POD REDUCED-ORDER MODELING FOR INVERSE MATERIAL CHARACTERIZATION FROM TRANSIENT DYNAMIC TESTS

by

Chenxi Hou

BS in Civil Engineering, Tongji University, China, 2012

Submitted to the Graduate Faculty of
the Swanson School of Engineering in partial fulfillment
of the requirements for the degree of
Master of Science

University of Pittsburgh

2014

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Chenxi Hou

It was defended on

November 21, 2014

and approved by

John C. Brigham, PhD, Assistant Professor, Department of Civil and Environmental Engineering and Bioengineering

Jeen-Shang Lin, ScD, Associate Professor, Department of Civil and Environmental Engineering

Qiang Yu, PhD, Assistant Professor, Department of Civil and Environmental Engineering

Thesis Advisor: John C. Brigham, PhD, Assistant Professor, Department of Civil and Environmental Engineering and Bioengineering

POD REDUCED-ORDER MODELING FOR INVERSE MATERIAL CHARACTERIZATION FROM TRANSIENT DYNAMIC TESTS

Chenxi Hou, M.S.

University of Pittsburgh, 2014

Inverse problem solution methods have been widely used for nondestructive material characterization problems in a variety of fields, including structural engineering, material science, aerospace engineering and medicine. A traditional inverse problem solution approach for material characterization is to create a numerical representation of the system, such as a finite element model, combined with nonlinear optimization techniques to minimize the difference between the experimental response and the numerical representation. Unfortunately, due to the high computational cost of analyzing the numerical representation of many systems, it can often be impractical to solve a given inverse problem by this traditional method.

A strategy for using reduced-order modeling, in particular the proper orthogonal decomposition (POD) model reduction approach in inverse material characterization problems is presented in this work. POD is used to derive a low-dimensional basis from a finite set of full-order numerical analyses of the system. The governing equations of the system are projected onto the obtained POD basis to construct a reduced-order model (ROM). The ROM is then used to replace the full-order modeling to reduce the high computational cost, while still keeping the accuracy of the response close to that of the full-order model. After that, the ROM is combined with a global optimization algorithm to identify an estimation of the material properties in the system. A case study of a damaged aluminum plate, which is subjected to a time-dependent harmonic sinusoidal excitation, is chosen to demonstrate that the ROM strategy is capable of accurately identifying material parameters of a system with minimal computational cost.

TABLE OF CONTENTS

PREFACE x

1.0 INTRODUCTION AND LITERATURE REVIEW 1

1.1 Introduction 1

1.2 Proper Orthogonal Decomposition 2

1.2.1 Principal Component Analysis 4

1.2.2 Karhunen-Loéve Decomposition 5

1.2.3 Singular Value Decomposition 6

1.2.4 Proper Orthogonal Decomposition 7

1.3 Inverse Problem Formulation and Solution Strategy 9

2.0 FORWARD PROBLEM FORMULATION 11

2.1 Time dependent transient dynamic solid mechanics problem 11

2.2 Finite Element Formulation 14

2.3 Time Integration 16

2.3.1 Explicit Time Integration 17

2.3.2 Time Increment 18

2.4 The Method of Snapshot for POD-ROM 19

2.5 Galerkin Projection 21

3.0 INVERSE PROBLEM STRATEGY AND OPTIMIZATION OF NDE 23

3.1 Random Search 23

3.2 Surrogate-Model Method 24

3.3 SMARS Algorithm 24

4.0 CASE STUDY 28

4.1 Thin Aluminum Plate 28

5.0 CONCLUSION AND FUTURE DIRECTIONS 43

BIBLIOGRAPHY 44

LIST OF TABLES

4.1 First 10 largest eigenvalues of all 250 eigenvalues in descending-order. 31

4.2 The POD snapshot projection error of the first 10 snapshots with (a) 3 POD basis members, (b) 5 POD basis members and (c) 10 POD basis members. . . 32

4.3 The errors of POD projection test with randomly generated responses at four time instants by the ROM of 3 POD basis members, 5 POD basis members and 10 POD basis members of (a) test 1 (X = 0.29m, Y = 0.87m) and (b) test 2 (X = 0.46m, Y = 0.64m). 36

4.4 Four random tests included three trial solutions of each with unknown damage region(ϵ^x, ϵ^y) and the corresponding values estimated by the inverse characterization process with 3 POD basis members used for the POD ROM, and the resulting relative L_2-error and $L_\infty-$error of the estimated Young's Modulus distribution with their average values for example-aluminum thin plate . . . 39

4.5 Four random tests included three trial solutions of each with unknown damage region(ϵ^x, ϵ^y) and the corresponding values estimated by the inverse characterization process with 5 POD basis members used for the POD ROM, and the resulting relative L_2-error and $L_\infty-$error of the estimated Young's Modulus distribution with their average values for example-aluminum thin plate . . . 40

4.6 Four random tests included three trial solutions of each with unknown damage region(ϵ^x, ϵ^y) and the corresponding values estimated by the inverse characterization process with 10 POD basis members used for the POD ROM, and the resulting relative L_2-error and $L_\infty-$error of the estimated Young's Modulus distribution with their average values for example-aluminum thin plate . . . 41

4.7 Average one-time computation time of full-order model, ROM of 3 POD basis members, ROM of 5 POD basis members and ROM of 10 POD basis members of 400 times computations for example-aluminum thin plate 42

LIST OF FIGURES

3.1 Flowchart of the SMARS Algorithm. 27

4.1 Schematics of excitation amplitude simulated by sinusoidal force with a frequency of 1000Hz. 29

4.2 Schematics of actuator location and sensor locations. 30

4.3 Comparison of nodal (a) X displacement (b) Y displacement of the third snapshot and the tenth snapshot for POD snapshot projection test. 33

4.4 Schematics of (a) the first mode (b) the second mode (c) the third mode corresponding to the first three largest eigenvalues in descending order. 34

4.5 Comparison of the full-order model displacement field and the low-dimensional approximation with 3, 5 and 10 POD basis members used at time instant 800 μs and damage location (a) x=0.29m y=0.87m (b) x=0.46m y=0.64m for POD projection test with randomly generated responses. 35

4.6 Comparison of nodal X displacement values and Y displacement values of the first POD projection test with randomly generated responses at (a) the 400th time instant and (b) the 800th time instant 37

PREFACE

It took me a long time to finish this work. The writing of this thesis experienced a lot of difficulties and setbacks. But fortunately, many people have helped and encouraged me when I had hard times. Without them, I can nerve achieve my goal and have this success. Firstly, I would like to thank my advisor and chair of my graduate committee, Professor John C. Brigham, with my deepest gratitude, for his guidance, patience, encouragement, and teaching me the attitude to engage in scholarship. Secondly, I would like to thank Professor Jeen-Shang Lin, Professor Qiang Yu for serving on my graduate committee and reviewing my thesis. Their advice helped me improve my thesis and lead me to a more comprehensive and deeper understanding of my work. Also, I would like to thank my colleagues in Professor John C. Brigham's group, Bahram, Hui, Jing, Matthew, Mengyu, Mohammad, Oliver, Pei, Shuang, Weijie and Zhanpeng as well as my best friends Billy, Neil and Wei for their tireless help and sincere encouragement in my study and life. Without them, I could not have accomplished anything. Last, but certainly not least, I would like to thank my family. Their love and support always give me the power to conquer any difficulties and obstacles not only in this thesis, but also in my whole life.

1.0 INTRODUCTION AND LITERATURE REVIEW

1.1 INTRODUCTION

Nondestructive evaluation(NDE), is a wide group of analysis techniques used in science and industry to evaluate the properties of a material, component or system from some type of nondestructive testing without causing damage[1]. After many years of development, NDE methods have been applied on a variety of fields, including structural engineering, mechanical engineering and medicine[1, 2, 3, 4, 5]. For material characterization problems, NDE can be interpreted as an inverse problem. The inverse problem for material characterization is the problem of defining the material behavior of a system through response observations provided by some physical testing method(e.g., nondestructive testing)[6]. The theory of inverse problem creates the theoretical foundation for NDE application on material characterization. A large amount of work has been done for nondestructive inverse material characterization problems and already applied in a variety of fields, regardless of the material is stiff structural material or soft tissues[7, 8, 9, 10, 11, 12].

A traditional approach of inverse material characterization method is to create a numerical representation of the material, like a finite element(FE) model, combined with nonlinear optimization techniques to estimate the properties and minimize the difference between the realistic experimental response and the numerical representation[12]. However, the computational cost of traditional (i.e. full-order) numerical representations is usually relatively high. Combined with the computational requirement of the global optimization method, this traditional inverse problem strategy can be impractical. Therefore, application of another method to reduce the computational expense of traditional numerical representations becomes necessary for the inverse material characterization problem.

Proper orthogonal decomposition(POD), is a method to create a low-dimensional basis that has the best approximation in an L_2 average sense that can be used to create a numerical representation of the system. This basis can be truncated as a low-dimensional basis which still has a good accuracy to represent the high-dimensional experimental or numerical simulation data. Then the low-dimensional basis can be used as the approximate representation of the high-dimensional model and called reduced-order model (ROM) [13]. This technique has been widely implemented in different fields from turbulent fluid flows, optimal control, heat transfer, material characterization to many other fields [12, 13, 14, 15, 16]. After the POD method is implemented, the numerical representation is approximately represented by the POD ROM to reduce the high computational cost which makes the nondestructive inverse material characterization strategy practical and applicable.

