which are computationally expensive, makes these approaches unpromising for SHM.

Using a small number of datasets, surrogate models can be built to efficiently explore the entire design space to determine areas of interest while lowering computational expense. Essentially, surrogates are models that will run in seconds on a single processor in contrast to the hours that it may take to run a more detailed analysis on a multiprocessor machine. On the other hand, one of the remarkable demerits of ANN is the requirement of large amounts of training data. ANN has, however, demonstrated undoubted efficiency for complex damage-detection schemes by seeking to discriminate between damaged and undamaged specimens, and has been widely employed for pattern recognition, classification, function approximation, signal processing, and damage identification [Addin et al. 2006]. ANN offers capabilities such as self-adaptiveness, generalization, abstraction capabilities, and suitability for real-time applications [Tsoukalas and Uhrig 1997]. These features make ANNs powerful tools for vibration-based damage identification. But the computational power and size of training data required to solve the inverse problem by ANN is usually exceedingly large. To fill this gap, the idea of *K*-means clustering algorithm

is introduced. Since the overall sum of design space is usually This algorithm smartly determines which numerical simulations should be run when resources are scarce and models are fit to these smartly chosen data points.

The content of this research focuses on SHM as a diagnostic tool for delamination damage detection via the utilization of efficient inverse algorithms with significant interest in their prediction-error quantification. In this aspect of SHM, the abstract data numbers are converted into quantities that relate directly to the responses of a structure, that is, the natural frequency measurements will be converted into quantities of delamination parameters via inverse algorithms. The key objective of the present study is therefore to provide a comparative analysis of different algorithms for delamination detection. Based on finite element simulation of a composite beam-type structure, a comparative study of several damage-identification algorithms is provided to illustrate the validity and effectiveness of the algorithms. The ANN adopted in this work is a response surface approximation method that is based on the concept of artificial intelligence. The ANN is to provide surrogate models to computationally expensive finite element (FE) models. The motivation for this is derived from the concept that constructing a surrogate model to approximate any expensive function can substantially reduce the computational cost for objective function evaluations during the course of optimization and improve the optimization search performance. Thus, once a neural network is effectively trained, it is capable of being used for future interpolation and approximation.

For the response surface methodology (RSM), a simple fourth-order polynomial model that associates input parameters to the output response is used for delamination prediction. RSM approximates the output of a given system as a function of some input variables (design variables) by solving a system of nonlinear equations. This method is effectively employed as an inexpensive low-order approximation model for delamination prediction and, because of the model form, minimal effort is required to build the model. The method also has relative flexibility in the range of problems it is able to model, unlike the ANN, which requires a huge effort to train its model.

Another advantage of RSM is the ease of implementation in damage-identification settings. The selection of the sampling points for building ANN surrogate models is vital and demanding because the prediction capabilities of an approximation function are highly influenced by the sampling points in the given design space [Kanungo et al. 2002]. The  $K$ -means clustering technique is used so as to ensure that the sampling points are evenly distributed over the design space. This method gives a systematic and efficient means of analyzing the complete design space. It explores the high-dimensional design space and screens the most clustered design points corresponding to the set of design variables.

To solve the optimization problems, local and global optimization algorithms are employed for delamination detection. Gradient-based optimization schemes are local optimizers that can get stuck at the first optimum obtained during the search process. When applied to continuous problems, this algorithm gives better performance than any other optimization scheme but is highly unsuitable when searching for global optima when there is a mixture of discrete and continuous variables [Schittkowski 1986]. Global optimizers like the nondominated sorting genetic algorithm (NSGA-II) are most promising in both discrete and continuous problems due to their robust and random nature of search but with a significantly high computing cost. The high computing cost involved in deducing the function objectives is greatly reduced using surrogate models, and hence taking advantage of their global optimization behavior.

## 2. Mathematical formulation

**2.1. Mathematical formulation of the optimization problem.** The goal of an optimization task is to ascertain a set of values that result in a maximum or minimum of a function called the objective function. The objective function is a mathematical expression describing a relationship of the optimization parameters. Optimization is an activity (which could be single or multiobjective) that aims at finding the best (that is, optimal) solution to a problem. Single-objective optimization is scalar valued with a single unique solution whereas when the objective is vector valued, the optimization process is referred to as multiobjective [Ray et al. 2001]. In this delamination detection optimization problem, the objective function is defined as a single objective function and two key components are effectively required to solve the optimization problem:

- the simulator (which essentially computes/simulates the natural frequencies) and
- the comparator (objective function).

The optimization objective is to compare and minimize the errors between measured (actual) and predicted natural frequencies. The percentage change in frequency caused by delaminations for  $i$ -th mode ( $dF_i$ ), where  $i = 1, \dots, n$ , is defined as

$$dF_i = \frac{F_{ui} - F_{di}}{F_{ui}} \times 100, \quad (1)$$

where  $F_{ui}$  and  $F_{di}$  are the numerically predicted natural frequencies of the undamaged and damaged composite beam, respectively.

The objective function (comparator) is the norm of the difference between the measured ( $dF_{Mi}$ ) natural frequencies for which the delamination parameters are to be determined and the numerically predicted natural frequencies ( $dF_i$ ). All the computational analysis for the delaminated natural frequencies (in Hz) is defined with respect to the beam in perfect condition and the deviation is expressed as a percentage change in the natural frequencies.

The error,  $E_i$ , between the measured and predicted shifts in frequencies due to the delamination for the  $i$ -th mode is given by

$$E_i = \left( \frac{dF_i - dF_{Mi}}{dF_{Mi}} \right)^2, \quad (2)$$

where  $dF_{Mi}$  is the measured percentage change in the natural frequencies for which the delamination parameters are to be ascertained.

The use of high-fidelity simulation tools to compute the natural frequencies of composite structures for the objective function always comes with an unavoidable computing time. For this reason, surrogate models are a key element to reduce the optimization cycle time by providing alternative function evaluations for the objective function. The objective function with surrogate,  $\text{Obj}_S$ , is therefore given by the sum of the errors

$$\text{Obj}_S = \sum_{i=1}^n E_i = E_1 + E_2 + E_3 + \dots + E_n. \quad (3)$$

Similarly, the objective function without surrogate,  $\text{Obj}$ , is given by

$$\text{Obj} = \sqrt{(dF_1 - dF_{M1})^2 + \dots + (dF_n - dF_{Mn})^2}, \quad (4)$$

where  $n$  is the maximum  $i$ -th mode.

In order to evaluate the objective function, the natural frequencies of the undelaminated and delaminated composite beams are simulated via finite element analysis (FEA). The objective function is evaluated using frequency shifts due to the delamination predicted using FEA and the measured frequency shifts. Very few simulations were used to build the database using  $K$ -means clustering and then surrogate-assisted optimization.

**2.2. Solution strategy.** An overview of the approach is shown in [Figure 1](#).

A detailed description of the proposed methods to detect delaminations can be summarized as follows:

*Optimization without surrogates.* Step 1: Develop an FE model with ANSYS that computes the natural frequencies before and after damage. This is also known as the simulator. Step 2: Engage the simulator directly with the optimizer to evaluate the objective function to be minimized to determine the delamination parameters. The objective function, also known as the comparator, essentially minimizes the sum of errors between the simulated natural frequency changes before and after damage and the actual ones to determine the delamination parameters for any number of variables ( $N_d$ ).

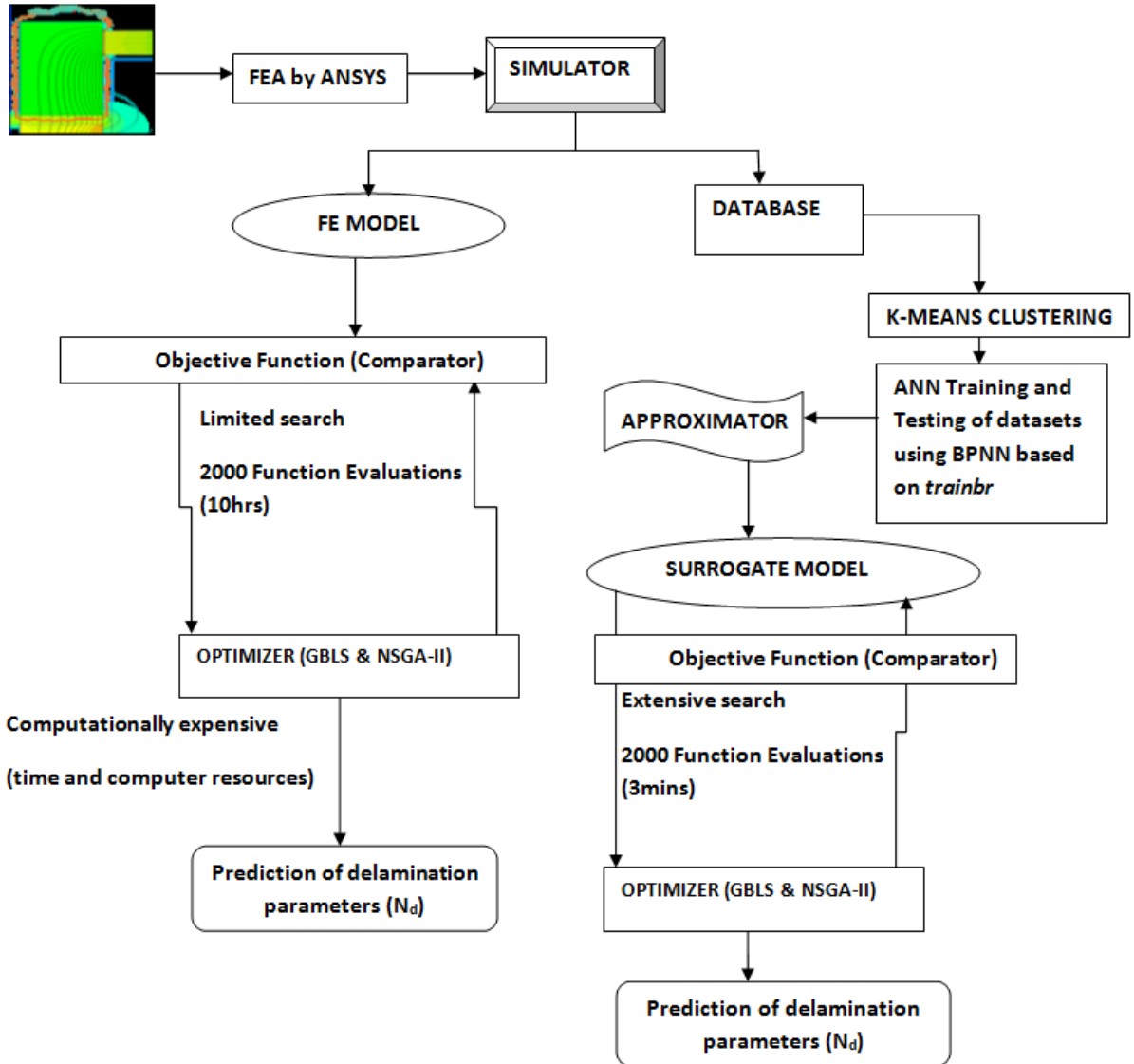
*Surrogate-assisted optimization.* Step 1: Develop an FE model with ANSYS that computes the natural frequencies before and after damage. Step 2: Use the simulator to generate a database in the case of optimization with surrogates. Step 3: Reduce the size of the generated database, which is usually large, by a  $K$ -means clustering method. This ensures that a small number of well-clustered datasets within the entire design space is used for delamination prediction. Step 4: Use the  $K$ -means clustered datasets to create a surrogate model. Step 5: Engage the surrogate model directly with the optimizer to evaluate the objective function to be minimized to determine the delamination parameters for any number of variables ( $N_d$ ).

*Direct solution via ANN and RSM.* Step 1: Develop an FE model with ANSYS that computes a database of natural frequencies before and after damage. Step 2: Do a  $K$ -means clustering if the size of the database is to be reduced. Step 3: Use the database to train ANN and RSM models that give the delamination prediction for any number of variables ( $N_d$ ).

**2.3. Modeling of the laminated composite beam.** We studied the vibration behavior of eight-layer  $[0^\circ/90^\circ/90^\circ/0^\circ]$ s glass-epoxy laminated composite cantilever beams with and without delaminations. Glass fiber (E-glass) is used as reinforcement in the form of unidirectional fibers with epoxy resin as matrix for the composite beam. The laminates are reinforced unidirectionally. The material properties are given in [Table 1](#) for the FE analysis and theoretical model. The composite beam has length  $L = 267$  mm, width  $W = 25.4$  mm, and thickness  $h = 1.778$  mm.

$E_1$	$E_2$	$G_{12}$	Poisson's ratio ( $\nu_{12}$ )	Density ( $\rho$ )
42.34 GPa	11.72 GPa	3.0025 GPa	0.27	1901.5 kg/m <sup>3</sup>

**Table 1.** Material properties of the composite beam laminates.



**Figure 1.** Schematic of solution methodology.

Considering through-width delaminations, delaminations are simulated extending through the width of the beam, with locations and sizes in the ranges  $0 < X \leq 70$  and  $0 < a \leq 58$ , respectively, satisfying the requirement that delamination must not extend outside the beam. For any delamination pattern  $([Z, X, a])$ ,  $Z$  denotes the interface, and the normalized delamination location is expressed as  $X = X_{\text{actual}}/L$ , where  $X_{\text{actual}}$  is the distance from the middle of the delamination to the fixed end of the beam and  $L$  is the total beam length. Similarly, the normalized delamination size is given as  $a = a_{\text{actual}}/L$ , where  $a_{\text{actual}}$  is the length of the delamination along the axis of the beam. Delaminations are simulated at different interfaces,  $Z = 1$  to  $4$ ;  $1$  being the midplane and  $4$  the outermost interface.