1.2 PROPER ORTHOGONAL DECOMPOSITION

POD is a powerful and elegant method for data analysis, which aims at finding a set of optimal low-dimensional data basis description to represent an ensemble of high-dimensional experimental or simulation data[13]. POD was developed by different people for different applications and was known by different names such as principal component analysis (PCA), singular value decomposition (SVD) and Karhunen-Loéve decomposition (KLD). These techniques were also widely used in applications of data analysis, image processing, signal analysis and many engineering fields in resent years[17, 18, 19, 20, 21, 22].

To demonstrate the mathematical interpretation of POD, one can suppose to approximate a vector valued function $\vec{u}(\vec{x}, t)$ over some domain of interest as a finite element sum in the variables-separated form[23, 24]

$$\vec{u}(\vec{x}, t) = \sum_{i=1}^{m} \alpha_i(t)\vec{\phi}_i(\vec{x}), \tag{1.1}$$

where $\alpha(t)$ is the POD coefficients and $\vec{\phi}(\vec{x})$ is the POD basis. The ideal expectation of this approximation is m tends to infinity where the approximation becomes exact, but the best approximation usually cannot be guaranteed. The representation of this equation is not unique. The properties and values of the POD basis are always changing when different methods are utilized to form this POD representation, such as the methods mentioned above. All kinds of POD basis regardless of the properties and values are subjected to orthogonality constraints which is stated as

$$\int_{\Omega} \vec{\phi}_i(\vec{x}) \cdot \vec{\phi}_j(\vec{x}) d\vec{x} = \delta_{ij}, \tag{1.2}$$

where δ_{ij} denotes the Kronecker delta symbol that is defined as

$$\delta_{ij} = \begin{cases} 0 \text{ for } i \neq j \\ 1 \text{ for } i = j \end{cases} \tag{1.3}$$

For a sequence of certain given POD basis, there is a certain choice of the sequence of coefficients. Since the POD basis has orthogonality constraints, the certain sequence of coefficients for a certain sequence of POD basis can be obtained as

$$\alpha_i(t) = \int_{\Omega} \vec{u}(\vec{x}, t) \cdot \vec{\phi}_i(\vec{x}) d\vec{x} \tag{1.4}$$

After the POD basis is obtained, using certain POD basis members to do the approximation follows the rules that each selected POD basis members gives the best approximation in a least square sense for Eqn.1.1, which means, for example, the first five selections of POD basis members give the best five-term approximation and the first ten selections of POD basis members give the best ten-term approximation of the function. Now the different methods of computing the POD basis are explained in the following.

1.2.1 Principal Component Analysis

PCA was developed as a statistical technique and first presented by Pearson(1901) and Hotelling(1933)[25]. Its purpose is to identify the dependence structure with collection of multivariate stochastic data. As a discrete version of POD[13], PCA has the same objective with POD that reducing the dimensionality of the data while maintaining as much information of the data as possible. To achieve this goal, PCA extracts the original variables in the data and transforms them into new uncorrelated variables which are called the principle components. The principle components are arrayed as descending order by the variations content so that the first principle component contains the most variation in the data and the second principle component contains the second most variation and so on. These principle components are used to rebuild the data and the dimensionality of the data is reduced by taking the first several principle components to approximately represent the whole data set, which is similar with selecting the POD basis members which give the best approximation by using current POD basis members. The derivation of PCA is explained in [25]. The basis obtained by this method is corresponding to the eigenvectors obtained by the eigenvalue problem of the covariance matrix Σ which is denoted as

$$\Sigma = E\left\{(U - E(U))(U - E(U))^T\right\} \tag{1.5}$$

where U is the discrete data set. The eigenvalue problem of Σ is described as

$$\Sigma V = \lambda V \tag{1.6}$$

The ith eigenvalue λ_i means the corresponding principle component contains λ_i variation of the data. Finally, the approximation problem is described as

$$U = \sum_{i=1}^{m} \alpha_i \phi_i \tag{1.7}$$

which is the same as the continuous form of POD shown as Eqn.1.1. In this equation, m is the dimension of the covariance matrix Σ, $\alpha_i (i = 1, 2, ...m)$ are the ith principle components, $\phi_i (i = 1, 2, ...m)$ are the eigenvectors of Σ corresponding to the descending-ordered eigenvalues of Σ. The dimensionality is reduced by set m in Eqn.1.7 equal to k where $k < m$.

1.2.2 Karhunen-Loéve Decomposition

KLD is another decomposition approach that was independently developed by Karhuen and expanded by Loéve during 1940s[23, 25]. This theory is developed regarding optimal series expansions of continuous stochastic processes. It can be regarded as the extension of PCA to infinite-dimensional spaces such as the contiguous-time function space. To reduce the dimensionality of the data while maintain a good accuracy of the low-dimensional data, instead of measuring the maximum variance contained in the principle component, KLD measures the minimum mean-square error produced by the approximation. The detail of derivation of discrete KLD is shown in [25]. As the same with PCA, the discrete KLD also solves the eigenvalue problem of the covariance matrix Σ shown in Eqn.1.5. Then the basis is obtained by projecting the eigenvectors onto the data set

$$\phi_i = V_i U \tag{1.8}$$

where $V_i (i = 1, 2, ..., m)$ are the eigenvectors of Σ in the order of corresponding eigenvalues of Σ in descending order. Then the coefficients are obtained by

$$\alpha_i = U \phi_i \tag{1.9}$$

which is a discrete process of Eqn.1.4. If $k(k < m)$ basis members are selected to approximate the data, the mean-square error $\varepsilon^2(k)$ is computed as

$$\varepsilon^2(k) = \sum_{i=k+1}^{m} \lambda_i \tag{1.10}$$

where $\lambda_i (i = k+1, k+2, ..., m)$ mean the ith eigenvalues of Σ in descending order. When the mean-square error remains at a satisfied minimal value after different k for small to large is tried, the low-dimensional approximation can be formed by the first k basis members, which is shown as

$$U = \sum_{i=1}^{k} \alpha_i \phi_i \tag{1.11}$$

1.2.3 Singular Value Decomposition

SVD was established and improved by serval people from 1870s to 1930s[25]. Unlike the other two methods that compute the eigenvalues and eigenvectors to measure the variances and least-square errors to form the low-dimensional approximation of the data, SVD computes the singular values to do the approximation. One advantage of SVD is that it is able to be used for a non-symmetric matrix while the other two can only deal with symmetric matrix. A data set matrix U can be written in the SVD form as[23]

$$U = S\Lambda D^T \tag{1.12}$$

where S is a $m \times m$ orthogonal matrix, D is a $n \times n$ orthogonal matrix and Λ is a $m \times n$ matrix with $r(r = \min(m, n))$ nonnegative values $\sigma_i(i = 1, 2, ..., r)$ on the diagonal and zeros on the other positions. S is called the left singular vector matrix and D is called the right singular vector matrix of matrix U. The values on the diagonal of σ_i is arranged as the descending order, i.e. $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$. Each of the σ is unique and called the singular value of matrix U. If U is real, S and D are always orthogonal. However, SVD does have strong relationship with the other two eigenvalue decomposition methods. If a U^T is multiplied on the left side of U in Eqn.1.12, D can be treated as the eigenvector of matrix U^TU and Λ^2 becomes the eigenvalue of U^TU.

$$\begin{aligned} U^TU &= D\Lambda^T S^T S\Lambda D^T \\ \Longrightarrow U^TUD &= D\Lambda^2 \end{aligned} \tag{1.13}$$

Similarly, if a U^T is multiplied on the right side of U in Eqn.1.12, S becomes the eigenvector of matrix UU^T and Λ^2 becomes the eigenvalue of UU^T.

$$\begin{aligned} UU^T &= S\Lambda D^T D\Lambda^T S^T \\ \Longrightarrow UU^TS &= S\Lambda^2 \end{aligned} \tag{1.14}$$

Therefore, each of the singular value is equal to the corresponding eigenvalue of U. Then treat $S\Lambda$ as Q, Eqn.1.5 can be written as

$$U = QD^T = \sum_{i=1}^{m} q_i d_i^T \tag{1.15}$$

which is the same as Eqn.1.1. Then the low-dimensional approximation is done by replacing S and D with their first $k(k < m)$ columns with use of the first k singular values, which means replace Λ with its first k rows and first k columns. These three methods are actually equal to each other and the detail of demonstration is in [25, 26].

1.2.4 Proper Orthogonal Decomposition

For the POD approach, each solution field over the position domain at certain time instant is called a snapshot. Next, the inner product $(\cdot,\cdot)$ and the L_2-norm $\|\cdot\|_{L_2(\Omega)}$ for the Hilbert space $L_2(\Omega)$ [13] are defined as

$$(\vec{f},\vec{g}) = \int_\Omega \vec{f}\cdot\vec{g}d\Omega \qquad \forall\vec{f},\vec{g}\in L_2(\Omega) \tag{1.16}$$

and

$$\|\vec{y}\|^2_{L_2} = (\vec{y},\vec{y}) \qquad \forall\vec{y}\in L_2(\Omega) \tag{1.17}$$

Then a finite dimensional subspace in the Hilbert space is defined as $V_m = \text{ span}\{\phi_i\}_{i=1}^m$ with a mode basis $\{\phi_i\}_{i=1}^m$. Remember that the approximation is represented as

$$\vec{u}(\vec{x},t) = \sum_{i=1}^m \alpha_i(t)\vec{\phi}_i(\vec{x}) \tag{1.18}$$

For $\vec{u}(\vec{x},t)$ at a exact instant k, $\vec{u}_k(\vec{x})$ is a snapshot, and is represented as

$$\vec{u}_k(\vec{x}) = \sum_{i=1}^m \alpha_{ki}\vec{\phi}_i(\vec{x}) \tag{1.19}$$

The best approximation of a snapshot u^*_k can be defined as

$$\|\vec{u}_k - \vec{u}^*_k\|^2_{L_2} = \text{inf}\|\vec{u}_k - \vec{\nu}\|^2_{L_2} \qquad \forall\nu\in V_m \tag{1.20}$$

For an orthogonal basis, the best approximation u^*_k can be found by projecting that field on to the basis, which can be written as[12]

$$\vec{u}^*_k = \sum_{i=1}^m \frac{(\vec{\phi}_i,\vec{u}_k)}{\|\vec{\phi}_i\|^2}\vec{\phi}_i \tag{1.21}$$

When all the snapshots are considered, the best approximation among all the snapshots can be found by solving an optimization problem which is shown as

$$\min\langle\|\vec{u}_k - \vec{u}_k^*\|_{L_2}^2\rangle \tag{1.22}$$

with respect to

$$\|\vec{\phi}_i\|_{L_2}^2 = 1 \tag{1.23}$$

where $\langle\cdot\rangle$ is the average symbol. Substitute Eqn.1.21 and introduce a Lagrange multiplier λ into Eqn.1.22. It gives

$$\min\left\langle\|\vec{u}_k\|_{L_2}^2 - (\vec{\phi}_i, \vec{u}_k)^2 + \lambda(\|\vec{\phi}_i\|_{L_2}^2 - 1)\right\rangle \tag{1.24}$$

Then the minimization problem can be converted to a maximization problem

$$\max\left[\langle(\vec{\phi}_i, \vec{u}_k)^2\rangle - \lambda(\|\vec{\phi}_i\|_{L_2}^2 - 1)\right] \tag{1.25}$$

After some operations(details are explained in [27]), this maximization problem can be transformed into an infinite dimensional eigenvalue problem which is shown as

$$\int_\Omega \langle\vec{u}_k(\vec{x}) \cdot \vec{u}_k(\vec{\xi})\rangle\vec{\phi}_i(\vec{x})d\vec{x} = \lambda\vec{\phi}_i(\vec{\xi}) \tag{1.26}$$

There are several ways to solve this problem, such as the direct method and the method of snapshot[15], both of which can reduce the computational effort under specific circumstance. When the number of the grid points of the numerical representation is relatively small, the direct method is quite efficient to solve this problem. When the number of snapshots is small, the method of snapshot is more efficient, which is going to be discussed in detail later in this paper.

1.3 INVERSE PROBLEM FORMULATION AND SOLUTION STRATEGY

In structure dynamics, the forward problem usually calculates the response of the structure such as displacement, strain and velocity with known properties, constraint and boundary conditions. On the other hand, the inverse problem here consists of the computation of the properties, geometry and boundary conditions from the response observed. For example, if an acoustic wave is scattered by the obstacle or damage of a material, one can observe the wave scattering far away from the obstacle or damage and obtain the properties, position or severity of the obstacle or damage through analyzing the wave's frequency, degree of attenuation and other properties.

In order to obtain a solution of inverse problem, the problem is presented as an optimization problem. This optimization problem seeks to identify the inverse problem solution parameters that minimize some error function that quantifies the difference between the measured response and the response predicted by the full-order model for a given set of parameter estimates. The error function for this work is defined as[12]

$$J(\vec{\gamma}) = \frac{1}{N^p} \left\| p^{exp}(\vec{x}, t) - p^{sim}(\vec{\gamma}, \vec{x}, t) \right\|_{L_2(\vec{x},t)} \tag{1.27}$$

where $\vec{\gamma}$ means the vector of parameters to be determined. N^p is the total numbers of discrete points of the numerical representation. p^{exp} is the experimental response with all parameters of the defect and the time instance known. p^{sim} denotes the response of the numerical representation with a set of trial parameters $\vec{\gamma}$ and a trial position $\vec{x}$ at a time instant t. $\| \cdot \|_{L_2(\vec{x},t)}$ means the discrete L_2-norm at the measurement point and time. Therefore, $J(\vec{\gamma})$ can be considered as the L_2-error of the inverse problem approximation.

Because of the fact that closed-form solutions are only possible to be found for simple geometries and boundary condition problems, finding analytical solutions for complex system inverse problem needs some other methods for help. Therefore, combination of numerical representation of the complex system as mentioned above and nonlinear optimization methods becomes necessary to solve the inverse problem under this circumstance. However, solving the numerical representations alone is already a computationally process due to the complicated geometries and boundary conditions, which largely compresses the space for computing nonlinear optimization method. In addition, these inverse problems are found to have non-convex surfaces with multiple local solutions and the search of parameters can vary in a large range of the material[12]. Therefore, an optimization algorithm which is able to search a broad range of material parameters with the fewest numerical analysis is needed here .

A lot of optimization algorism have been developed and applied into optimization problems related to inverse problems, such as genetic algorithm (GA), random search method(RS) and artificial neural network surrogate model methods[28, 29, 30, 31, 32, 33]. Generally, gradient based methods and non-gradient based methods are the two main types of optimization methods and both of them have inherent advantages and disadvantages. For the gradient based method, it typically only needs relatively small number of steps to converge to a solution, but is also very likely to converge into a local minimized solution if started far from the global solution. On the other hand, the non-gradient based method requires no gradient information and is more robust to converge to a global minimized solution, but at the same time, needs more steps to converge to a global solution, which makes this method inefficient and cost higher computational expense.

In order to obtain an accurate solution and make the search process efficient and consistent, the search space of the unknown material properties needs to be relatively wide enough to provide enough information. The methods such as GA and RS are used to calculate a global minimum solution. However, almost all of the global search methods need a large amount of numerical analysis to finish the whole optimization process which makes the optimization process computationally expensive and time-consuming. Alternatively, surrogate-model methods can be used instead to solve the optimization problem with very little computational expense[34, 35, 36]. But surrogate model methods also have the difficulty to converge to a global minimum solution. A hybrid optimization algorithm, called the Surrogate-Model Accelerated Random Search(SMARS), which combines both the advantages of global search method and surrogate-model method, is used in this work to reduce the computation cost while obtain a global minimum solution at the same time. The details of this algorithm will be introduced later.

2.0 FORWARD PROBLEM FORMULATION

2.1 TIME DEPENDENT TRANSIENT DYNAMIC SOLID MECHANICS PROBLEM

The forward numerical representation considered in this work is a time dependent transient dynamic problem in solid mechanics and is defined as a boundary value problem (BVP). This solid mechanics system can be represented by a partial differential equation (PDE) which is derived from the standard balance of linear momentum. The body force are not considered here, so the governing equation is given by

$$\rho\vec{\ddot{u}}(\vec{x},t) - \nabla \cdot \sigma(\vec{x},t) = \vec{0} \text{ on } \Omega \times I \tag{2.1}$$

Where ρ is the mass density of the solid, $\vec{x}$ is the spatial position vector, $\sigma(\vec{x},t)$ is the solid stress tensor, $\ddot{u}(\vec{x},t)$ is the acceleration vector, t is the time instant, Ω is the solid domain and I is the time domain. Based on Cauchy's stress principle, the natural boundary condition is given by

$$\sigma(\vec{x},t) \cdot \vec{n}(\vec{x}) = \vec{T}(\vec{x},t) \text{ on } \Gamma_T \times I \tag{2.2}$$

and the essential boundary condition is given by

$$\vec{u}(\vec{x},t) = \vec{g}(\vec{x},t) \text{ on } \Gamma_D \times I \tag{2.3}$$

where $n(\vec{x})$ is the outward vector normal to the solid surface, $T(\vec{x},t)$ is the prescribed boundary traction vector, $u(\vec{x},t)$ is the solid displacement field, $g(\vec{x},t)$ is the prescribed boundary displacement, Γ_T is the portion of solid boundary where external traction is prescribed and Γ_D is the portion of solid boundary where displacement is prescribed.

To solve this time dependent dynamic problem, the initial value of displacement and velocity need to be specified additionally as

$$\vec{u}(\vec{x}, t_0) = \vec{u}_0(\vec{x}) = \vec{0} \text{ on } \Omega \tag{2.4}$$

and

$$\vec{\dot{u}}(\vec{x}, t_0) = \vec{v}_0(\vec{x}) = \vec{0} \text{ on } \Omega \tag{2.5}$$

where t_0 is the initial time instant, $u_0(\vec{x})$ is the initial displacement vector and $v_0(\vec{x})$ is the initial velocity vector. The initial displacement and initial velocity are set to be zero at the initial time step.