**2.3.1. Finite element (FE) analysis of the composite beam.** Herath et al. [2010] investigated the effects of delamination size and location in composite beams using a FEA model under quasistatic loading and an analytical Euler-beam model. Ramanamurthy and Chandrasekaran [2011] used FE modeling to develop a damage-detection method in a composite cantilever beam with an edge crack. Ishak et al. [2001] employed the application of the strip element method and adaptive multilayer perceptron networks (MLP) for inverse identification of interfacial delaminations in carbon/epoxy laminated composite beams. In this paper, FEA is used to solve the forward problem and generate data for frequency shifts for known delamination parameters. Numerical analysis is carried out using the commercial finite element program ANSYS 12.1 to build the FE models for both the undelaminated and delaminated glass fiber reinforced composite beams to investigate their vibration behavior. Analysis is carried out on a three-dimensional eight-node layered solid element (SOLID185) with three degrees of freedom at each node. The shell section is adopted to define the layer information.

In the FE model for the delaminated beam, the delaminated beam was modeled as two volumes, separated along the interface at which the delamination is located. The nodes situated along the interface of undelaminated segments were merged together while nodes in the interface of the delaminated area are left unmerged [Zhang et al. 2010]. Contact elements (TARGE170/CONTAC173) were introduced between the delaminated surfaces to prevent separation and interpenetration, so the upper and lower sublaminates had the same deflection, acting as two separate beams constrained to move together. To build the model, the number of elements required to provide acceptable levels of accuracy and also determine how fine the mesh should be in order to get convergence of the numerical results without excessive use of computational time was determined using a convergence analysis. The final FE mesh employed had 200 elements along the length, six elements across the width, and one element for each layer of the eight-ply laminate.

The natural frequencies obtained from the undelaminated model are consistent with those from theory. To determine the natural frequencies of the delaminated beam, ANSYS Workbench batch mode simulation was used to setup the eigenvalue modal analysis. Further, using the block Lanczos method, the natural frequencies of the first eight bending modes were extracted, discarding the torsion and in-plane bending modes. The natural frequencies of the undamaged and damaged glass fiber/epoxy composite beam were computed. The natural frequencies obtained from the FE model have been compared to theoretical results for validation purposes.

**2.3.2. Theoretical modeling of laminated composite beam.** The theoretical formulation of the vibration characteristics of the models of beams with and without delaminations were based on the pioneering works in this field. Ramkumar and Kulkarni [1979] were the first of these pioneers. They developed a simplified model to compute the free vibration frequencies of a cantilever laminated beam with a single through-the-width delamination at the interlaminar position. Their basic concept was to deduce mathematically the actual vibration properties of four different Timoshenko beams combined together by considering the delaminated and undelaminated portions of the beams. Their mathematical model of the eigenvalue problem fulfilled all the necessary boundary and continuity conditions between the adjoining beams. They also carried out experimental studies, which showed that the analytical computations of the natural frequencies were uniformly lower than the experimental ones. As a follow up, Wang and Liu [1982] employed the analytical model of [Ramkumar and Kulkarni 1979] using classical Bernoulli–Euler

$[Z, X, a]$	Mode	Theory ( $F_{ui}$ )	Theory ( $F_{di}$ )	FE ( $F_{ui}$ )	FE ( $F_{di}$ )	% diff. ( $F_{ui}$ )	% diff. ( $F_{di}$ )
[1, 54, 24]	1	16.06	15.97	16.04	15.94	0.159	0.14
	2	100.67	100.54	100.46	100.33	0.211	0.2
	3	281.9	252.38	281.04	251.49	0.349	0.3
	4	549.4	522.12	550.07	52.81	0.443	0.12
	5	913.09	780.59	907.91	776.82	0.482	0.57
	6	1364.04	1192.95	1353.75	1184.85	0.679	0.75
	7	1905.15	1636.83	1886.64	1624.22	0.771	0.97
	8	2536.44	2191.25	2505.46	2170.49	0.947	1.22
[1, 62, 18]	1	16.06	16.032	16.039	16.007	0.154	0.14
	2	100.67	100.333	100.458	100.120	0.213	0.2
	3	281.9	275.044	281.042	274.152	0.324	0.3
	4	549.4	504.202	550.068	501.887	0.459	0.119
	5	913.09	894.735	907.913	889.365	0.600	0.57
	6	1364.04	1199.253	1353.748	1191.622	0.636	0.75
	7	1905.15	1725.247	1886.636	1709.563	0.909	0.97
	8	2536.44	2291.992	2505.461	2271.168	0.909	1.22

**Table 2.** Quantitative comparison between the analytical and numerical natural frequencies (in Hz).

beam theory to obtain consistent results by incorporation of the coupling between flexural and axial vibrations of the delaminated sublaminates of the model. Then later on, Mujumdar and Suryanarayan [1988] applied a pressure distribution between two respective delaminated sections to impose a constraint between the two beam sections in order to obtain similar flexural deformation. This kind of proposed model was termed a “constrained model” in contrast with the so-called “free model” proposed in [Wang and Liu 1982]. Their analytical model, demonstrated for isotropic materials, was found to have the ability to determine the natural frequencies of a delaminated beam at any interface. Della and Shu [2005] extended the model of [Mujumdar and Suryanarayan 1988] to composite beams, by using the effective bending stiffness terms of composite laminates. In what could be termed a similar approach, Pardoen [1989] developed a constrained model to predict the natural frequencies of a simply supported composite beam considering only midplane delaminations.

In this analytical study, the model first adopted by [Ramkumar and Kulkarni 1979] is customized for composite laminates, to determine the changes in natural frequencies due to delaminations located at different interfaces.

**2.4. Validation of the FE analysis.** A comparison of the analytical and ANSYS numerical results for undelaminated ( $F_{ui}$ ) and delaminated ( $F_{di}$ ) beams is shown in Table 2. It is observed that the natural frequencies decrease as delaminations occur and the percentage errors increase for the higher modes. The objective of this exercise is to verify the results of the numerical model. The numerical results are consistent with the analytical results with errors less than 1%. Numerical modeling is preferred to analytical modeling particularly in complex systems because it is relatively easy to incorporate different boundary conditions, loading configurations, etc.

### 3. Algorithms for solution of the inverse problem

**3.1. Optimization search algorithms.** Global search methods are search schemes that are not prone to being trapped by local optima. They require huge amounts of computational effort. Local search methods, on the other hand, restrict candidate potential solutions to a confined design space starting at an initial guess [Shang et al. 2001]. Thus, a combination of these is an attractive choice, which has led to the development of the *memetic algorithm*.

**3.1.1. Gradient-based local search.** The gradient-based local search (GBLS) method is a kind of optimization algorithm that basically employs gradient information of the objective function for determining the direction of the subsequent search points from a given start point. This technique uses a function which seeks the minimization of a scalar (single) objective function of multiple design variables within a region specified by linear constraints and bounds using the sequential quadratic programming (SQP) algorithm.

The SQP is the most famous gradient-based algorithm. The SQP method is most successful specifically for problems of nonmultimodal behavior. The GLBS used in this work is based on SQP. SQP methods embody the state of the art in nonlinear programming methods. An overview of SQP is found in [Schittkowski 1986]. In this method, the function solves a quadratic programming subproblem at each iteration by obtaining and updating iteratively at every step an estimate or approximation of the Hessian of the Lagrangian function using a quasi-Newton updating method based on the Broyden–Fletcher–Goldfarb–Shanno formula [Schittkowski 1986]. The method uses iteration from an initial guess until it reaches a feasible local optimum. During the process of every iteration, quadratic nonlinear programming is used to solve the objective at that point. If the problem is unconstrained, then the method reduces to Newton's method for finding a point where the gradient of the objective disappears. The solutions of the quadratic programming are used to initiate a search space towards a better solution until the optimum set is found.

However, the GLBS can be trapped in local optima and hence a reinitialization is used to avoid stagnation. Reinitialization is a process adopted to prevent to some extent the trapping of the technique at the local optimum by starting the optimization process from different start points. The final optimum may not be the global optimum, but this process effectively increases the optimization cycle time [Shang et al. 2001]. This technique has been implemented in MultiStart from the MATLAB Optimization Toolbox.

**3.1.2. Global optimizer based on evolutionary algorithms.** To ensure global optimum results and increase the accuracy of results while reducing the search time via surrogates, biologically inspired evolutionary algorithms (EA) are used. An EA is a kind of algorithm that encodes (both numerical and nonnumerical) design parameters to shuffle multiple candidates via parallel and interactive search. During the search process, there is firstly a selection performed based on survival of the fittest. To generate the next generation of possible candidate solutions, some parameter values are exchanged between two candidates (crossover) and new values introduced (mutation). EAs cannot be easily trapped in local minima or maxima as a result of crossover and mutation operations, which makes them an ideal method to effectively handle multimodal optimization problems.

EAs are therefore representatives of the class of stochastic (random) and robust optimization algorithms that do not require gradient information during the course of an optimization process but rather use an objective function value, which makes them more computationally expensive than the GBLS, which requires fewer iterations. However, they can handle a wide variety of problem characteristics such

as discrete and continuous design variables and multimodality, in contrast with the GBLs. Surrogates are used to contain their computationally expensive nature, hence exploring the advantage of their global optimization behavior.

The choice of an efficient optimization algorithm is based on the problem under study. For our optimization problem in which the variables are a combination of discrete and continuous, it is difficult to use conventional optimization algorithms such as the gradient-based method to obtain the global optimum, since they rely on the use of continuous variables. In an attempt to solve such discrete problems, a population-based stochastic algorithm, and in particular a nondominated sorting genetic algorithm (NSGA-II) [Deb et al. 2002], are found as an ideal choice to effectively manipulate the optimization task for detecting delaminations in composite laminates as they are the most promising in both discrete and continuous multimodal problems. While an evolutionary algorithm has been used in the current study, other forms of evolutionary algorithms such as particle swarms, differential evolution (DE), etc., can also be used. DE is peculiar to solving single objective optimization problems with continuous variables. Several DE variants exist such as the DE/rand/1/bin strategy and self-adaptive DE.

The parameters of NSGA-II and their values are listed in Table 3 while its flowchart is depicted in the authors' previous work [Ihesiulor et al. 2012a] (see Figure 1). The detailed description of NSGA-II processes includes:

- (1) Generation of initial population ( $G$ ): Generation of the first parent population of size  $N$ . This is randomly generated within the predefined feasible region (the upper and lower bounds of the design space).
- (2) Nondominated sort: Individual population are evaluated and sorted based on nondomination. A solution ( $s_1$ ) dominates (is preferred to) another solution ( $s_2$ ) if and only if  $s_1$  is better than  $s_2$  in the objective function. For every generation, fast nondominated sorting is applied to identify nondominated solutions to construct the nondominated front. This produces a set of candidate solutions that are nondominated by any individual in the population. These solutions are then discarded from the population temporarily until the next-best nondominated set is identified. This process goes on and on until all solutions are classified and assigned ranks equal to their nondomination level assuming fitness minimization.
- (3) Crowding distance: This is basically the niche safeguarding in the design objective space. The crowding distance assigned equals the front density in the neighborhood (the distance of each candidate solution from its nearest neighbors).
- (4) Selection by ranking: The initial population is sorted in ascending order according to fitness functions. Potentially better solutions are ranked higher than the worst solutions. Individuals are selected by the use of a binary tournament selection with the crowded comparison operator.
- (5) Genetic operators: A mating pool is formed by bonding of the parent and child populations. The simulated binary crossover operator and polynomial mutation are used to create an offspring of new population. Mutation occurs by random walk around individuals. The best population of parents and offspring with higher fitness is selected to reproduce the next generation.
- (6) Recombination and selection: Combination of the traits of the offspring population and parent population is done to reproduce the extended population of the next generation ( $2N$ ) [Ayala and dos

Maximum population size	$N = 200$
Crossover probability	$P_c = 0.9$
Mutation probability	$P_m = 0.1$
Maximum number of exact function evaluations	4000
Evolutionary operators (binary tournament selection)	$s = 2$
Evolutionary operators (polynomial mutation)	$\eta_m = 10$
Evolutionary operators (simulated binary crossover)	$\eta_c = 10$
Evolutionary operators (elitism, nondomination rank, crowding distance)	80
Number of independent runs (stochastic and can be run more than once)	$\geq 10$

**Table 3.** NSGA-II parameters.

[Santos Coelho 2008]. This is done to ensure elitism and to keep diversity in generating subsequent successive populations [Sastry et al. 2006]. The replacement criteria keeps the best among the parents and offspring based on the fast nondominated sorting and maintains the best diversified individuals to provide a larger search space. The new population size with the initial population is filled with individuals from the sorting fronts starting from the best. The crowding distance method is recalled to maintain diversity if a front incompletely fills the next generation. This ensures that convergence in one direction does not take place. This process repeats itself to generate subsequent generations until a stopping criterion is reached.

**3.2. Database creation for ANN training using *K*-means clustering algorithm.** The ANN model and its results are highly dependent on the training dataset provided. It is hence necessary to maintain the diversity of the training set to obtain a good prediction model by ensuring that the training data is not clustered around one part of the design domain. The most vital strategy in the selection of the training datasets is to find the ideal set that is a true representation of all the possible samples in the total design space. This is done using *K*-means clustering.