Assume the material considered here is a linear elastic body. Then the strains and stresses can be assumed to be linear elastic, which can be written as

$$\sigma(\vec{x}, t) = C^{IV} : \varepsilon(\vec{x}, t) \tag{2.6}$$

where C^{IV} is the fourth order elasticity tensor and $\varepsilon(\vec{x}, t)$ is the strain tensor. In Voigt notation, it can be written as

$$\{\sigma\} = [D]\{\varepsilon\} \tag{2.7}$$

and $[D]$ is called the elasticity matrix. To simplify the process, assume the solid considered here is isotropic and the problem is a two dimensional problem so that this problem can be treated as a 2-D plane stress problem. Then the elasticity matrix $[D]$ becomes a three-by-three matrix and there are only two independent constants. The Young's modulus E and the Poisson ratio ν are needed to determine the elasticity matrix $[D]$ for isotropic linear elastic solid. The elasticity matrix $[D]$ can be written as

$$[D] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{2.8}$$

For further simplification, the Lamé's Parameters are introduced here. In linearly elasticity, the Lamé parameters consist of two parameters, λ, called the Lamé's first parameter, and

μ, called the Lamé's second parameter, which is also referred as the shear modulus written as G. These two parameters are defined as

$$\lambda = \frac{E\nu}{1-\nu^2} \tag{2.9}$$

and

$$\mu = \frac{E}{2(1+\nu)} \tag{2.10}$$

Therefore, the elasticity matrix $[D]$ is simplified as

$$[D] = \begin{bmatrix} 2\mu+\lambda & \lambda & 0 \\ \lambda & 2\mu+\lambda & 0 \\ 0 & 0 & \mu \end{bmatrix} \tag{2.11}$$

Then the Eqn.2.7 can be written as

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix} = \begin{bmatrix} 2\mu+\lambda & \lambda & 0 \\ \lambda & 2\mu+\lambda & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{xy} \end{bmatrix}, \tag{2.12}$$

or

$$\sigma(\vec{x},t) = 2\mu\varepsilon(\vec{x},t) + \lambda Tr(\varepsilon(\vec{x},t))I \tag{2.13}$$

where $Tr()$ is the trace operator and I is the identity tensor. Furthermore, to obtain the relationship between strains and displacement, the small strain definition is introduced and the relationship is represented as

$$\varepsilon(\vec{x},t) = \frac{1}{2}\left[\nabla\vec{u}(\vec{x},t) + (\nabla\vec{u}(\vec{x},t))^T\right] \tag{2.14}$$

Then substitute Eqn.2.14 into Eqn.2.13. It gives

$$\sigma(\vec{x},t) = \mu\left[\nabla\vec{u}(\vec{x},t) + (\nabla\vec{u}(\vec{x},t))^T\right] + \frac{1}{2}\lambda Tr(\left[\nabla\vec{u}(\vec{x},t) + (\nabla\vec{u}(\vec{x},t))^T\right])I \tag{2.15}$$

2.2 FINITE ELEMENT FORMULATION

To solve the problem described in Eqn.2.1, the Galerkin weak-form finite element method is implemented here. With the divergence theorem, essential boundary condition and natural boundary condition applied, the weak form of the transient time dependent dynamic problem of solid mechanic is expressed as

$$\int_{\Omega} \sigma(\vec{x},t) : \nabla\vec{\omega}(\vec{x},t)dV + \int_{\Omega} \rho\vec{\omega}(\vec{x},t) \cdot \ddot{\vec{u}}(\vec{x},t)dV - \int_{\Gamma_T} \vec{\omega}(\vec{x},t) \cdot \vec{T}(\vec{x},t)dS = \vec{0} \tag{2.16}$$

where $\omega(\vec{x},t)$ is an arbitrary weight function which meets all the essential boundary conditions homogeneously of the domain. To obtain the algebraic system of this equation, the domain is meshed into finite elements and the displacement, acceleration and weight functions are approximated within the element field as

$$\vec{u}(\vec{x},t) = [N(\vec{x})]\{u^e(t)\} \tag{2.17}$$

$$\ddot{\vec{u}}(\vec{x},t) = [N(\vec{x})]\{\ddot{u}^e(t)\} \tag{2.18}$$

and

$$\vec{\omega}(\vec{x},t) = [N(\vec{x})]\{\omega^e(t)\} \tag{2.19}$$

where $\{u^e(t)\}$ and $\{\omega^e(t)\}$ are the nodal displacement vector and nodal weight function vector, respectively, and $[N(\vec{x})]$ is the shape function matrix. Then the strain and the gradient of weight function can be written as

$$\varepsilon(\vec{x},t) = [B(\vec{x})]\{u^e(t)\} \tag{2.20}$$

and

$$\nabla\vec{\omega}(\vec{x},t) = [B(\vec{x})]\{\omega^e(t)\} \tag{2.21}$$

where $[B(\vec{x})]$ is the matrix of spatial derivatives of the shape function matrix. After application of these approximations and combined with the assumption that mass density is a constant, the algebraic system of the weak-form functions of the full-order model can be written as

$$\sum_{elements} \{\omega^e(t)\}^T \left[\int_{\Omega^e} [B(\vec{x})]^T [D][B(\vec{x})] dV \{u^e(t)\} + \rho \int_{\Omega^e} [N(\vec{x})]^T [N(\vec{x})] dV \{\ddot{u}^e(t)\} - \int_{\Gamma_T^e} [N(\vec{x})]^T \cdot \vec{T}(\vec{x},t) dS \right] = 0 \tag{2.22}$$

This equation must be true for any arbitrary $\{\omega^e(t)\}^T$ and when treat

$$\sum_{elements} \int_{\Omega^e} [B(\vec{x})]^T [D][B(\vec{x})] dV = [K] \tag{2.23}$$

$$\sum_{elements} \rho \int_{\Omega^e} [N(\vec{x})]^T [N(\vec{x})] dV = [M] \tag{2.24}$$

and

$$\sum_{elements} \int_{\Gamma_T^e} [N(\vec{x})]^T \cdot \vec{T}(\vec{x},t) dS = \{F(t)\} \tag{2.25}$$

Eqn.2.22 can be written as

$$[M]\{\ddot{u}(t)\} + [K]\{u(t)\} = \{F(t)\} \tag{2.26}$$

In this algebraic system, $[M]$ is the global mass matrix. $[K]$ is the global stiffness matrix and $\{F(t)\}$ is the external force vector.

2.3 TIME INTEGRATION

To solve this dynamic problem, a time discretization scheme is added here, which means the equation is evaluated at discrete time instants specified at certain time step. There are two common types of time integration methods that are widely used nowadays, the implicit dynamic analysis and the explicit dynamic analysis. Implicit dynamic analysis, for this work, is the method that computes the material response at current time step based on the material response at the current and previous time steps. In other words, the displacement is directly solved by inverting the effective stiffness matrix, which is the summation of the mass matrix and the stiffness matrix multiplied by some coefficients. Because of this process, this method has the character of unconditional stability so that a bigger time increment can be taken. However, the inversion of the stiffness matrix and the subsequent computation also give the implicit method its biggest problem, high computational cost.

In contrast, instead of solving the displacement directly, the explicit dynamic analysis solves the acceleration first and by using the explicit time integration subsequently, the displacement is obtained indirectly. The advantage of solving acceleration instead of solving displacement directly is that there is no need to invert any matrix because the mass matrix is usually modified to a lumped matrix, namely, a diagonal matrix, which means the inversion of mass matrix is trivial, or, namely, the system of equations becomes uncoupled, leading to a higher computation efficiency. The problem of explicit method is that it is conditionally stable, resulting in the requirement of a much smaller time step to make the computation stable and accurate, which is time consuming and costly on the contrary. Since the information measured by many common nondestructive testing methods occurs at relatively high frequency, and thus requires small time steps anyway, the explicit method was used in the present study.

2.3.1 Explicit Time Integration

The explicit central difference integration is used here for the implementation of explicit time integration. This rule is defined as

$$\dot{u}^{(t+\frac{\Delta t}{2})} = \dot{u}^{(t-\frac{\Delta t}{2})} + \Delta t \ddot{u}^{(t)} \tag{2.27}$$

and

$$u^{(t+\Delta t)} = u^{(t)} + \Delta t \dot{u}^{(t+\frac{\Delta t}{2})} \tag{2.28}$$

where t is the current time instant and Δt is the specific time increment. Then Eqn.2.26 can be written as

$$[M]\{\ddot{u}^t\} + [K]\{u^t\} = \{F^t\} \tag{2.29}$$

With the initial displacement prescribed by Eqn.2.4 at the initial time instant and the known of the external force at every time step, the acceleration at the initial time instant can be obtained from Eqn.2.29 as

$$\{\ddot{u}^0\} = [M]^{-1}(\{F^0\} - [K]\{u^0\}) \tag{2.30}$$

At the initial time instant, the velocity at $t - \Delta t/2$ is treated to be equal to the velocity at time zero, which is known from Eqn.2.5. By substituting the initial velocity and the acceleration at the initial time instant into Eqn.2.27, the velocity at the next time instant can be obtained as

$$\dot{u}^{(0+\frac{\Delta t}{2})} = \dot{u}^0 + \Delta t \ddot{u}^0 \tag{2.31}$$

Then with the velocity at the next time moment applied into Eqn.2.28, the displacement of the next time moment is captured as

$$u^{(0+\triangle t)} = u^0 + \Delta t \dot{u}^{(0+\frac{\Delta t}{2})} \tag{2.32}$$

Again substitute the displacement of the next time instant into Eqn.2.29 and the acceleration of the next time instant is obtained. All the steps above form a circuit that solves the acceleration, velocity and displacement at any time instant.

One thing that needs to be stated here is when solving acceleration by using Eqn.2.29, one should remember to modify the mass matrix to a diagonal matrix. There are many methods can be used to modify a matrix into a diagonal matrix, such as Hinton-Rock-Zienkiewicz method and row sum (or column sum) method. For simplification, in this work, row sum method is applied. It is expressed as

$$M^e_{ii} = \sum_{j=1}^{n} M^e_{ij} \tag{2.33}$$

where the first subscript means the row number of the element, the second subscript means the column number of the element and n means the total column number of the matrix.

2.3.2 Time Increment

For the requirement of conditional stability of the explicit time integration, a critical time increment for each time step was found which all the time steps must be less than to prevent the unstable numerical results. This critical time increment is defined in [37] as

$$\Delta t_i \leq \Delta t_{critical} = \frac{2}{\omega_i} \tag{2.34}$$

where ω_i is obtained from the eigenvalue problem defined as

$$[K]\{x\} = \omega^2[M]\{x\} \tag{2.35}$$

In this equation, $[K]$ and $[M]$ are the global stiffness matrix and global mass matrix, respectively. There is also a easier way to find the critical time step by selecting the largest eigenvalue of the problem which satisfies the condition

$$\omega_{max} \leq \max(\omega^e) \tag{2.36}$$

where

$$[K^e]\{x^e\} = (\omega^e)^2[M^e]\{x^e\} \tag{2.37}$$

where ω^e means element eigenvalue, $[K^e]$ means element stiffness matrix and $[M^e]$ means the element mass matrix. Then Eqn.2.34 can be written as

$$\Delta t \leq \Delta t_{critical} = \frac{2}{\omega_{max}} \tag{2.38}$$

Except this limitation, the upper limit of the time step is also restricted by the stability of wave propagation-based nondestructive testing which is going to be used in the case study later. This restriction is defined as[38]

$$\Delta t = \frac{1}{20 f_{max}} \tag{2.39}$$

where f_{max} is the highest frequency of the waves propagating through the structure. The smaller value between Eqn.2.38 and Eqn.2.39 will be selected as the real critical time increment. Meanwhile, the mesh size of the model is restricted by the wavelength. For a good spatial resolution of the wave propagation, the required length of finite element size is 20 nodes per wavelength, which can be written as[38]

$$L_e = \frac{\lambda_{min}}{20} \tag{2.40}$$

where L_e is the finite element size and λ_{min} is the smallest wavelength of the waves propagating through the structure. There are more details and examples about simulation of guided wave propagating in solid medium in [38, 39, 40].

2.4 THE METHOD OF SNAPSHOT FOR POD-ROM

Since this work is about transient dynamic problem with explicit time integration method utilized, the number of snapshots are much less than the number of total grid nodes of the FE model. Therefore, the method of snapshot is selected for the derivation of POD basis. Recall from Eqn.1.26 and let

$$D_k = \int_\Omega \vec{u}_k(\vec{x}) \cdot \vec{\phi}_i(\vec{x}) d\vec{x} \tag{2.41}$$

Then the eigenvalue problem described in Eqn.1.26 can be written as

$$\frac{1}{N} \sum_{k=1}^{N} \vec{u}_k(\vec{\xi}) D_k = \lambda \vec{\phi}_i(\vec{\xi}) \tag{2.42}$$

where N is the number of total snapshots. By multiplying another term $\vec{u}_l(\vec{\xi})$ on both side with integrating over the domain, Eqn.2.42 can be written as

$$\frac{1}{N}\sum_{k=1}^{N}\int_{\Omega}\vec{u}_l(\vec{\xi})\cdot\vec{u}_k(\vec{\xi})D_k d\vec{\xi} = \lambda\int_{\Omega}\vec{\phi}_i(\vec{\xi})\cdot\vec{u}_l(\vec{\xi})d\vec{\xi} \tag{2.43}$$

and

$$\lambda\int_{\Omega}\vec{\phi}_i(\vec{\xi})\cdot\vec{u}_l(\vec{\xi})d\vec{\xi} = \lambda D_l \tag{2.44}$$

As a consequence, Eqn.2.42 becomes

$$\frac{1}{N}\sum_{k=1}^{N}\left[\int_{\Omega}\vec{u}_l(\vec{\xi})\cdot\vec{u}_k(\vec{\xi})d\vec{\xi}\right]D_k = \lambda D_l \tag{2.45}$$

Then let

$$A_{lk} = \int_{\Omega}\vec{u}_l(\vec{\xi})\cdot\vec{u}_k(\vec{\xi})d\vec{\xi} \tag{2.46}$$

Finally this problem becomes another eigenvalue problem described as

$$\frac{1}{N}\sum_{k=1}^{N}A_{lk}D_k = \lambda D_l \tag{2.47}$$

After this problem is solved, the *ith* POD basis member can be determined as

$$\vec{\phi}_i(\vec{x}) = \frac{1}{\lambda^i N}\sum_{k=1}^{N}\vec{u}_k(\vec{x})D_k^i \tag{2.48}$$

where D_k^i is the kth component of the ith eigenvector of the eigenvalue problem defined by Eqn.2.47 and λ^i is the corresponding eigenvalue. Under most circumstance, λ^i can be verified to be the indicator of the approximation capability of the *ith* mode. To achieve the low-dimensional approximation, the eigenvalues are arrayed in descending-order, and the first M POD basis members whose corresponding eigenvalues have 99% of the sum of the total eigenvalues are chosen to form the ROM. One note is that if the inner product of the POD basis and its transposition is not orthogonal, it is probably due to the linear dependence of the selected snapshots. This phenomenon always happens when the time step is very small so that the wave does not propagate a long enough distance, especially for the explicit time integration strategy, which requires much smaller time step to make computation stable. One way to solve this problem is to select the snapshot based on the time instant with larger time interval.

2.5 GALERKIN PROJECTION

Recall that the inverse problem is solved by an optimization method defined as Eqn.1.27. If a set of random parameters $\vec{\gamma}$ is given, the next step is to calculate the response of the numerical representation p^{sim} by the ROM. Since the POD basis are already known, from the basic idea of POD, as shown in Eqn.1.1, to complete the POD approximation, the coefficient needs to be calculated. But with the response function $u_i(\vec{x}, t)$ being unknown, the coefficient can not be calculated by Eqn.1.4. Therefore, another method named the Galerkin projection is applied here to calculate the coefficient. This method is quite similar with the Galerkin weak-form method which is used to solve the PDE of the forward problem. The main idea of Galerkin projection is to project the residual of the governing equation defined by Eqn.2.1 onto each POD basis member in the ROM, which can be written as

$$\left(\rho\ddot{\vec{u}}(\vec{x}, t) - \nabla \cdot \sigma(\vec{x}, t), \vec{\phi}_j(\vec{x})\right) = \vec{0} \qquad \forall j = 1, 2, ..., n \tag{2.49}$$

where n is the number of truncated POD basis. After applying the divergence theorem and natural boundary condition, the equation above becomes

$$\begin{aligned}\int_\Omega \sigma(\vec{x}, t) : \nabla\vec{\phi}_j(\vec{x}, t)dV + &\int_\Omega \rho\vec{\phi}_j(\vec{x}, t) \cdot \ddot{\vec{u}}(\vec{x}, t)dV \\ &- \int_{\Gamma_T} \vec{\phi}_j(\vec{x}, t) \cdot \vec{T}(\vec{x}, t)dS = \vec{0} \qquad \forall j = 1, 2, ..., n\end{aligned} \tag{2.50}$$

From Eqn.1.1, it is obvious that

$$\ddot{\vec{u}}(\vec{x}, t) = \sum \ddot{\alpha}_i(t)\vec{\phi}_i(\vec{x}) \tag{2.51}$$

Then substitute Eqn.1.1, Eqn.2.15 and Eqn.2.51 into Eqn.2.50. It gives

$$\begin{aligned}\sum \alpha_i(t) \int_\Omega \left\{\mu\left[\nabla\vec{\phi}_i + (\nabla\vec{\phi}_i)^T\right] + \frac{1}{2}\lambda Tr(\left[\nabla\vec{\phi}_i + (\nabla\vec{\phi}_i)^T\right])I\right\} : \nabla\vec{\phi}_j(\vec{x}, t)dV \\ + \sum \ddot{\alpha}_i(t)\rho \int_\Omega \vec{\phi}_i(\vec{x}) \cdot \vec{\phi}_j(\vec{x})dV = \int_{\Gamma_T} \vec{\phi}_j(\vec{x}, t) \cdot \vec{T}(\vec{x}, t)dS \qquad \forall j, i = 1, 2, ..., n\end{aligned} \tag{2.52}$$

Write Eqn.2.59 into matrix form and it becomes

$$\rho \int_\Omega [\phi_i][\phi_j]dV\{\ddot{\alpha}_i(t)\} + \int_\Omega [\sigma^*][\nabla\phi_j]dV\{\alpha_i(t)\} = \int_{\Gamma_T} [\phi_j]\{T\}dS \qquad \forall j, i = 1, 2, ..., n \tag{2.53}$$

where $[\sigma^*]$ is represented as

$$[\sigma^*] = \mu\left([\nabla\phi_i] + [\nabla\phi_i]^T\right) + \frac{1}{2}\lambda Tr\left([\nabla\phi_i] + [\nabla\phi_i]^T\right)[I] \tag{2.54}$$

Then treat

$$\rho\int_\Omega [\phi_i][\phi_j]dV = [A] \tag{2.55}$$

$$\int_\Omega [\sigma^*][\nabla\phi_j]dV = [B] \tag{2.56}$$

$$\int_{\Gamma_T} [\phi_j]\{T\}dS = [F] \tag{2.57}$$

Eqn.2.53 can be represented as

$$[A]\{\ddot{\alpha}_i(t)\} + [B]\{\alpha_i(t)\} = [F] \tag{2.58}$$

After this coupled system of ordinary deferential equations is solved, the coefficient $\alpha_i(t)$ is obtained. To make sure that the number of the POD basis members selected meets the requirement, or, in other words, the POD basis members can give a good approximation of the original data set, one should always remember to check the error of the low-dimensional approximation to the original data field obtained from the full-order model. The error is calculated by the L_2-norm which is defined as

$$Error = \frac{\|\vec{u}_i^{FOM} - \vec{u}_i^{ROM}\|_{L_2}}{\|\vec{u}_i^{FOM}\|_{L_2}} \tag{2.59}$$

If the error is not satisfied, one can use more POD basis members to form the low-dimensional approximation. If a lot of POD basis members are used while the error is still high, one may consider to modify the original snapshots, such as narrow the time interval of each time instant at which to capture more snapshots in time domain, or narrow the interval of selecting each parameter to provide more information for the original snapshots. Another thing needs to be noticed is that when the second method is used, more numerical analysis will be made and the computational cost will be higher. However, when one numerical analysis is done under one parameter set, there are plenty of time instants at which to capture more snapshots while maintaining the linear independence of each snapshots. Therefore, the method that narrowing the time interval is more recommended since it does not increase the burden of full-order model computation.