*K*-means clustering is used for grouping large datasets into smaller sets called clusters. The number of *k*-groups or objects needed is specified. Each object is represented by some feature vector in *n*-dimensional space, *n* being the number of all characteristics used to describe the objects to cluster. The algorithm then randomly chooses *k*-points in that vector space. These points serve as the initial centers of the clusters and all objects are assigned to the center they are closest to [MacQueen 1967]. Basically, the *K*-means clustering algorithm finds a subset of *k*-groups known as centroids that minimizes the mean squared distance from each data point to its nearest center in an entire dimensional space of *n*-data points [Kanungo et al. 2002]. The algorithm is implemented in the following steps:

- Step 1: Specify *k*-points into the space represented by the objects that are being clustered. These points represent the initial group centroids.
- Step 2: Each object is allocated to the group with the closest centroid.
- Step 3: When all objects have been assigned, the positions of the *k*-centroids are recomputed.
- Steps 2 and 3 are iterated until the centroids no longer move. This introduces a separation of the objects into groups from which the metric to be minimized can be computed.

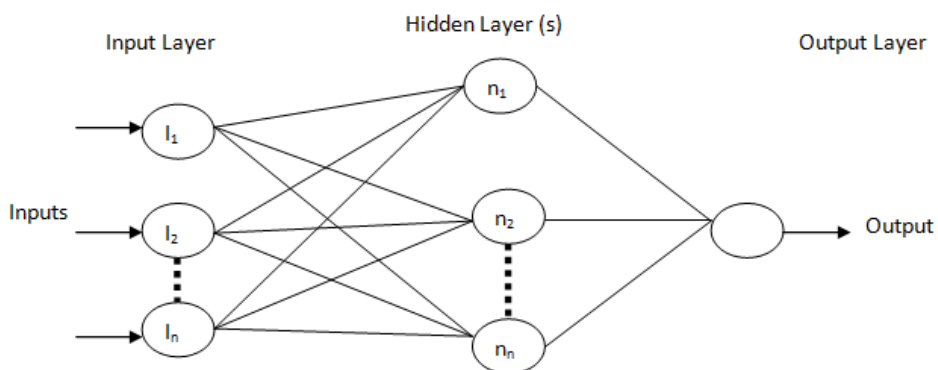
**3.3. Introduction to ANN as an inverse solver and surrogate creator.** Artificial neural networks (ANNs) have evolved as one of the promising artificial intelligence concepts used in real-world applications. ANN have been extensively used in structural engineering applications in the areas of failure prediction, delamination identification, crack detection, as well as others [Mahdi and El Kadi 2007]. ANN is a very powerful interpolator that can be used to map functions and derive a relationship between a set of input parameters and their output responses.

There are different types of ANN architectures, namely, multilayer perceptron (MLP), radial basis function (RBF), and so on. The MLP type [Delashmit and Manry 2005] is adopted in this study because it effectively provides a complex nonlinear mapping between the input and output variables. MLPs are feed-forward nets with one or more hidden layers between the input and output neurons as shown in Figure 2. ANN based on MLP is trained using a back propagation neural network (BPNN) algorithm, a gradient-based method that has emerged as successful in the training of multilayered neural nets using supervised learning. In supervised learning, the network learns using input and output data and provides an approximation of the functional mapping between the two.

A typical BPNN is based on the fact that a feed-forward neural net (FFNN) with at least one hidden layer can approximate any continuous nonlinear function with arbitrary accuracy the number of hidden neurons is sufficient. MLP is a FFNN which consists essentially of an input layer with several neurons (depending on the number of inputs,  $dF_i$  to  $dF_n$ ), a layer of output neurons and one or more layers of hidden neurons that wholly perform the application expected objective. The primary building blocks of MLP are the artificial neurons or processors. The neurons in each layer are fully interconnected to the preceding and subsequent layers. Every one of them is connected by adjustable associated weights to enable the network to map complex associations between the input and output data. The activation function ( $a$ ) for each neuron is defined as the summation of all the inputs multiplied by their connection weights and biases ( $wt_1, wt_2, \dots, wt_n$ ), given as

$$a = dF_1wt_1 + dF_2wt_2 + \dots + dF_nwt_n. \quad (5)$$

The activation function is transmitted through a link to other neurons via a feed-forward network design and then fed to a transfer function which could be linear or nonlinear to generate the output. The structure of the network is a function of the interaction between these neurons. Functions such as



**Figure 2.** A schematic framework of one-hidden-layer architecture.

the sigmoidal and radial (Gaussian) are used to build the neuron activity. A tangent-sigmoidal transfer function is used in the hidden layer and to avoid limiting the output to a small range, a linear transfer function is employed in the output layer. A Bayesian regularization (*trainbr*) back-propagation learning algorithm is utilized to speed up the convergence of the MLP model. Bayesian regularization is a robust iterative training algorithm that learns patterns based on input and output data; it essentially provides stronger and more efficient generalization ability by regularization. The ANN training process is stopped when a maximum number of 1000 epochs is reached. The error is measured based on the root-mean-square error (RMSE) between the predicted values and the output for all elements in the training and testing set. The networks are initialized to return neural network nets with weight and bias values updated according to the network initialization function.

**3.3.1. Pros of ANN.** The most vital advantages of ANN include:

- Its capability to model and map complex nonlinear systems (nonlinearity) by deriving a relationship between a set of input and output responses (input-output mapping).
- The ability to learn which allows the network to adapt to changes in the surrounding environment (adaptivity).
- Once ANNs are properly trained, damage identification is relatively fast and mathematical models do not need to be constructed.
- There are no limitations on the type of vibration parameters to be used as inputs for ANNs. The input and output parameters can be selected with much flexibility without increasing the complexity of network training.

**3.3.2. Cons of ANN.** A major demerit of ANNs is that the resulting weights and nets of the trained network are difficult to interpret (that is, an inability to obtain adequate solutions of complex problems with physical mathematical methods in contrast to RSM). Others include:

- It is difficult to find an appropriate network architecture.
- It usually suffers from the problems of underfitting (inaccurate approximation of the training data) resulting from too-small networks, and overfitting (inadequate generalization) due to too-large networks.
- It requires several networks of different architectures to be trained, and their performance compared on a separate set of test data to estimate their generalization properties.
- The training data of the database should be large enough to have a close relationship with the associated parameters (that is, sufficient training data for complex ANNs are necessary, requiring availability of a large database).

**3.3.3. Modifications to the basic ANN.** Extensive work has been done on the basic ANN to improve its generalization ability. The following are some contributions made to the ANN model to improve its generalization capability, accuracy of approximation, output variable handling, and training time.

- Single and ensemble nets: The capabilities of ANN to handle multidimensional outputs are well known. However, it is preferred to use individual ANN models to predict each output because it is not feasible to use ensemble (multiple) nets to predict all the output variables. Training single nets

enhances the prediction accuracy because only one output is predicted. This improves the handling capability of the optimizers when coupled with surrogates during the optimization process.

- The concept of the  $K$ -means clustering algorithm introduced enables one to effectively reduce the size of dataset required for ANN training and hence determine the minimum amount of datasets just sufficient for ANN training.
- The adoption of the *trainbr* training algorithm showed great effectiveness in the ANN results. Generalization enhancement by Bayesian regularization produces a network that performs well with the training data and exhibits smoother behavior when presented with new data.
- Increasing the number of hidden neurons and layers. Larger numbers of neurons in the hidden layer can give the network more flexibility because the network has more parameters it can optimize. However, the number of layers should be increased gradually because large hidden layers lead to undercharacterization of the network since the network must optimize more parameters than there are data vectors to constrain these parameters.
- Adoption of network architectures with tansig and purelin transfer functions yields better results for low RMSE values and high  $R^2$  values than other possible combinations tried.
- Preprocessing of the network inputs and targets improves the efficiency of the neural network training. It basically reduces the dimensions of the input vectors to increase network performance. This is essentially the normalization or scaling of inputs and targets so that they fall in the range  $[-1, 1]$ . However, it was seen that scaling had no great significant effect on the network performance for the proposed problem under consideration.

**3.3.4. Steps in designing ANN model.** The ANN training process is not an easy task and involves finding an appropriate ANN model for a given problem. Hence, the necessary requirements for a successful ANN development include: a sufficient database, careful selection of the parameters, network architecture (number of hidden layers, number of hidden and output nodes), transfer functions, and effective training and learning algorithms. These choices completely depend on the approximation function.

The process of building an ANN model can hence be outlined and summarized as follows:

- Selection of the number of input and output variables of the neural network and database creation (the dataset obtained from FEM analysis after a  $K$ -means clustering is divided into a training set and a testing set).
- Determination of network architecture by trial and error method (that is, number of hidden layers and number of hidden nodes — information processing occurs at many simple elements called neurons) required to generalize the design space.
- Selection of the training algorithm and functions and measure of its effectiveness based on RMSE performance.
- Invoking of a back-propagation algorithm to train multilayer feed-forward networks with differentiable transfer functions to perform function approximation. Training by back propagation is described as the process by which derivatives of network error with respect to network weights and biases are computed. This process is subdivided into:



- Feed forward of the input training data (input vector of set of design variables from the input layer is fed forward by propagation to the hidden layer and subsequently to the output layer) in a process termed the forward pass of the back propagation algorithm.
- The computation and back propagation of the associated weight and errors. The predicted values of the output layer are compared with the target values. The error between the predicted and target values is calculated and propagated back toward the hidden layer in a process known as the backward pass of the back propagation algorithm. The error is used to update the weight matrices between the input-hidden layers and hidden-output layers.
- The mean square error of the network is computed by calculating the amount between the predicted and target values. Consequently, this error is minimized by a predetermined training algorithm using optimization algorithms based on a gradient-based back-propagation process which repeatedly changes the performance values depending on the network connection weights. The *trainbr* function is used for this purpose. This function trains the network by randomly initializing and updating the weights and bias values according to a Bayesian regularization algorithm.
- Terminating the network training using 1000 maximum epochs to reduce the effect of random weights on training the network. This method stops the training when the maximum number of training cycles given is reached.
- Computing the average performance of each of the network architecture with RMSE as the performance criterion and selecting the best network performance with the minimum RMSE.
- The best model is applied to the test data to investigate the performance on an unseen set of data.

**3.3.5. ANN configuration to analyze the best network architectures.** Choosing the number of hidden layers and neurons in the hidden layer is also a demanding task and it is the principal limitation of ANN. Since the ANN configuration has a great influence on the predictive output, various arrangements have been considered. It is essential to designate a formula to describe the ANN configuration as  $(I-n_1-n_2-O)$ . For example, 2-20-1 means a one-hidden-layer ANN with two input parameters and one output parameter, with the hidden layers containing 20 elements (neurons); 3-10-10-2 denotes a three-input and two-output ANN, with ten neurons in two hidden layers. The MATLAB neural network toolbox has been used as the basis in which the networks can be configured in a very wide variety of architectures, and the training algorithms can be also chosen with ease.

Evaluation of the network performance is measured based on RMSE analysis and coefficient of determination ( $R^2$ ). The power of prediction can be quantified by the RMSE of the predicted output from test data. The smaller the RMSE of the test dataset is, the higher the predictive capability of the network.

The RMSE (that is, the root mean square of the differences between the actual and predicted values, which should be very close to zero) is expressed as

$$\text{RMSE} = \sqrt{\text{mean}((R_A - R_P)^2)}, \quad (6)$$

where  $R_A$  and  $R_P$  are the actual and predicted values from the network.

Similarly,  $R^2$  (that is, a measure of how well the variation in the output is explained by the targets; if

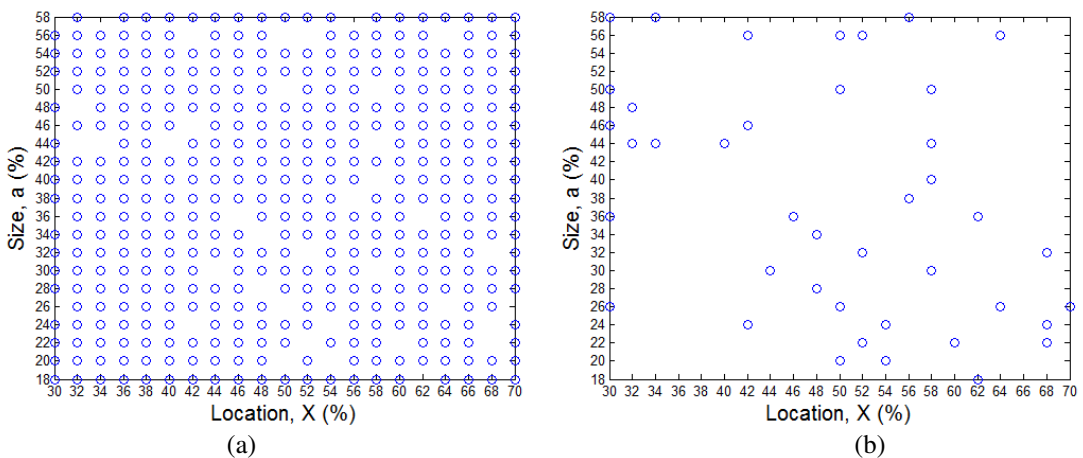
this number is equal to 1 there is a perfect correlation between the targets and outputs) is given by

$$R^2 = \frac{\sum(R_A - \bar{R}_A)(R_P - \bar{R}_P)}{\sqrt{\sum(R_A - \bar{R}_A)^2(\sum(R_P - \bar{R}_P))^2}}, \quad (7)$$

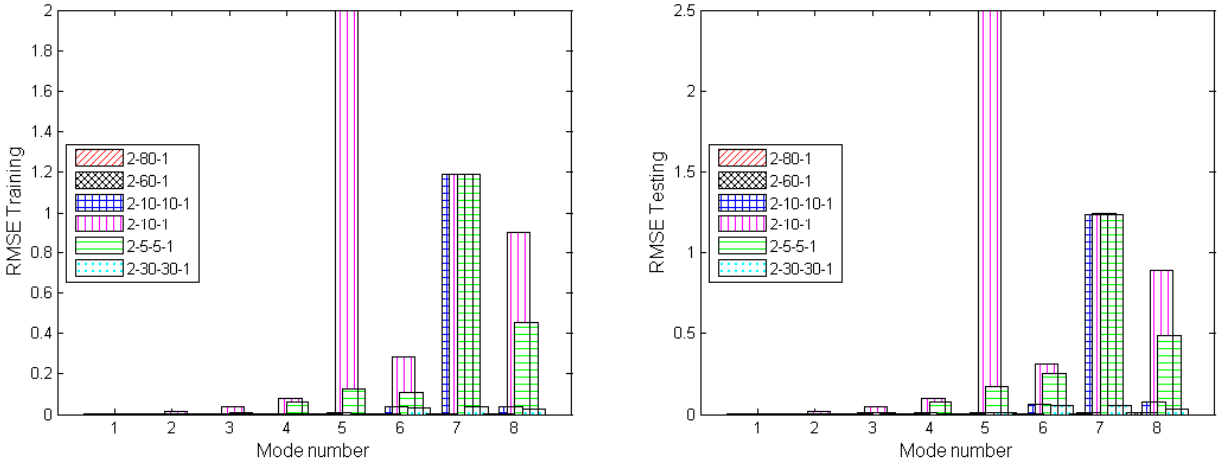
$\bar{R}_A$  and  $\bar{R}_P$  are defined as the means of the actual and predicted values from the network.