3.0 INVERSE PROBLEM STRATEGY AND OPTIMIZATION OF NDE

As mentioned above, the variability of structure materials and the nature of non-convex error surfaces require optimization methods to be non-gradient based methods with the ability of searching a large range of parameters and few number of numerical analysis invovled[41]. In view of this situation, the optimization algorithm called Surrogate-Model Accelerated Random Search algorithm (SMARS) is used. It is a combination of random search method and surrogate-model method which gives the SMARS method the merits of both methods and removes the demerits of both side at the same time.

3.1 RANDOM SEARCH

The process of RS can be described as : firstly, randomly generate several parameter vectors and compute the error between experimental solution and numerical representation solution of each parameter vector. Secondly, select the parameter vector which gives the lowest error, and then randomly generate new parameter vectors around that best solution. These two steps are repeated until the error is below the designed error tolerance or the maximum computation number is reached. The selection of the new parameter vectors is based on the normal distribution with the current best set being the center.

To be a non-gradient based method, RS is a good method to obtain a global solution. Additionally, it is also not sensitive to the searching space and the initial parameter sets. However, as discussed before, RS has the disadvantage that too many numerical representation solutions are required for capturing a global solution. The high computational cost of the numerical representation makes the implementation of RS alone nearly impossible. To solve this problem, the second optimization method, surrogate-model method is introduced here.

3.2 SURROGATE-MODEL METHOD

Surrogate-model is the method that creates an approximate model for a system that captures the relationship between inputs and outputs to estimate the performance of the system. The general surrogate-model problem can be described as obtaining a global approximation function which adequately represents the original function by training a given set of parameters and its corresponding solutions, which are obtained through the original function by inputting the given set of parameters. Various surrogate-model methods are developed during the past ten years, such as Radial Basis Functions(RBF), Artificial Neural Networks(ANN), Support Vector Mechies(SVM) and Polynomial Response Surface Model(PRSM).

The advantage of surrogate-model method is it doesn't need many numerical representation solutions to obtain the approximation functions while maintaining an adequate approximation of the original model. However, on the other hand, there can be a huge deviation of the result of surrogate-model if the offered inputs and outputs are not descriptive. A validation of the data used to generate surrogate-model is necessary to make sure the data is comprehensive. Moreover, Surrogate-model can only estimate the solution within the domain of the input and output data, so that it is very likely for surrogate-model to converge to a local solution. Therefore, it is a good combination of surrogate-model with RS to overcome the disadvantages of both with each other's strong points.

3.3 SMARS ALGORITHM

SMARS algorithm uses both RS and surrogate-model method to obtain a global solution efficiently for inverse problem, while maintain a low requirement of numerical analysis. The flows of SMARS algorithm is given as follows[41]:

Step1. Set the tolerance for optimization error *tol* and the maximum iteration number *num*.

Step2. Set the random search domain for random selection of each parameter vector $D(\vec{a}, \vec{b})$. a and b are the upper bounds and lower bounds of the prescribed parameters in each parameter vector, respectively.

Step3. Randomly generate several parameter vectors $P = \{\vec{\alpha}^1, \vec{\alpha}^2, \vec{\alpha}^3, ..., \vec{\alpha}^k\}$.

Step4. Compute the solutions of numerical representation and the errors between numerical representation solutions and the experimental solution $A = \{J(\vec{\alpha}^1), J(\vec{\alpha}^2), J(\vec{\alpha}^3), ..., J(\vec{\alpha}^k)\}$ by Eqn.1.27.

Step5. Find the lowest error and its corresponding parameter vector as $J(\vec{\alpha}^*) = \min(A)$.

Step6. Check whether $J(\vec{\alpha}^*)$ is lower than the error tolerance *tol*. If it is , trial solution $p^{sim}(\vec{\alpha}^*)$ is the final solution for the inverse problem and the whole process stops. If it is not, check whether the maximum iteration is reached. If it is, trial solution $p^{sim}(\vec{\alpha}^*)$ is the final solution for the inverse problem and the whole process stops. If it is not, go to the next step.

Step7. Update the random search domain $D(\vec{a}, \vec{b})$ for surrogate-model to generate several subsets of the current parameter vectors.

Stpe8. Train those subsets and their corresponding solutions by surrogate-model within the new domain to obtain the approximation function p^{sm}.

Step9. Find the parameter vector $\vec{\alpha}^{sm*}$ which gives the lowest error between the surrogate-model solution and experimental solution, which is defined as

$$J^{sm}(\vec{\alpha}^{sm*}) = \min\{J^{sm}(\vec{\alpha}^{sm1}), J^{sm}(\vec{\alpha}^{sm2}), J^{sm}(\vec{\alpha}^{sm3})......J^{sm}(\vec{\alpha}^{smk})\} \tag{3.1}$$

where

$$J^{sm}(\vec{\alpha}^{sm}) = \sum_{i=1}^{N^p} \|p_i^{exp}(\vec{\alpha}^{sm}) - p^{sm}(\vec{\alpha}^{sm})_i\|_{L_2} \tag{3.2}$$

Step10. Check whether $J^{sm}(\vec{\alpha}^{sm*})$ is lower than the error tolerance *tol*. If it is , trial solution $p^{sm}(\vec{\alpha}^{sm*})$ is the final solution for the inverse problem and the whole process stops. If it is not, set $\vec{\alpha}^{sm*}$ as $\vec{\alpha}^{k+1}$ and add it into the original error collection P. Then the error collection becomes $Q = \{\vec{\alpha}^1, \vec{\alpha}^2, \vec{\alpha}^3, ..., \vec{\alpha}^k, \vec{\alpha}^{k+1}\}$.

Step11.Select n parameter vectors from error collection Q and make up a new parameter vector set. $R = \{\vec{\alpha}^1, \vec{\alpha}^2, \vec{\alpha}^3, ..., \vec{\alpha}^n\}$. Then select two parameter vectors β_1 and β_2 that belong to this vector set R

Step12. Generate m new parameter vectors by the side of β_1 and β_2 and $2m$ new parameter vectors are obtained as $O = \{\vec{\alpha}^1, \vec{\alpha}^2, \vec{\alpha}^3, ..., \vec{\alpha}^m, \vec{\alpha}^{m+1} ..., \vec{\alpha}^{2m}\}$.

Step13. Compute the error of each parameter vector in O by Eqn.1.27 and obtain $B = \{J(\vec{\alpha}^1), J(\vec{\alpha}^2), J(\vec{\alpha}^3), ..., J(\vec{\alpha}^m), J(\vec{\alpha}^{m+1}), ..., J(\vec{\alpha}^{2m})\}$. Then add this solution set to the original solution set Q which creates a new trial solution set $W = \{J(\vec{\alpha}^1), J(\vec{\alpha}^2), J(\vec{\alpha}^3), ..., J(\vec{\alpha}^j)\}$, where $j = k + 1 + 2m$

Step14. Find a new $J(\vec{\alpha}^*) = \min(W)$ and go back to **Step6**.

The flow chart of SMARS algorithm can be found in Figure.3.1. It is obvious that in order to solve inverse problems, SMARS algorithm makes a circuit to find the best solution. For this work, the surrogate-model is implemented by the ANN algorithm. In **Step7**, the updating of random search domain $D(\vec{a}, \vec{b})$ is on account of the unknown of representability of the training data from RS and the bias distribution of the numerical representation solutions, which can affect the generalization capabilities of ANN. This new domain after being updated can be named the search window of surrogate-model, which can be selected as scalar multiples of $\vec{\alpha^*}$ in **Step6**.

By updating the search domain, the ANN will receive more concentrated training data since the results of RS are generated mostly around current best solutions. In **Step9**, the minimization problem is done by GA. This method needs many solutions from the numerical representation which makes it costly. Fortunately, the ANN algorithm is relatively inexpensive to compute so that the implementation of GA here is feasible.

Finally, In **Step11**, for this work, two new parameter vectors, β_1 and β_2, are generated. The rule to choose β_1 and β_2 is defined as follows: β_1 is equal to the parameter vector corresponding to the best solution in R. β_2 is equal to the parameter vector corresponding to the farthest solution to the best solution, while still within the top 20% of the solution ranking. The purpose of this rule is to maintain the diversity of RS input parameter vectors and to keep the representability of RS output data at the same time.

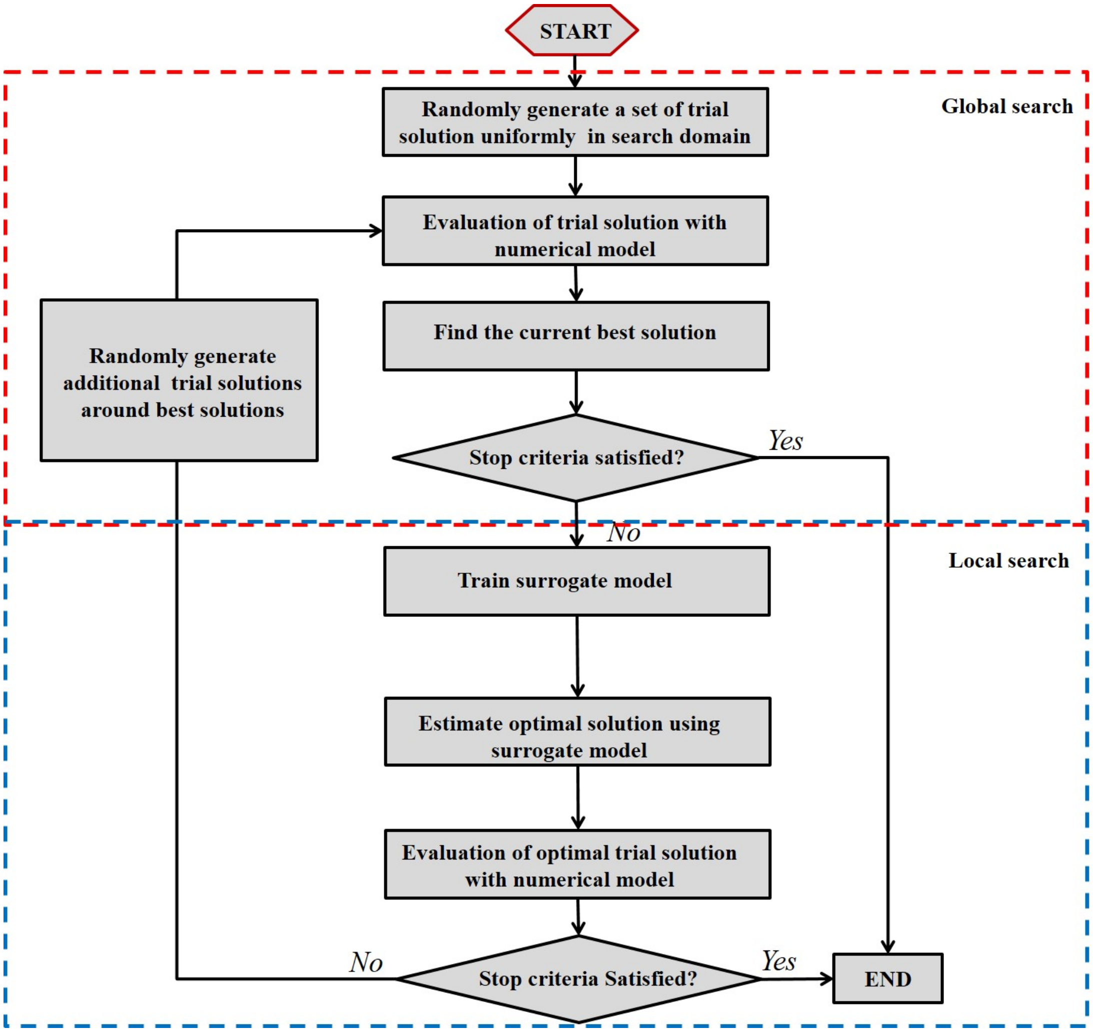

START
Global search
Randomly generate a set of trial solution uniformly in search domain
Evaluation of trial solution with numerical model
Find the current best solution
Randomly generate additional trial solutions around best solutions
Stop criteria satisfied?
Yes
No
Local search
Train surrogate model
Estimate optimal solution using surrogate model
Evaluation of optimal trial solution with numerical model
No
Stop criteria Satisfied?
Yes
END

Figure 3.1: Flowchart of the SMARS Algorithm.

4.0 CASE STUDY

The NDE approach was implemented here by wave propagation-based nondestructive testing. The example was a square aluminum thin plate with excited force in the middle of the plate. Since the thickness of the plate was far less than the other two sides, this plate can be simulated as a 2-D models based on plane stress assumption. Four sensors were mounted uniformly on the plate to receive the signal of the propagating wave in the plate. The damage on the plate was represented by the reduction of Young's Modulus of the material. The X coordinate and Y coordinate of the damage center are considered as the unknown parameters for the damage on the plate. Therefore, these two parameters were selected uniformly over the whole plate and the snapshots were collected with all the combinations of these tow parameters being calculated. The reason why to create the snapshots by uniformly selecting damage parameters is that it is manifested that uniformly selecting damage parameters have a much better capability of inverse identification than randomly selecting damage parameters[42]. Then the ROM was obtained with the POD basis being calculated by the POD strategy, method of snapshots. Finally the SMARS algorism was applied to solve the inverse optimal characterization problem with different trial X and Y coordinates of damage location being applied.

4.1 THIN ALUMINUM PLATE

The size of the square thin aluminum plate considered here was 1m × 1m × 0.02m. The bottom boundary of the plate was assumed to be fixed with the free deformation along other three boundaries. The pitch-catch methods using a piezoelectric transducer(PZT) as

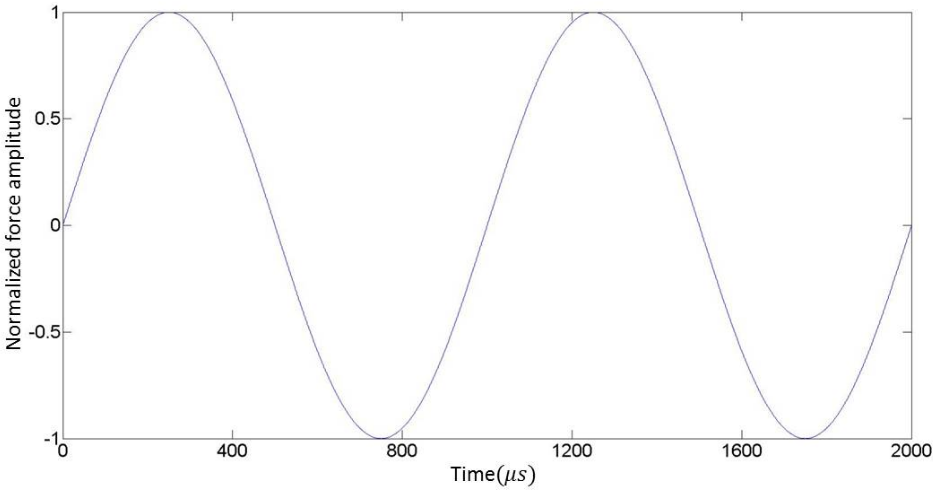


Figure 4.1: Schematics of excitation amplitude simulated by sinusoidal force with a frequency of 1000Hz.

actuator mounted on the middle surface of the plate to create the excitation on the plate with a set of PZTs as sensor to receive the transient signal were modeled here. The PZT actuator was simulated as 12 equal harmonic forces on a square in the middle surface of the plate, with 3 forces in each direction normal to each side of the plate. The size of square with force applied on was 4 mm to remove the possible time shift of the wave signal[40]. The frequency of the force is 1kHz and the normalized amplitude of the force is showed in Figrue.4.1. The location of the actuator was $X_A = 0.48m$, $Y_A = 0.52m$ with respect to the bottom right corner. The four sensors were distributed uniformly with the fixed horizontal orientation to measure the horizontal and vertical displacements at four sensor positions. the X and Y coordinates of each sensors were $X_{s1} = 0.25m$, $Y_{s1} = 0.25m$, $X_{s2} = 0.25m$, $Y_{s2} = 0.75m$, $X_{s3} = 0.75m$,$Y_{s3} = 0.25m$, $X_{s4} = 0.75m$, $Y_{s4} = 0.75m$. Figure.4.2 illustrate the actuator location which is the blue square with arrows showing the force directions and four sensor locations which are the four red crosses. And the sensors were simulated by

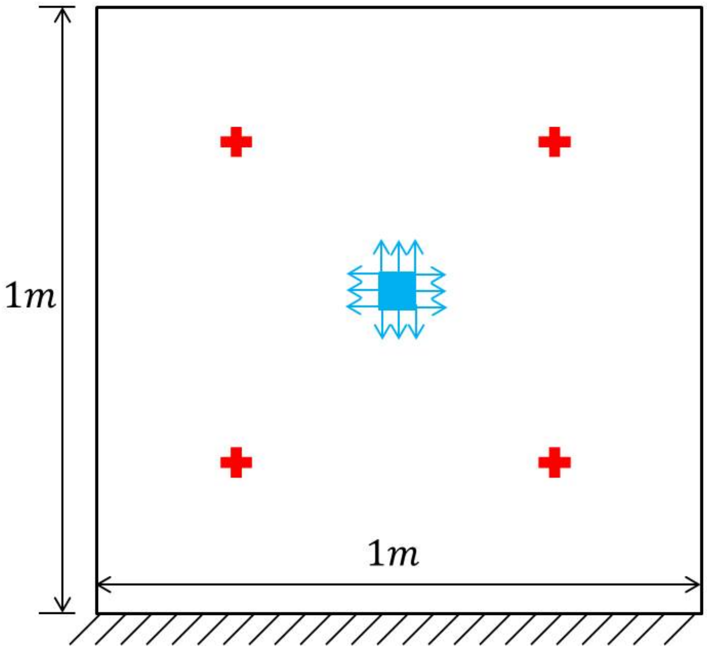


Figure 4.2: Schematics of actuator location and sensor locations.

measuring the displacement of the sensor locations. The attenuation of the ultrasonic wave was not considered in the present work.