**3.3.6. Training performance of trial neural networks for the forward problem.** Several neural network architectures were tried since there is no laid down rule for choosing the optimal number of hidden layers and processing elements for each layer; it is basically problem dependent. However, incorporation of more processing elements enhances the generalization capability of the network for a larger number of training data points. The motivation was to determine the best network with the least RMSE and good coefficient of determination ( $R^2$ ) irrespective of the time. In other words, the length of computation time was not considered as a factor for choosing the best network. However, larger network architecture takes longer running time in all analysis, and smaller networks take a shorter time. It is shown that when enough training datasets have been used, the root mean square of the output error converges to zero. The neural network optimum network architecture in most cases consisted of one input layer, one or two hidden layers, and one output layer, with the circles representing processing elements or neurons in Figure 2. For the inverse problem, the eight inputs were the differences between the damaged beam and the undamaged beam for the first eight natural frequencies and the outputs were the interface, delamination size, and location, and vice versa for the forward problem. The hidden layers contained a maximum of 80 neurons.

For data generation, considering only midplane delaminations, 441 FE models equally spaced at gaps of 2% were run in a batch process for two hours assuming normalized delaminations located from 30% to 70% (30:2:70) of the total beam length and having normalized sizes ranging from 18% to 58% at gaps of 2% (18:2:58). The 441 generated datasets were randomized using  $K$ -means clustering. As shown in Figure 3(a), 90.7% (400) of the 441 simulations run in ANSYS were used for training a neural network



**Figure 3.** Design space for training and testing datasets. (a) 400 training datasets and (b) 41 testing datasets.



**Figure 4.** Comparison of performance of different network architectures on RMSE of training data (left) and testing data (right).

model and the remaining 9.3% (41) of the dataset were used for testing of the network as shown in Figure 3(b).

Then, consider the solution of a forward problem trained with ANN using different network architectures. Figure 4 demonstrates the trial performance of different network architectures with their training and testing RMSE values for the individual eight percentage frequency changes (individual nets) taken as output to the network with two inputs (location and size of delamination). It is shown that 2-80-1 is the best network with the least RMSE. It is evident that high network architectures give better results than low network architecture for both training and testing datasets.

**3.3.7. Training performance of trial neural networks for the inverse problem.** For the inverse problem where the eight natural frequencies are taken as input to the network and delamination location and size as network outputs, high network architectures are also found to give better results, in terms of low RMSE and high values of  $R^2$ , than small network architectures.

**3.4. Surrogate approach.** In the surrogate approach, ANN is used to build surrogate models for approximation of the computationally expensive FE models by solving the forward problem. This approach is solving the forward problem to approximate the output of the natural frequencies simulated using the FE models from the given corresponding delamination parameters. Later, these surrogates are used in the optimization loop instead of direct optimization via FE models as shown in Figure 1. The surrogate approach is extensively employed as an inexpensive approximation of the true function evaluations instead of the computationally expensive FEM simulations. Surrogates are especially interesting for expensive objective functions, since the necessary computational effort to build the surrogate is smaller than the effort of the objective function evaluation.

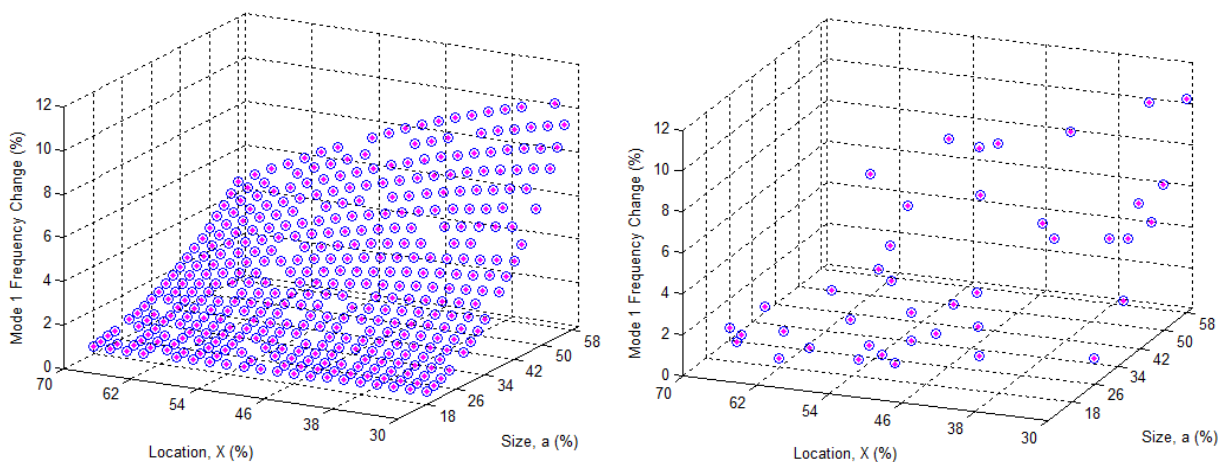
ANN, RBF, RSM, Kriging, and design and analysis of experiments are some examples of surrogate models [Sul et al. 2011]. The choice of surrogate models depends on the problem under consideration. In the current work ANN is to provide surrogate models to computationally expensive FE models and is found to be very effective. This justifies the need for surrogate models in reducing the optimization cycle

$[Z, X, a]$	Mode	FE	ANN	% error	$[Z, X, a]$	Mode	FE	ANN	% error
[1, 54, 24]	$dF_1$	0.8013	0.8026	0.1711	[1, 62, 18]	$dF_1$	0.4049	0.4044	0.1312
	$dF_2$	0.3182	0.3179	0.0831		$dF_2$	0.5266	0.5266	0.0075
	$dF_3$	10.6770	10.6769	0.0006		$dF_3$	2.6310	2.6310	0.0013
	$dF_4$	5.6542	5.6545	0.0050		$dF_4$	8.9072	8.9068	0.0039
	$dF_5$	14.5527	14.5530	0.0020		$dF_5$	2.1741	2.1740	0.0017
	$dF_6$	12.5625	12.5636	0.0083		$dF_6$	12.0654	12.0639	0.0126
	$dF_7$	13.9558	13.9579	0.0155		$dF_7$	9.4308	9.4367	0.0627
	$dF_8$	13.3759	13.3778	0.0134		$dF_8$	9.3618	9.3599	0.0199

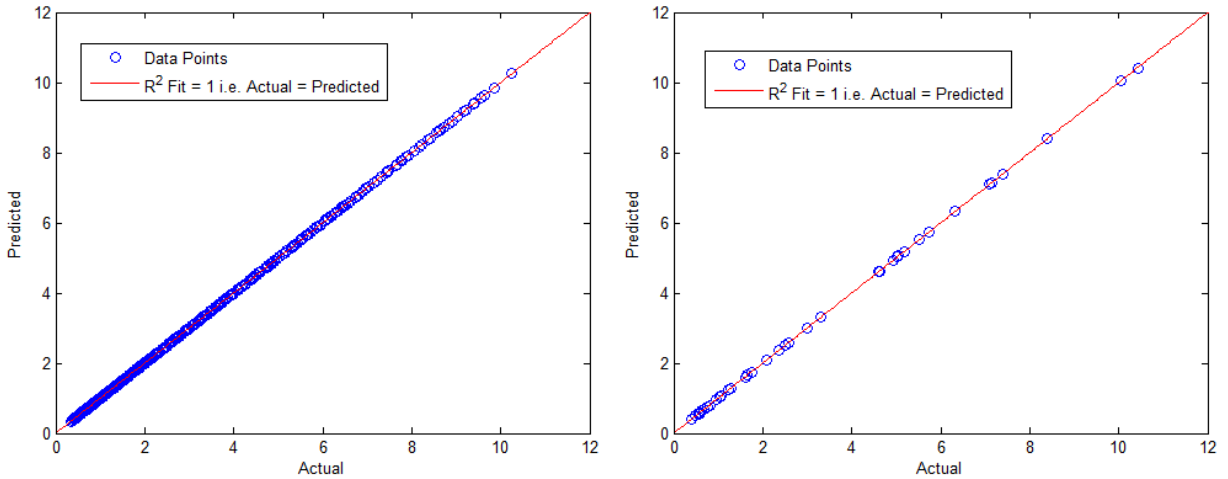
**Table 4.** Comparison of  $dF_i$  between FE simulations and ANN approximations.

time and exploring the complete design space with minimal computational cost. RSM was first tried, and failed to consistently provide a true approximation model due to high multimodality of the delamination detection problem in higher modes, say modes 4–8. Since ANN was found to perform very well for the solution of the problem under consideration, it was selected for the unique purpose of building surrogate models to expensive FE models [Ihesiulor et al. 2012b]. Other surrogate models were not tried.

**3.4.1. Validation of the surrogate model.** A validity check of our surrogate model is necessary to prevent misleading of the search by the optimizers due to poor approximations. To validate the performance of the trained ANN, a perfect match of true function evaluation by ANSYS and the approximated functions by ANN is shown in Table 4 for delamination signatures [1, 54, 24] and [1, 62, 18] for the first eight percentage changes in frequencies. The results are shown to have negligible error of not more than 0.17%. Also, perfect fits plot and regression ( $R^2$ ) plots for the 400 training and 41 test datasets between the simulated percentage change in frequencies and the predicted output from the neural network training for mode 1 are shown in Figures 5 and 6, respectively.



**Figure 5.** Simulated percentage change in natural frequencies: actual values (circles) versus predicted output (red stars) from the ANN model for mode-1 training (left) and testing (right) datasets.



**Figure 6.** Simulated percentage change in natural frequencies:  $R^2$  plot between actual and the predicted output from the ANN model for mode-1 training (left) and testing (right) datasets.

**3.5. Introduction to RSM.** Response surface methodology (RSM) can be defined as development of the mathematical and statistical techniques applied in the modeling and analysis of engineering problems in which the output of interest is governed by some input variables and the key objective is to optimize this output response [Montgomery 2005]. RSM is essentially a statistical method that employs quantitative data from appropriate simulations or experiments to determine and solve simultaneously multivariate equations.

RSM, also known as polynomial fitness function modeling, adopts regression curve fitting to obtain mathematical approximations of responses of a given system as functions of some input design variables [Todoroki 2001]. This method is widely employed as an inexpensive low-order approximation model instead of the more time-consuming but accurate calculations using FEM simulations. Response surfaces can easily be fitted to data by least-squares approach. The order of the polynomial is important; quadratic or cubic polynomials are mainly used, with quadratic polynomials best suited for continuous, unimodal problems. RSM models are widely used in polynomial approximation schemes due to their flexibility and ease of use. In the polynomial approximation method, the response surface model is a polynomial of  $n$ -th degree whose coefficients are determined from a linear system of equations. The linear system is solved using least-square minimization of the error between the predicted and actual values.

**3.5.1. Advantages of RSM.** In the current study, RSM is adopted as another inverse-problem solver because of the following benefits it enjoys over ANN:

- (1) It does not require too much computational effort and resources to generate its mathematical models (ease of calculations and use).
- (2) Its solution to the inverse problems can be approximately solved without the constraints of modeling.
- (3) Its approximation model can be easily validated through statistical means.

**3.5.2. Analysis of a first-order model response surface.** In RSM, the factors that are considered as most important are used to build a polynomial model in which the independent variable is response from experiments or numerical simulations. A first-order multiple regression model with  $N$  simulation runs carried out on  $k$  input variables ( $N > k$ ) and an output response  $R$  can be expressed as

$$R_i = C_0 + \sum_{j=1}^k C_j dF_{ij} + \epsilon_i \quad (i = 1, 2, \dots, N). \quad (8)$$

The response  $R_i$  is a function of the input variables  $dF_1, dF_2, \dots, dF_n$ , plus the error. The  $C_j$  are the regression coefficients, and  $dF_{ij}$  corresponds to the  $i$ -th sample and  $j$ -th independent variable.

Equation (8) can be expressed in matrix form as

$$R = dF * C + \epsilon, \quad (9)$$

$$\underbrace{\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix}}_R = \underbrace{\begin{bmatrix} 1 & dF_{11} & dF_{12} & \cdots & dF_{1k} \\ 1 & dF_{21} & dF_{22} & \cdots & dF_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & dF_{N1} & dF_{N2} & \cdots & dF_{Nk} \end{bmatrix}}_{dF} \underbrace{\begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_k \end{bmatrix}}_C + \underbrace{\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}}_\epsilon.$$

Invoking the least-square error method, the estimated coefficient  $\hat{C}$  of the coefficient vector ( $C$ ) can be given as

$$\hat{C} = (dF^T dF)^{-1} dF^T R. \quad (10)$$

**3.5.3. Analysis of a second-order model response surface.** The presence of high curvature in the response surface makes the first-order models unsuitable for complex problems. A second-order model becomes handy in approximating the true response surface. The second-order model accommodates all the terms in the first-order model, in addition to quadratic terms ( $C_{11} dF_{1i}^2$ ) and all cross-product terms ( $C_{13} dF_{1i} dF_{3j}$ ). The method of least squares can also be applied to estimate the coefficients in the model. The equation based on a second-order polynomial is given by

$$R_i = C_0 + \sum_{j=1}^k C_j dF_j + \sum_{j=1}^k C_{jj} dF_j^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^k C_{ij} dF_i dF_j + \epsilon. \quad (11)$$

Hence, to develop the relationship between the variations in the simulated natural frequencies and the corresponding size and location of delamination, response surfaces of fourth-degree polynomials were generally adequate in this study.

The RSM procedure described above is employed to fit a fourth-order polynomial equation using the simulation data where 400 FE model simulations are performed and 41 datasets used for testing. From the Minitab output, the fourth-order polynomial equation for predicting location and size is given below, where the output responses are the delamination location and size and the inputs are the first percentage changes in natural frequencies. The least-square error method is adopted to obtain the unknown coefficients of the polynomials. All the insignificant interaction terms are removed from the models using an ANOVA table.

**3.5.4. Validation and adequacy check of the developed models.** In order to ascertain the adequacy and goodness of the developed response surface approximation models, the coefficient of determination ( $R^2$ ) as already defined and the absolute average deviation (AAD) are adopted. The efficiency of the model in terms of its predictive power can be determined by both  $R^2$  and AAD, because  $R^2$  alone cannot be effectively used to measure the performance of the developed models. The  $R^2$  is basically a measure of how well the variation in the output is explained by the targets; if this number is equal to 1 there is a perfect correlation between targets and outputs. However, a large value of  $R^2$  does not necessarily imply that the regression model is a good one [Bas and Boyaci 2005]. Thus, it is possible for models that have large values of  $R^2$  to yield poor predictions of new observations or estimates of the mean response. Plotting actual results versus predicted results from the model gives a straight line passing the origin with an angle of  $45^\circ$ , but in practical cases the model fails to give accurate results to new data. This limitation of  $R^2$  is eliminated by using AAD analysis, which is a direct method for describing the deviations in the actual and predicted outputs by the models [Bas and Boyaci 2005]. Minitab was used to conduct all analysis.

The AAD is calculated by

$$\text{AAD} = \sum_{i=1}^N (|\text{abs}(R_A - R_P)/R_A|/N) * 100. \quad (12)$$

The expression for  $R^2$  is given in (7). Also, adjusted  $\bar{R}^2$  is defined as the improvement of  $R^2$  when the number of terms in a model is adjusted, which is and lower than the  $R^2$  value. It is given as

$$\bar{R}^2 = 1 - \frac{1 - R^2}{N - 1/N - k - 1} \quad (13)$$

where  $R_A$  and  $R_P$  are the actual and predicted responses, and  $N$  is the number of simulation runs.

Evaluation of the  $R^2$  and AAD values together was just adequate to check the accuracy of the developed models. The  $R^2$  must be close to 1 and the AAD between the predicted and actual output must be as small as possible tending towards 0. Acceptable values of  $R^2$  and AAD mean that the model equations define the true behavior of the system and they can be used for interpolation in the simulation design space.

From the equation shown in Table 5 for  $X$ , the  $R^2$  and adjusted  $R^2$  values for delamination location ( $X$ ) are calculated as 99.3% and 99.2%, respectively. The near-perfect prediction ( $R^2$  value) of the model is an indicator that the model generated has been perfected to fit the given data and thus is highly significant. The high  $R^2$  is a good indication of the predictive power of the developed model. Similarly, from the equation shown in Table 6 for  $a$ , the  $R^2$  and adjusted  $R^2$  values for the delamination size ( $a$ ) are deduced to be 100% and 100%, respectively. This shows that the developed model for delamination size has more predictive power in terms of accuracy of prediction results than the model for delamination location.

Also, from the equations shown in Tables 5 and 6, the calculated AAD for the delamination location and size are, respectively, obtained as 2.78% and 0.37%. This indicates that the fourth-order polynomial model for delamination size  $a$  is highly significant and adequate to represent the actual relationship between the response and the significant input variables, with very small AAD value (0.37%) and a satisfactory coefficient of determination ( $R^2 \approx 1$ ).

$$\begin{aligned}
&78.1 + 28.1 dF_1 - 29.6 dF_2 + 11.1 dF_3 + 16.5 dF_4 - 7.65 dF_5 - 1.74 dF_6 + 7.90 dF_7 + 6.22 dF_8 \\
&+ 6.62 dF_1 dF_2 + 12.2 dF_1 dF_3 - 0.647 dF_1 dF_4 - 0.289 dF_1 dF_5 + 1.79 dF_1 dF_6 - 1.70 dF_1 dF_7 \\
&+ 1.05 dF_1 dF_8 - 1.90 dF_2 dF_3 + 0.642 dF_2 dF_4 + 0.279 dF_2 dF_5 + 0.253 dF_2 dF_6 + 0.591 dF_2 dF_7 \\
&- 0.0537 dF_2 dF_8 - 0.0150 dF_3 dF_5 + 0.546 dF_3 dF_6 + 0.284 dF_3 dF_7 - 0.170 dF_3 dF_8 \\
&- 0.234 dF_4 dF_5 - 0.0908 dF_4 dF_6 - 0.150 dF_4 dF_7 - 0.0582 dF_4 dF_8 - 50.5 dF_1^2 \\
&- 0.015 dF_2^2 - 1.21 dF_3^2 + 0.666 dF_4^2 + 0.314 dF_5^2 + 0.274 dF_6^2 - 0.015 dF_7^2 \\
&- 0.087 dF_8^2 + 2.14 dF_1^3 + 0.0324 dF_2^3 + 0.0286 dF_3^3 - 0.00935 dF_4^3 - 0.0156 dF_5^3 \\
&- 0.0186 dF_6^3 + 0.00317 dF_7^3 + 0.0029 dF_8^3 - 0.0570 dF_1^4 - 0.00114 dF_2^4 - 0.000677 dF_3^4 \\
&+ 0.000030 dF_4^4 + 0.000114 dF_5^4 + 0.000264 dF_6^4 - 0.000075 dF_7^4 - 0.000055 dF_8^4
\end{aligned}$$

**Table 5.** Fourth-order polynomial equation for generated delamination location ( $X$ ) in terms of input percentage change in natural frequencies.

$$\begin{aligned}
&13.8 - 6.16 dF_1 + 1.23 dF_2 + 0.667 dF_3 + 0.214 dF_4 + 0.0664 dF_5 - 0.0667 dF_6 \\
&- 0.276 dF_7 + 0.541 dF_8 + 0.247 dF_1 dF_2 - 0.0635 dF_1 dF_3 + 0.0177 dF_1 dF_4 \\
&+ 0.0152 dF_1 dF_5 + 0.0530 dF_1 dF_6 - 0.0268 dF_1 dF_7 - 0.168 dF_1 dF_8 \\
&+ 0.00728 dF_2 dF_3 + 0.0328 dF_2 dF_4 - 0.0118 dF_2 dF_5 - 0.00396 dF_2 dF_6 \\
&+ 0.00520 dF_2 dF_7 + 0.0443 dF_2 dF_8 + 0.00523 dF_3 dF_5 - 0.00468 dF_3 dF_6 \\
&+ 0.00086 dF_3 dF_7 + 0.0228 dF_3 dF_8 - 0.00051 dF_4 dF_5 - 0.00669 dF_4 dF_6 \\
&+ 0.00139 dF_4 dF_7 - 0.00449 dF_4 dF_8 + 0.733 dF_1^2 - 0.213 dF_2^2 - 0.00885 dF_3^2 \\
&+ 0.0203 dF_4^2 + 0.0339 dF_5^2 + 0.0289 dF_6^2 + 0.0547 dF_7^2 - 0.0752 dF_8^2 - 0.0746 dF_1^3 \\
&+ 0.00782 dF_2^3 + 0.000083 dF_3^3 - 0.00118 dF_4^3 - 0.00184 dF_5^3 - 0.00128 dF_6^3 \\
&- 0.00289 dF_7^3 + 0.00389 dF_8^3 + 0.00197 dF_1^4 - 0.000149 dF_2^4 + 0.000020 dF_3^4 \\
&+ 0.000020 dF_4^4 + 0.000034 dF_5^4 + 0.000021 dF_6^4 + 0.000047 dF_7^4 - 0.000061 dF_8^4
\end{aligned}$$

**Table 6.** Fourth-order polynomial equation generated for delamination size ( $a$ ) in terms of input percentage change in natural frequencies.

For the response surface model for delamination location  $X$ , it was found that the low AAD was obtained due to high nonlinear curvature in the delamination location with respect to the percentage change in frequencies.

#### 4. Comparison of delamination prediction efficiencies of different algorithms

The approach or algorithm one uses will have a great effect on the accuracy of results. In solving the problem of delamination detection in composite laminates, different algorithms have been employed as seen in the previous sections and in [Ihesiulor et al. 2012a]. Hence, it is worthwhile to do a comparative analysis of these algorithms to ascertain the most effective and efficient in terms of accuracy of prediction. Efficiency of algorithms with respect to the minimization optimization algorithms can be measured as the minimum time needed to lower the error below a certain specified value associated with the value of the objective function after a given number of runs. The efficiency of the ANN and RSM algorithms can be defined as the measure of performance in terms of RMSE and AAD, respectively. Accordingly,



Algorithm	Name
1	Artificial neural network (ANN)
2	Response surface method (RSM)
3	GBLS with surrogates (GBLS <sub>WoS</sub> )
4	GBLS without surrogates (GBLS <sub>WS</sub> )
5	NSGA-II with surrogates (NSGA-II <sub>WS</sub> )
6	NSGA-II without surrogates (NSGA-II <sub>WoS</sub> )

**Table 7.** List of proposed algorithms.

one is only interested in the algorithm that gives the least prediction error regardless of the time, but the evaluation time should not be too large before expected results are evaluated for effective use in online SHM. With these factors under consideration, a comparative analysis is made in this section between four key algorithms (ANN, RSM, GBLS, and NSGA-II) under six different approaches (ANN, RSM, GBLS<sub>WS</sub>, NSGA-II<sub>WS</sub>, GBLS<sub>WoS</sub>, and NSGA-II<sub>WoS</sub>) as shown in Table 7, where GBLS<sub>WS</sub> and GBLS<sub>WoS</sub> are GBLS methods with and without surrogates, respectively. Similarly, NSGA-II<sub>WS</sub> and NSGA-II<sub>WoS</sub> are NSGA-II methods with and without surrogates, respectively.

So far we have shown that most algorithms are fast and contains several capabilities, such as:

- Ease of use and implementation and allowing a user to specify input data for which the delamination signature is to be ascertained easily (RSM).
- Generation of random results up to a significant number of runs and selection of the best result in terms of the minimum objective function value (NSGA-II).

For this study, in terms of dataset generation, the dataset scenario described earlier containing 400 training datasets and 41 test datasets was considered for this comparative analysis and two hours was required to generate the database. The training data used for this comparison was simulation data benchmarked at 400 datasets for all algorithms and ten damage cases, as shown in Table 8, out of the 41 reserved tests were selected for the performance comparative study. This comparison was based on the two-variable problem, that is, predicting delamination location and size at midplanes.

**4.1. Algorithm 1: ANN.** The ANN model used in the benchmark tests had 8 input nodes, 80 hidden nodes, and 2 output nodes (that is, a network architecture of 8-80-2). The training data contained 400 input and output vector pairs. The training time for the ANN algorithm was measured to be about 133 s. Table 9 gives the prediction results of the selected ten cases using ANN. It is seen that ANN gives for

S/N	Delamination signature $[X, a]$	S/N	Delamination signature $[X, a]$
1	[50, 26]	6	[58, 40]
2	[68, 24]	7	[48, 34]
3	[50, 20]	8	[58, 30]
4	[30, 36]	9	[48, 28]
5	[34, 44]	10	[62, 36]

**Table 8.** Selected test cases to ascertain the method with best performance.

S/N	Actual $[X, a]$	Predicted $[X, a]$	% ( $[(E - X), (E - a)]$ )
1	[50, 26]	[49.9946, 26.0000]	[0.0109, 0.0001]
2	[68, 24]	[68.0095, 23.9990]	[0.0139, 0.0042]
3	[50, 20]	[50.0068, 19.9999]	[0.0136, 0.0003]
4	[30, 36]	[30.0009, 36.0000]	[0.0031, 0.0001]
5	[34, 44]	[33.9994, 44.0000]	[0.0019, 0.0001]
6	[58, 40]	[57.9997, 40.0004]	[0.0006, 0.0011]
7	[48, 34]	[48.0056, 34.0001]	[0.0116, 0.0002]
8	[58, 30]	[57.9919, 30.0003]	[0.0139, 0.0011]
9	[48, 28]	[47.9980, 27.9999]	[0.0043, 0.0004]
10	[62, 36]	[61.9995, 36.0005]	[0.0008, 0.0013]
Total			[0.0745, 0.0088]

**Table 9.** Percentage errors of ten test cases using ANN modeling.

the ten cases total percentage errors of 0.07% and 0.009% in location and size, respectively. This shows that ANN is very unique in performance, giving a maximum error of 0.014% in all the cases under consideration.

**4.2. Algorithm 2: RSM.** Invoking the RSM regression models developed in the equations shown in Tables 5 and 6 using 400 datasets for delamination location and size, respectively, the prediction results of the ten selected cases are shown in Table 10. Results show that RSM gives adequate approximations as an inverse tool using the variations in natural frequencies for delamination detection. It is seen that RSM can be used to successfully predict delamination location and size. The test data helps to establish that a model that closely matches the actual values is developed and it is still a mathematical fit over the training data. It is seen that the results in predicting delamination size are more accurate than for delamination location because the percentage change in frequency increases monotonically with increase

S/N	Actual $[X, a]$	Predicted $[X, a]$	% ( $[(E - X), (E - a)]$ )
1	[50, 26]	[50.0724, 25.9779]	[0.1448, 0.0850]
2	[68, 24]	[69.0697, 23.9806]	[1.5731, 0.0808]
3	[50, 20]	[50.4786, 19.9550]	[0.9572, 0.2250]
4	[30, 36]	[29.9083, 36.1363]	[0.3057, 0.3786]
5	[34, 44]	[35.1090, 44.2064]	[3.2618, 0.4691]
6	[58, 40]	[59.3110, 40.0181]	[2.2603, 0.0452]
7	[48, 34]	[48.1630, 34.0942]	[0.3396, 0.2771]
8	[58, 30]	[58.4583, 29.9824]	[0.7902, 0.0587]
9	[48, 28]	[47.9457, 27.9744]	[0.1131, 0.0914]
10	[62, 36]	[62.8340, 35.9788]	[1.3452, 0.0589]
Total			[11.0909, 1.7698]

**Table 10.** Percentage errors of ten test cases using RSM modeling.

S/N	Actual $[X, a]$	Minimum <sub>Obj</sub>	Predicted $[X, a]$	% $([(E - X), (E - a)])$
1	[50, 26]	0.09550	[49.6446, 25.4675]	[0.7108, 2.0482]
2	[68, 24]	0.00553	[68.0706, 23.8523]	[0.1039, 0.6156]
3	[50, 20]	0.00852	[50.0634, 19.4827]	[0.1268, 2.5863]
4	[30, 36]	1.56888	[31.3309, 36.1582]	[4.4363, 0.4396]
5	[34, 44]	0.69202	[34.6365, 43.4115]	[1.8721, 1.3376]
6	[58, 40]	0.73237	[57.8042, 38.9887]	[0.3376, 2.5282]
7	[48, 34]	0.00528	[48.0291, 33.2507]	[0.0606, 2.2037]
8	[58, 30]	2.82593	[42.2571, 30.0895]	[27.1428, 0.2983]
9	[48, 28]	0.00543	[48.0293, 27.3201]	[0.0610, 2.4284]
10	[62, 36]	0.00481	[62.1006, 35.6956]	[0.1622, 0.8456]
Total				[35.0140, 15.3315]

**Table 11.** Percentage errors of ten test cases using NSGA-II<sub>woS</sub>.

in delamination size whereas for the delamination location versus percentage frequency change, the frequency changes increase with a very sharp curvature resulting in a high nonlinear complex function. Hence, it can be deduced that the total sums of errors for the ten test cases are given to be 11% and 2% for delamination location and size, respectively. However the maximum error for individual cases is shown to be 3% and 0.5% in delamination location and size, respectively. These results suggest that the performance of the RSM algorithm results is quite satisfactory. This approach also requires less time for computation, less than 5 s. The only significant time involved in this approach is the time required to create the 400 data points used for the surface fitting.

**4.3. Algorithm 3: NSGA-II<sub>woS</sub>.** The NSGA-II<sub>woS</sub> approach is basically the minimization of the objective function directly from the FE models. This approach is very time-consuming, or in other words, computationally expensive. To be consistent with the amount of datasets used for database creation only 400 function evaluations of the FE models are allowed for about two hours. Because of this significant amount of time required to produce 400 function evaluations, only one test run was allowed. Over this limit of function evaluations, Table 11 shows that prediction results are highly unsatisfactory, with the total sum of errors for the ten test cases given as 35% and 15% in location and size, respectively. The individual maximum errors in location and size predictions are seen to be 27% and 2.6%, respectively.

**4.4. Algorithm 4: NSGA-II<sub>WS</sub>.** The NSGA-II<sub>WS</sub> approach is essentially the minimization of the objective function via the surrogate models instead of the FE models. This approach is very time saving and increases optimization performance and results. At 400 function evaluations, Table 12 shows the prediction results of the best and mean predicted values over ten runs for each test case. The results are highly satisfactory with the maximum total sum of errors for the ten test cases given as 1% and 2% in location and size, respectively. The individual maximum errors in location and size predictions are 0.9% and 1%, respectively. The mean and best predicted results over the ten runs are also tabulated. It takes 107 s by the surrogate model to get to the lowest minimum objective function value at  $1.95 \times 10^{-8}$ . The standard deviation (std) over the ten runs for each case is found to be reasonable enough, confirming the correlation of results. The total time taken for this approach is calculated as the time taken for building

S/N	Actual $[X, a]$	Best $[X, a]$	Mean $[X, a]$
1	[50, 26]	[49.9291, 25.8786]	[49.6955, 25.5635]
2	[68, 24]	[68.0011, 24.0027]	[68.0426, 24.0560]
3	[50, 20]	[50.1820, 20.0756]	[49.9154, 19.8753]
4	[30, 36]	[30.0322, 36.0383]	[30.2427, 36.0153]
5	[34, 44]	[33.9957, 44.0029]	[34.0788, 43.8958]
6	[58, 40]	[57.9630, 40.0507]	[58.0399, 40.0705]
7	[48, 34]	[47.8901, 34.0011]	[48.1582, 34.1024]
8	[58, 30]	[58.0443, 29.8654]	[57.7149, 29.9079]
9	[48, 28]	[47.9878, 28.0775]	[47.7617, 27.6020]
10	[62, 36]	[61.9884, 36.0114]	[62.0230, 36.0974]
S/N	Best $\min_{\text{Obj}S}$	Best % $([(E-X), (E-a)])$	Std.
1	$1.03 \times 10^{-4}$	[0.1419, 0.4668]	[0.5757, 0.6222]
2	$1.36 \times 10^{-6}$	[0.0016, 0.0111]	[0.2205, 0.1768]
3	$4.41 \times 10^{-4}$	[0.3640, 0.3781]	[0.3117, 0.2228]
4	$3.10 \times 10^{-6}$	[0.1073, 0.1065]	[0.2449, 0.2519]
5	$1.95 \times 10^{-8}$	[0.0126, 0.0066]	[0.4450, 0.3539]
6	$4.46 \times 10^{-6}$	[0.0637, 0.1267]	[0.3116, 0.4229]
7	$7.75 \times 10^{-5}$	[0.2290, 0.0032]	[0.5521, 0.4906]
8	$6.91 \times 10^{-5}$	[0.0764, 0.4485]	[0.8881, 0.3545]
9	$4.85 \times 10^{-5}$	[0.0255, 0.2767]	[0.6261, 0.9934]
10	$6.10 \times 10^{-7}$	[0.0187, 0.0318]	[0.1003, 0.2432]
Total		[1.0406, 1.8559]	

**Table 12.** Percentage errors of ten test cases using NSGA-II<sub>WS</sub> via surrogate models over ten runs.

the database for the surrogate models and the optimization time using surrogates, or

$$\begin{aligned} \text{Total time} &= \text{time to build the database for surrogate modeling} + \text{average optimization time for 10 runs} \\ &= 7135 + 107 = 7242 \text{ s.} \end{aligned}$$

**4.5. Algorithm 5: GBLS<sub>WoS</sub>.** The GBLS<sub>WoS</sub> approach is basically the minimization of the objective function directly from the FE models. This approach is very time-consuming, requiring about three hours to execute an average of 603 function calls over 100 different start points. The function calls cannot be limited to the 400 function evaluations used in the other algorithms and it takes a bit more function calls to obtain the result shown in Table 13. Prediction results are unsatisfactory with the maximum total sum of errors for the ten test cases given as 29% in both location and size. The individual maximum errors in location and size predictions are 7% and 6%, respectively. This approach requires no database creation and takes an average total time of 10016 s.

**4.6. Algorithm 6: GBLS<sub>WS</sub>.** The GBLS<sub>WS</sub> approach is essentially the minimization of the objective function via the surrogate models instead of the FE models. This approach is very time saving and

S/N	Actual $[X, a]$	Predicted $[X, a]$	Minimum <sub>Obj</sub>	% $([(E - X), (E - a)])$
1	[50, 26]	[50.7110, 27.0946]	2.7400	[1.4220, 4.2100]
2	[68, 24]	[67.8292, 24.8455]	2.9079	[0.2512, 3.5229]
3	[50, 20]	[52.4490, 18.7351]	3.0489	[4.8980, 6.3245]
4	[30, 36]	[30.7860, 36.3687]	1.0421	[2.6200, 1.0242]
5	[34, 44]	[34.7297, 45.7929]	4.1068	[2.1462, 4.0748]
6	[58, 40]	[54.8648, 39.4625]	3.9815	[5.4055, 1.3438]
7	[48, 34]	[51.1505, 34.1076]	1.5220	[6.5635, 0.3165]
8	[58, 30]	[59.9530, 31.3064]	4.4239	[3.3672, 4.3547]
9	[48, 28]	[48.2997, 29.0019]	2.4912	[0.6244, 3.5782]
10	[62, 36]	[61.0460, 35.9119]	1.8632	[1.5387, 0.2447]
Total				[28.8367, 28.9942]

**Table 13.** Percentage errors of ten test cases using  $GBLS_{woS}$  over 100 start points.

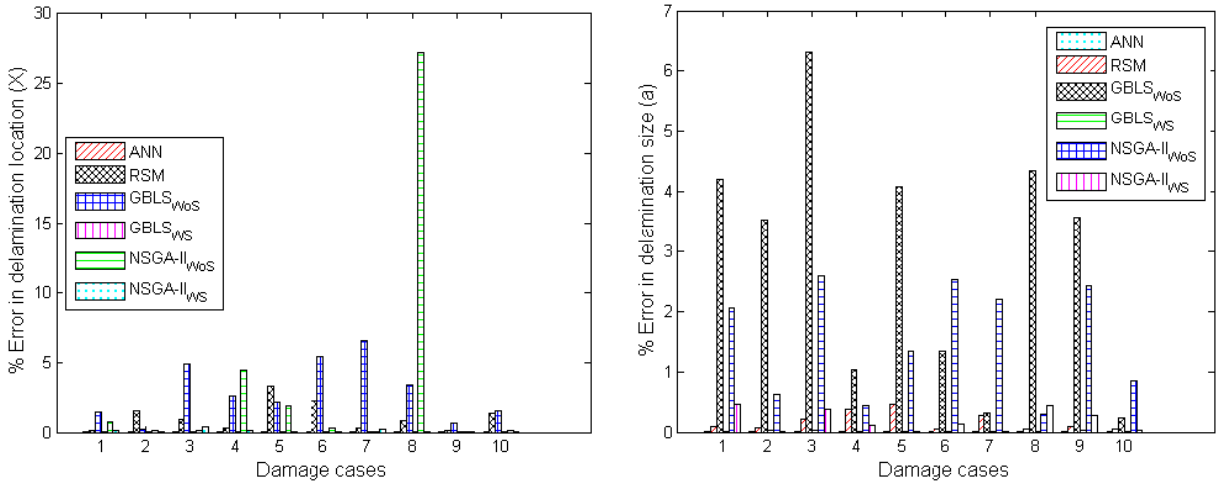
increases optimization performance and results. At an average 524 function calls over ten different start points for the ten test cases, the prediction results using this method are shown in Table 14. The results are also highly satisfactory, with the total sums of errors for the ten test cases being 0.04% and 0.12% in location and size, respectively. The individual maximum errors in location and size predictions are 0.06% and 0.04%, respectively. The excellent results of  $GBLS_{wS}$  can be attributed to a very low objective function, with the lowest at  $3.63 \times 10^{-9}$ . The total time taken for this approach is calculated as the time taken for building the database for the surrogate models and the optimization time using surrogates:

$$\begin{aligned} \text{Total time} &= \text{time to build database for surrogate modeling} + \text{average optimization time for 10 runs} \\ &= 7135 + 124 = 7259 \text{ s.} \end{aligned}$$

**4.7. Summary of comparative results.** Tables 15 and 16 give a comprehensive comparative analysis in terms of completion time and least minimum objective function value and prediction errors, respectively,

S/N	Actual $[X, a]$	Predicted $[X, a]$	# of calls	Minimum <sub>ObjS</sub>	% $([(E - X), (E - a)])$
1	[50, 26]	[49.9983, 26.0013]	657	$4.40 \times 10^{-7}$	[0.0034, 0.0050]
2	[68, 24]	[68.0023, 24.0063]	477	$1.26 \times 10^{-6}$	[0.0034, 0.0262]
3	[50, 20]	[50.0011, 19.9926]	292	$1.08 \times 10^{-5}$	[0.0022, 0.0370]
4	[30, 36]	[30.0017, 36.0010]	530	$4.45 \times 10^{-9}$	[0.0057, 0.0028]
5	[34, 44]	[33.9993, 44.0010]	545	$3.63 \times 10^{-9}$	[0.0021, 0.0023]
6	[58, 40]	[57.9984, 40.0036]	635	$5.50 \times 10^{-8}$	[0.0028, 0.0090]
7	[48, 34]	[48.0014, 34.0022]	661	$5.22 \times 10^{-8}$	[0.0029, 0.0065]
8	[58, 30]	[58.0036, 29.9994]	552	$7.76 \times 10^{-8}$	[0.0062, 0.0020]
9	[48, 28]	[48.0024, 28.0058]	670	$2.83 \times 10^{-7}$	[0.0050, 0.0207]
10	[62, 36]	[61.9991, 36.0038]	523	$4.48 \times 10^{-8}$	[0.0015, 0.0106]
Total					[0.0350, 0.1220]

**Table 14.** Percentage errors of ten test cases using  $GBLS_{wS}$  over ten start points.



**Figure 7.** Performance comparison of different methods based on error of prediction in delamination location  $X$  (left) and size  $a$  (right).

for the different algorithms ANN, RSM, NSGA-II<sub>WoS</sub>, NSGA-II<sub>WS</sub>, GBLS<sub>WoS</sub>, and GBLS<sub>WS</sub>. The prediction percentage errors in delamination location and size for the ten test cases are plotted for better comparison in Figure 7. From Tables 15 and 16 and Figure 7, the following deductions can be made.

- Giving a general conclusion on the algorithm with the best performance in terms of negligible prediction error irrespective of the time, we see from Table 16 that ANN outperforms all other methods, with a maximum prediction error of 0.012% in predicting delamination location and size. This is followed by GBLS<sub>WS</sub>, with a maximum prediction error of 0.06% in terms of delamination location and size, and then NSGA<sub>WS</sub>, with a maximum prediction error of 0.5%. Fourth in the performance order is the highly flexible RSM, with maximum prediction errors of 3.3% and 0.4% in predicting delamination location and size, respectively. The optimization methods without surrogates (GBLS<sub>WoS</sub> and NSGA-II<sub>WoS</sub>) both perform badly, with high amounts of error, which justifies the approach and objective of this study since optimization without surrogates is not only computationally demanding but yields poor results in terms of prediction accuracy because of the huge amount of computational effort required to explore the design space.

S/N	Algorithm	ACT = DT + RT (s)	Minimum objfun
1	ANN	7135 + 133 = 7268	N/A
2	RSM	7135 + 5 = 7140	N/A
3	GBLS <sub>WoS</sub>	10016	4.42
4	GBLS <sub>WS</sub>	7135 + 124 = 7259	$3.63 \times 10^{-9}$
5	NSGA-II <sub>WoS</sub>	7135	2.82
6	NSGA-II <sub>WS</sub>	7135 + 107 = 7242	$1.95 \times 10^{-8}$

**Table 15.** Average completion time (ACT) and lowest minimum objective function values (minimum objfun) for different algorithms, where DT is the time to create database and RT is the run time for the algorithm.

S/N	ANN	RSM	GBLS <sub>W0S</sub>
1	[0.0109, 0.0001]	[0.1448, 0.0850]	[1.4220, 4.2100]
2	[0.0139, 0.0042]	[1.5731, 0.0808]	[0.2512, 3.5229]
3	[0.0136, 0.0003]	[0.9572, 0.2250]	[4.8980, 6.3245]
4	[0.0031, 0.0001]	[0.3057, 0.3786]	[2.6200, 1.0242]
5	[0.0019, 0.0001]	[3.2618, 0.4691]	[2.1462, 4.0748]
6	[0.0006, 0.0011]	[2.2603, 0.0452]	[5.4055, 1.3438]
7	[0.0116, 0.0002]	[0.3396, 0.2771]	[6.5635, 0.3165]
8	[0.0139, 0.0011]	[0.7902, 0.0587]	[3.3672, 4.3547]
9	[0.0043, 0.0004]	[0.1131, 0.0914]	[0.6244, 3.5782]
10	[0.0008, 0.0013]	[1.3452, 0.0589]	[1.5387, 0.2447]
Total	[0.0745, 0.0088]	[11.0909, 1.7698]	[28.8367, 28.9942]
S/N	GBLS <sub>WS</sub>	NSGA-II <sub>W0S</sub>	NSGA-II <sub>WS</sub>
1	[0.0034, 0.0050]	[0.7108, 2.0482]	[0.1419, 0.4668]
2	[0.0034, 0.0262]	[0.1039, 0.6156]	[0.0016, 0.0111]
3	[0.0022, 0.0370]	[0.1268, 2.5863]	[0.3640, 0.3781]
4	[0.0057, 0.0028]	[4.4363, 0.4396]	[0.1073, 0.1065]
5	[0.0021, 0.0023]	[1.8721, 1.3376]	[0.0126, 0.0066]
6	[0.0028, 0.0090]	[0.3376, 2.5282]	[0.0637, 0.1267]
7	[0.0029, 0.0065]	[0.0606, 2.2037]	[0.2290, 0.0032]
8	[0.0062, 0.0020]	[27.1428, 0.2983]	[0.0764, 0.4485]
9	[0.0050, 0.0207]	[0.0610, 2.4284]	[0.0255, 0.2767]
10	[0.0015, 0.0106]	[0.1622, 0.8456]	[0.0187, 0.0318]
Total	[0.0350, 0.1220]	[35.0140, 15.3315]	[1.0406, 1.8559]

**Table 16.** Summary of comparative prediction percentage error results — that is, pairs  $[(E - X), (E - a)]$ — for different proposed algorithms.

- ANN and GBLS<sub>WS</sub> give the best delamination predictions. Results show not more than 0.06% error using both methods for predicting both delamination location and size at a known interface. However, ANN results are achieved by solving only the inverse problem whereas the optimization method requires the solution of the forward problems.
- When ANN and RSM are compared, it is evident that ANN's results beat those of RSM in terms of prediction results; however, the ease and flexibility of the RSM methods when compared to ANN and even other methods offer vital advantages. This is basically because, unlike other methods, RSM gives mathematical expressions (models) that can be used to predict delamination location and size of a modeled structure at any given time.
- GBLS<sub>WS</sub>, in comparison with NSGA-II<sub>WS</sub>, offers better results in terms of accuracy, as is evident in the lowest minimum objective function values for the former and latter,  $3.63 \times 10^{-9}$  and  $1.95 \times 10^{-8}$ , respectively, as shown in Table 15. However, subsequent comparisons between the two approaches when the number of variables considered increases from two to five, as in a more complicated

composite plate, revealed that NSGA-II<sub>WS</sub> performs consistently better than GBLS<sub>WS</sub> due to its ability to accommodate both discrete and continuous variables, which are limited in other methods.

- When the minimum objective function values are compared in Table 15 for the optimization techniques, it is shown that GBLS<sub>WS</sub> gives the lowest minimum value of objective function, followed by NSGA-II<sub>WS</sub>. The objective functions for optimization without surrogates are very high leading to their inefficiency in delamination predictions.
- The number of iterations or function calls for GBLS cannot be controlled, unlike NSGA-II, and hence it yields better results even with function evaluations of less than 300 calls. This justifies the need to use surrogate models.
- It can also be deduced from Table 15 that it takes about the same completion time for all the algorithms to successfully predict delaminations.

## 5. Conclusions

This paper presents potential candidate methods for the detection of delamination damage in composite laminates using variations of natural frequencies before and after damage. Key algorithms (artificial neural networks (ANN), response surface methodology (RSM), and optimization techniques) for real-world implementation of structural health monitoring (SHM) systems for delamination detection in composite structures are presented. The structural health monitoring systems proposed are incorporated with vibration-based damage detection techniques, which has been regarded as an efficient way to assess delamination damage in a structure and foresee probable costly failures. This research should help to identify starting points for damage detection via vibration-based monitoring and also guide practitioners in the field in choosing and implementing the most effective available damage-identification algorithms. The conclusions of our findings can be summarized as thus:

- Different inverse algorithms for delamination damage detection for SHM have been successfully developed and tested with numerical results. The notable excellent delamination prediction results obtained reiterate the robustness and accuracy of the algorithms as well as the approach. The results also underline the advantages of using *K*-means clustering for the choice of database while integrating surrogates in the optimization loop.
- An interesting finding in this research is that related to the superiority of the incorporation of surrogate models in the optimization loop. Surrogates with adequate approximation accuracy can be used to replace computationally expensive analysis. Management of such surrogates, that is, training regimes, selection of training data, and validation schemes, play an important role in any surrogate-assisted optimization exercise. Hence, surrogates enhance the optimization search performance by exploring the entire design space in a relatively short time.
- The use of optimization techniques directly in the optimization loop for delamination detection in composite beams and plates requires evaluations of large numbers of candidate solutions, thereby making the evaluations computationally expensive. The research reported in this thesis is focused on improving the efficiency of delamination detection results by allowing minimum computational cost. The approach adopted was to use surrogate models in lieu of expensive simulations to evaluate the



natural frequencies of the laminates. Further, an optimization strategy has been developed by integration with surrogate models. The optimization methodology was successfully applied to delamination prediction with the objective of improving structural integrity and minimizing computational costs.

- ANN is a superior and more accurate modeling technique as compared with RSM, as it represents nonlinearities in a much better way. However, a major disadvantage of ANN is that the resulting weights of the trained network are difficult to interpret, unlike RSM, which provides physical mathematical models that are easy to compute and interpret. Another problem peculiar to ANN, in contrast to RSM, is the difficulty of finding an appropriate network architecture. The RSM algorithm is straightforward in that it gives a physical mathematical expression for a delamination prediction and its model can easily be validated by statistical means.
- Both RSM and ANN require a large number of numerical experiments for obtaining a trained ANN or regressed response surfaces for predicting the delamination location and size from the measured data. By means of *K*-means clustering, the total number of numerical experiments required to build an ANN and RSM model is reduced to a substantial size. This represents a significant computational cost reduction associated with the developed approach. It can be seen from the previous work by the authors that with small training datasets, ANN and RSM fail to give accurate prediction results. But when solving a forward problem by creating surrogates with small datasets (say of size 40) and doing surrogate-assisted optimization with gradient-based local search (GBLS) and a nondominated sorting genetic algorithm (NSGA-II), the prediction results are very satisfactory.
- When one is interested in solving only the two-variable problem (delamination prediction at a known interface), the RSM and ANN algorithms deliver a good job. However, for three variables (delamination prediction at an unknown interface), surrogate-assisted optimization using GBLS and NSGA-II is very efficient. Finally, for the five-variable problem, as in composite plates, NSGA-II is preferred.

## References

- [Addin et al. 2006] A. O. Addin, S. M. Sapuan, E. Mahdi, and M. Osman, “Review on prediction and detection of failures in laminated composite materials using neural networks”, *Polym. Polym. Compos.* **14**:4 (2006), 433–441.
- [Ayala and dos Santos Coelho 2008] H. V. H. Ayala and L. dos Santos Coelho, “A multiobjective genetic algorithm applied to multivariable control optimization”, pp. 736–745 in *19th COBEM: International Congress of Mechanical Engineering* (Brasilia, 2007), edited by P. E. Miyagi et al., ABCM Symposium Series in Mechatronics **3**, Associação Brasileira de Engenharia e Ciências Mecânicas, Rio de Janeiro, 2008.
- [Bas and Boyaci 2005] D. Bas and I. H. Boyaci, “Modeling and optimization, I: Usability of response surface methodology”, *J. Food Eng.* **78**:3 (2005), 836–845.
- [Broughton et al. 2000] W. R. Broughton, M. J. Lodeiro, G. D. Sims, B. Zeqiri, M. Hodnett, R. A. Smith, and L. D. Jones, “Standardised procedures for ultrasonic inspection of polymer matrix composites”, in *ICCM-12 Europe: 12th International Conference on Composite Materials* (Paris, 1999), edited by M. Thierry and A. Vautrin, ICCM, Tours, 2000. Paper #1030.
- [Buynak et al. 1989] C. F. Buynak, T. J. Moran, and R. W. Martin, “Delamination and crack imaging in graphite-epoxy composites”, *Mater. Eval.* **47** (1989), 438–441.
- [Cantwell and Morton 1992] W. J. Cantwell and J. Morton, “The significance of damage and defects and their detection in composite materials: a review”, *J. Strain Anal. Eng. Des.* **27**:1 (1992), 29–42.
- [Cawley and Adams 1979] P. Cawley and R. D. Adams, “The location of defects in structures from measurements of natural frequencies”, *J. Strain Anal. Eng. Des.* **14**:2 (1979), 49–57.

- [Chakraborty 2005] D. Chakraborty, “Artificial neural network based delamination prediction in laminated composites”, *Mater. Des.* **26**:1 (2005), 1–7.
- [Chen et al. 2004] H.-P. Chen, H. Le, J.-H. Kin, and A. Chattopadhyay, “Delamination detection problems using a combined genetic algorithm and neural network technique”, in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* (Albany, NY, 2004), AIAA, Reston, VA, 2004. AIAA #2004-4397.
- [Dackermann 2010] U. Dackermann, *Vibration-based damage identification methods for civil engineering structures using artificial neural networks*, Ph.D. thesis, University of Technology, Sydney, 2010, Available at <http://hdl.handle.net/2100/1127>.
- [Deb et al. 2002] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II”, *IEEE Trans. Evol. Comput.* **6**:2 (2002), 182–197.
- [Delashmit and Manry 2005] W. H. Delashmit and M. T. Manry, “Recent developments in multilayer perceptron neural networks”, in *Proceedings of the 7th Annual Memphis Area Engineering and Science Conference* (Memphis, TN, 2005), MAESC, Memphis, TN, 2005.
- [Della and Shu 2005] C. N. Della and D. Shu, “Vibration of beams with double delaminations”, *J. Sound Vib.* **282** (2005), 919–935.
- [Doebbling et al. 1998] S. W. Doebbling, C. R. Farrar, M. B. Prime, and D. W. Shevitz, “A review of damage identification methods that examine changes in dynamic properties”, *Shock Vib. Digest* **30**:2 (1998), 91–105.
- [Fang and Tang 2005] X. Fang and J. Tang, “Frequency response based damage detection using principal component analysis”, pp. 407–412 in *Proceedings of the IEEE International Conference on Information Acquisition* (Hong Kong and Macau, 2005), edited by M. Meng et al., IEEE, Piscataway, NJ, 2005.
- [Harrison and Butler 2001] C. Harrison and R. Butler, “Locating delaminations in composite beams using gradient techniques and a genetic algorithm”, *AIAA J.* **39**:7 (2001), 1383–1389.
- [Herath et al. 2010] M. T. Herath, K. Bandyopadhyay, and J. D. Logan, “Modelling of delamination damage in composite beams”, pp. 1189–1198 in *6th Australasian Congress on Applied Mechanics* (Perth, 2010), edited by K. Teh et al., Engineers Australia, Perth, 2010.
- [Ihesiulor et al. 2012a] O. K. Ihesiulor, K. Shankar, Z. Zhang, and T. Ray, “Delamination detection using methods of computational intelligence”, pp. 303–310 in *Proceedings of the Sixth Global Conference on Power Control and Optimization*, edited by N. Barsoum et al., American Institute of Physics Conference Series **1499**, August 2012.
- [Ihesiulor et al. 2012b] O. K. Ihesiulor, K. Shankar, Z. Zhang, and T. Ray, “Effectiveness of artificial neural networks and surrogate-assisted optimization techniques in delamination detection for structural health monitoring”, pp. 198–203 in *Proceedings of the IASTED International Conference on Modelling and Simulation*, edited by S. Al Zahir et al., July 2012. 783-042.
- [Ishak et al. 2001] S. I. Ishak, G. R. Liu, H. M. Shang, and S. P. Lim, “Locating and sizing of delamination in composite laminates using computational and experimental methods”, *Compos. B Eng.* **32**:4 (2001), 287–298.
- [Islam and Craig 1994] A. S. Islam and K. C. Craig, “Damage detection in composite structures using piezoelectric materials (and neural net)”, *Smart Mater. Struct.* **3**:3 (1994), 318–328.
- [Kanungo et al. 2002] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: analysis and implementation”, *IEEE Trans. Pattern Anal. Mach. Intell.* **24**:7 (2002), 881–892.
- [Kessler et al. 2005] S. Kessler, K. Amaratunga, and B. Wardle, “An assessment of durability requirements for aircraft structural health monitoring sensors”, in *Proceedings of the 5th International Workshop on Structural Health Monitoring* (Stanford, CA, 2005), edited by F.-K. Chang, DEStech Publications, Lancaster, PA, 2005.
- [Kim and Yiu 2004] H. L. Kim and W. M. Yiu, “Delamination detection in smart composite beams using Lamb waves”, *Smart Mater. Struct.* **13** (2004), 544–551.
- [MacQueen 1967] J. B. MacQueen, “Some methods for classification and analysis of multivariate observations”, pp. 281–297 in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, I: Statistics* (Berkeley, CA, 1965–1966), edited by L. M. Le Cam and J. Neyman, University of California Press, Berkeley, CA, 1967.
- [Mahdi and El Kadi 2007] E. S. Mahdi and H. El Kadi, “Crushing behavior of laterally compressed composite elliptical tubes: experiments and predictions using artificial neural networks”, *Compos. Struct.* **83**:4 (2007), 399–412.

- [Montgomery 2005] D. C. Montgomery, *Design and analysis of experiments: response surface method and designs*, 6th ed., Wiley, Hoboken, NJ, 2005. 8th ed. published in 2012.
- [Mufti 2001] A. Mufti, *Guidelines for structural health monitoring*, ISIS Canada, Winnipeg, 2001.
- [Mujumdar and Suryanarayan 1988] P. J. Mujumdar and S. Suryanarayan, “Flexural vibrations of beams with delaminations”, *J. Sound Vib.* **125**:3 (1988), 441–461.
- [Nag et al. 2002] A. Nag, D. R. Mahapatra, and S. Gopalakrishnan, “Identification of delamination in composite beams using spectral estimation and a genetic algorithm”, *Smart Mater. Struct.* **11**:6 (2002), 899–908.
- [Okafor et al. 1996] A. C. Okafor, K. Chandrashekhara, and Y. P. Jiang, “Delamination prediction in composite beams with built-in piezoelectric devices using modal analysis and neural network”, *Smart Mater. Struct.* **5**:3 (1996), 338–347.
- [Pardoen 1989] G. C. Pardoen, “Effect of delamination on the natural frequency of composite laminates”, *J. Compos. Mater.* **23**:12 (1989), 1200–1215.
- [Ramanamurthy and Chandrasekaran 2011] E. V. V. Ramanamurthy and K. Chandrasekaran, “Vibration analysis on a composite beam to identify damage and damage severity using finite element method”, *IJEST* **3**:7 (2011), 5865–5888.
- [Ramkumar and Kulkarni 1979] R. Ramkumar and S. Kulkarni, “Free vibration frequencies of a delaminated beam”, in *Reinforcing the future: proceedings of the 34th Annual Conference, Reinforced Plastics/Composites Institute* (New Orleans, LA, 1979), Society of the Plastics Industry, New York, 1979.
- [Rao 2007] M. R. P. D. Rao, “Review of nondestructive evaluation techniques for FRP composite structural components”, Master’s thesis, West Virginia University, Morgantown, WV, 2007, Available at <http://tinyurl.com/Rao-2007-masters-thesis>.
- [Ray et al. 2001] T. Ray, T. Kang, and K. C. Seow, “Multi-objective design optimization by an evolutionary algorithm”, *Eng. Optim.* **33**:4 (2001), 399–424.
- [Sastry et al. 2006] K. Sastry, D. D. Johnson, A. L. Thompson, D. E. Goldberg, T. J. Martinez, J. Leiding, and J. Owens, “Multi-objective genetic algorithms for multiscaling excited state direct dynamics in photochemistry”, pp. 1745–1752 in *GECCO '06: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (Seattle, WA, 2006), edited by M. Cattolico, ACM, New York, 2006.
- [Schittkowski 1986] K. Schittkowski, “NLPQL: a FORTRAN subroutine solving constrained nonlinear programming problems”, *Ann. Oper. Res.* **5**:4 (1986), 485–500.
- [Shang et al. 2001] Y. Shang, Y. Wan, M. P. J. Fromherz, and L. S. Crawford, “Toward adaptive cooperation between global and local solvers for continuous constraint problems”, in *International Conference on Constraint Programming, CP 01: Proceedings of the CoSolv01 Workshop on Cooperative Solvers in Constraint Programming* (Paphos, 2001), 2001.
- [Su et al. 2005] Z. Su, H.-Y. Ling, L.-M. Zhou, K.-T. Lau, and L. Ye, “Efficiency of genetic algorithms and artificial neural networks for evaluating delamination in composite structures using fibre Bragg grating sensors”, *Smart Mater. Struct.* **14**:6 (2005), 1541–1553.
- [Sul et al. 2011] J. H. Sul, B. G. Prusty, and T. Ray, “Prediction of low cycle fatigue life of short fibre composites at elevated temperatures using surrogate modelling”, *Compos. B Eng.* **42**:6 (2011), 1453–1460.
- [Todoroki 2001] A. Todoroki, “The effect of number of electrodes and diagnostic tool for monitoring the delamination of CFRP laminates by changes in electrical resistance”, *Compos. Sci. Technol.* **61**:13 (2001), 1871–1880.
- [Tsoukalas and Uhrig 1997] L. H. Tsoukalas and R. E. Uhrig, *Fuzzy and neural approaches in engineering*, Wiley, New York, 1997.
- [Valoor and Chandrashekhara 2000] M. T. Valoor and K. Chandrashekhara, “A thick composite-beam model for delamination prediction by the use of neural networks”, *Compos. Sci. Technol.* **60**:9 (2000), 1773–1779.
- [Wang and Liu 1982] J. Wang and Y. Liu, “Vibration of split beams”, *J. Sound Vib.* **84**:4 (1982), 491–502.
- [Watkins et al. 2002] S. E. Watkins, G. W. Sanders, F. Akhavan, and K. Chandrashekhara, “Modal analysis using fiber optic sensors and neural networks for prediction of composite beam delamination”, *Smart Mater. Struct.* **11**:4 (2002), 489–495.
- [Zhang et al. 2010] Z. Zhang, K. Shankar, M. Tahtali, and E. V. Morozov, “Vibration modelling of composite laminates with delamination damage”, in *ICA 2010: Proceedings of 20th International Congress on Acoustics* (Sydney, 2010), edited by M. Burgess et al., Australian Acoustical Society, Kensington, 2010.

[Zheng et al. 2011] S.-J. Zheng, Z.-Q. Li, and H.-T. Wang, “A genetic fuzzy radial basis function neural network for structural health monitoring of composite laminated beams”, *Expert Syst. Appl.* **38**:9 (2011), 11837–11842.

Received 1 Nov 2012. Accepted 13 Jan 2013.

OBINNA K. IHESIULOR: [obinna.ihesiulor@student.adfa.edu.au](mailto:obinna.ihesiulor@student.adfa.edu.au)

*School of Engineering and Information Technology, University of New South Wales at the Australian Defense Force Academy, Northcott Drive, Canberra ACT 2600, Australia*

KRISHNA SHANKAR: [k.shankar@adfa.edu.au](mailto:k.shankar@adfa.edu.au)

*School of Engineering and Information Technology, University of New South Wales at the Australian Defense Force Academy, Northcott Drive, Canberra ACT 2600, Australia*

ZHIFANG ZHANG: [zhifang.zhang@student.adfa.edu.au](mailto:zhifang.zhang@student.adfa.edu.au)

*School of Engineering and Information Technology, University of New South Wales at the Australian Defense Force Academy, Northcott Drive, Canberra ACT 2600, Australia*

TAPABRATA RAY: [t.ray@adfa.edu.au](mailto:t.ray@adfa.edu.au)

*School of Engineering and Information Technology, University of New South Wales at the Australian Defense Force Academy, Northcott Drive, Canberra ACT 2600, Australia*

# JOURNAL OF MECHANICS OF MATERIALS AND STRUCTURES

[msp.org/jomms](http://msp.org/jomms)

Founded by Charles R. Steele and Marie-Louise Steele

## EDITORIAL BOARD

ADAIR R. AGUIAR University of São Paulo at São Carlos, Brazil  
KATIA BERTOLDI Harvard University, USA  
DAVIDE BIGONI University of Trento, Italy  
IWONA JASIUK University of Illinois at Urbana-Champaign, USA  
THOMAS J. PENCE Michigan State University, USA  
YASUhide SHINDO Tohoku University, Japan  
DAVID STEIGMANN University of California at Berkeley

## ADVISORY BOARD

J. P. CARTER University of Sydney, Australia  
R. M. CHRISTENSEN Stanford University, USA  
G. M. L. GLADWELL University of Waterloo, Canada  
D. H. HODGES Georgia Institute of Technology, USA  
J. HUTCHINSON Harvard University, USA  
C. HWU National Cheng Kung University, Taiwan  
B. L. KARIHALOO University of Wales, UK  
Y. Y. KIM Seoul National University, Republic of Korea  
Z. MROZ Academy of Science, Poland  
D. PAMPLONA Universidade Católica do Rio de Janeiro, Brazil  
M. B. RUBIN Technion, Haifa, Israel  
A. N. SHUPIKOV Ukrainian Academy of Sciences, Ukraine  
T. TARNAI University Budapest, Hungary  
F. Y. M. WAN University of California, Irvine, USA  
P. WRIGGERS Universität Hannover, Germany  
W. YANG Tsinghua University, China  
F. ZIEGLER Technische Universität Wien, Austria

**PRODUCTION** [production@msp.org](mailto:production@msp.org)

SILVIO LEVY Scientific Editor

Cover photo: Wikimedia Commons

---

See [msp.org/jomms](http://msp.org/jomms) for submission guidelines.


---

JoMMS (ISSN 1559-3959) at Mathematical Sciences Publishers, 798 Evans Hall #6840, c/o University of California, Berkeley, CA 94720-3840, is published in 10 issues a year. The subscription price for 2013 is US\$555/year for the electronic version, and \$705/year (+\$60, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues, and changes of address should be sent to MSP.

---

JoMMS peer-review and production is managed by EditFLOW® from Mathematical Sciences Publishers.

PUBLISHED BY

 **mathematical sciences publishers**  
nonprofit scientific publishing

<http://msp.org/>

© 2013 Mathematical Sciences Publishers

- Efficiencies of algorithms for vibration-based delamination detection: A comparative study**                      **OBINNA K. IHESIULOR, KRISHNA SHANKAR, ZHIFANG ZHANG and TAPABRATA RAY**                      **247**
- Evaluation of the effective elastic moduli of particulate composites based on Maxwell's concept of equivalent inhomogeneity: microstructure-induced anisotropy**                      **VOLODYMYR I. KUSHCH, SOFIA G. MOGILEVSKAYA, HENRYK K. STOLARSKI and STEVEN L. CROUCH**                      **283**
- On successive differentiations of the rotation tensor: An application to nonlinear beam elements**                      **TEODORO MERLINI and MARCO MORANDINI**                      **305**
- Predicting the effective stiffness of cellular and composite materials with self-similar hierarchical microstructures**                      **YI MIN XIE, ZHI HAO ZUO, XIAODONG HUANG and XIAOYING YANG**                      **341**
- On acoustoelasticity and the elastic constants of soft biological tissues**                      **PHAM CHI VINH and JOSE MERODIO**                      **359**
- Identification of multilayered thin-film stress from nonlinear deformation of substrate**                      **KANG FU**                      **369**