As stated before, the damage was represented by a semi-localized reduction of Young's Modulus which was defined as a radial basis function(RBF)[43]

$$E(\vec{x}) = E_h \left[1 - D \cdot \exp\left(-\frac{\|\vec{x} - \vec{\epsilon}\|^2}{c} \right) \right] \tag{4.1}$$

where E_h is the healthy Young's Modulus of the plate, D is the Young's Modulus percent reduction of the RBF, $\vec{\epsilon}$ is the location of the center of the RBF, which contains the X and Y coordinates, c is the breadth of the RBF and $\|\cdot\|$ represents the standard L_2-norm. The Young's Modulus percent reduction and the breadth of the RBF were set to be fixed as 0.55 and 0.016, respectively. The location of the RBF was set to be the unknown parameters and was bounded within $\vec{\epsilon} \in [0,1] \times [0,1]$.

Full-order model can be built and solved as demonstrated in Chapter 2. The unknown parameter $\vec{\epsilon}$, or X and Y coordinates of the damage center were selected 5 evenly spaced

Table 4.1: First 10 largest eigenvalues of all 250 eigenvalues in descending-order.

λ_1	λ_2	λ_3	λ_4	λ_5
1.119×10^{-7}	7.386×10^{-9}	3.695×10^{-10}	2.429×10^{-10}	1.753×10^{-10}
λ_6	λ_7	λ_8	λ_9	λ_{10}
4.546×10^{-11}	2.072×10^{-11}	1.760×10^{-11}	1.373×10^{-11}	1.219×10^{-11}

value through the domain (X = 0.0, 0.25, 0.50, 0.75, 1.00m and Y = 0.0, 0.25, 0.50, 0.75, 1.00m). Every combination of these two values are chosen to be simulated. Therefore, the full-order model was computed 25 times to collect the snapshots. The time step of the full-order model is $1\mu s$, which satisfied the stability requirement. The total simulation time steps were 2000 and the total simulation time for the full-order model was 0.002s, among which the measurements were taken at every ten time instants(200,400,800,...2000μs). Combined with the 25 parameters combinations, there were totally 250 snapshots that were collected to form the ROM. Then the eigenvalues of the matrix A defined as Eqn.2.46 were calculated and truncated by the third, fifth and tenth eigenvalues of all 250 eigenvalues arrayed in descending-order. Tabal.4.1 shows the first 10 largest eigenvalues of all the 250 eigenvalues, among which the first two eigenvalues already had 99% of the sum of all the eigenvalues. Then the first 10 POD basis members were obtained by the methods as shown in Eqn.2.48. To verify that the POD basis and the number of the POD basis members is enough to form a satisfactory approximation, the POD snapshot projection error were checked. To check the POD snapshot projection error, each of the POD basis member was projected onto all the first 10 snapshots denoted as Eqn.1.4 and the coefficients of each snapshots were obtained. Then, as shown in Eqn.1.1, the displacement field snapshots under certain time instants and material parameter set which were the same as the ones used to obtain the first 10 snapshots were gained. Then the L_2-error of each snapshots and its corresponding reconstructed displacement field was calculated and shown as the Table.4.2. The reason why the tenth reconstruction error was higher than the other 9 errors was this snapshot was

Table 4.2: The POD snapshot projection error of the first 10 snapshots with (a) 3 POD basis members, (b) 5 POD basis members and (c) 10 POD basis members.

(a)

	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$	$S9$	$S10$	Ave.
$L_2 - Error(\%)$	9.016	8.306	7.436	4.634	4.764	6.164	5.231	5.425	2.604	20.296	7.388

(b)

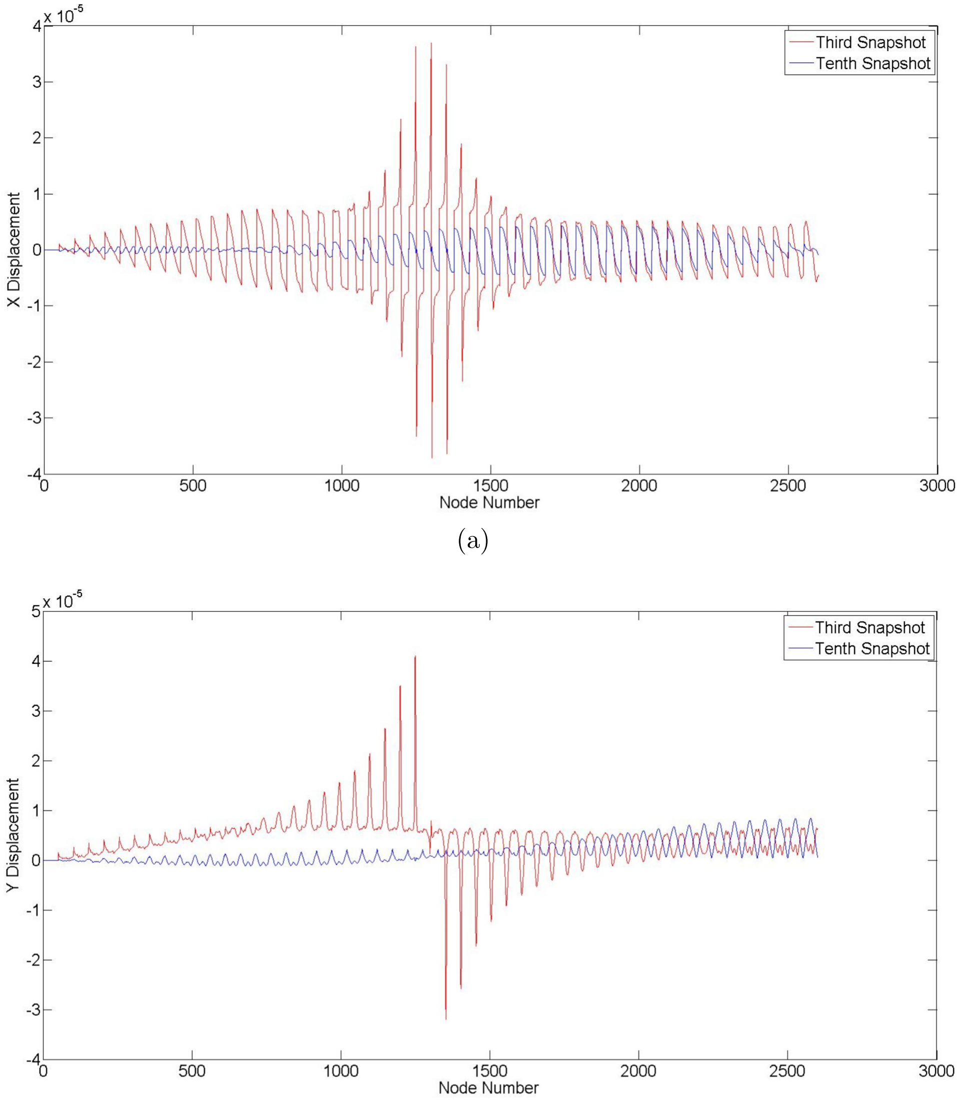
	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$	$S9$	$S10$	Ave.
$L_2 - Error(\%)$	3.090	4.000	7.130	2.801	4.596	3.145	2.794	4.277	2.350	19.324	5.351

(c)

	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$	$S9$	$S10$	Ave.
$L_2 - Error(\%)$	2.341	2.526	4.632	1.954	3.067	2.019	2.093	3.216	1.743	12.296	3.589

taken at the the 2000th time instant, which was at the end of the excitation cycle, leading to the displacement values of the plate at this time instant being small compared to the displacement values at other 9 time instants. The comparison of the displacements at the tenth snapshot and other snapshot can be found in Figure.4.3. Due to the small displacement of the plate at the tenth snapshot, the reconstruction error of the tenth snapshot was less important than the other 9 errors, of which all were relatively low. Therefore, the number of the POD basis members that were used to form the ROM was proper for self-reconstruction. Figure.4.4 shows the magnitude plots of the first three POD basis members.

As mentioned in Section.2.5, the coefficients were then calculated to form the approximation of the displacements at the four sensor locations and 50 time instants uniformly selected over the 2000 μs simulation time. Another verification was implemented to make sure that the coefficients obtained were valid and the number of POD basis members that used was eligible to generate a satisfying low-dimensional approximation of any randomly generated responses. All of the three different circumstances, of which 3, 5 and 10 POD basis members were



(a)

(b)

Figure 4.3: Comparison of nodal (a) X displacement (b) Y displacement of the third snapshot and the tenth snapshot for POD snapshot projection test.

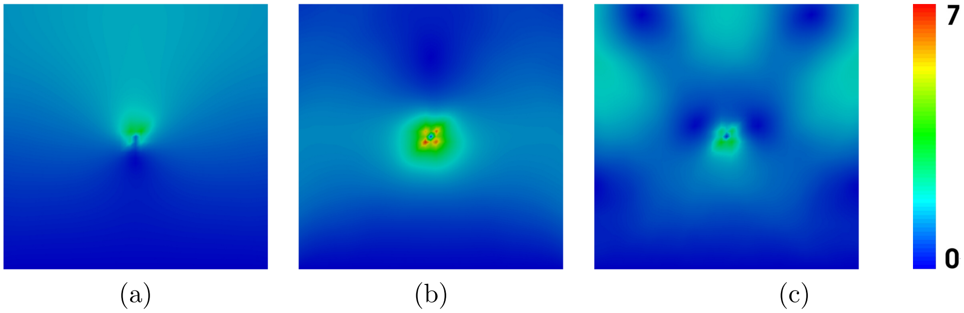
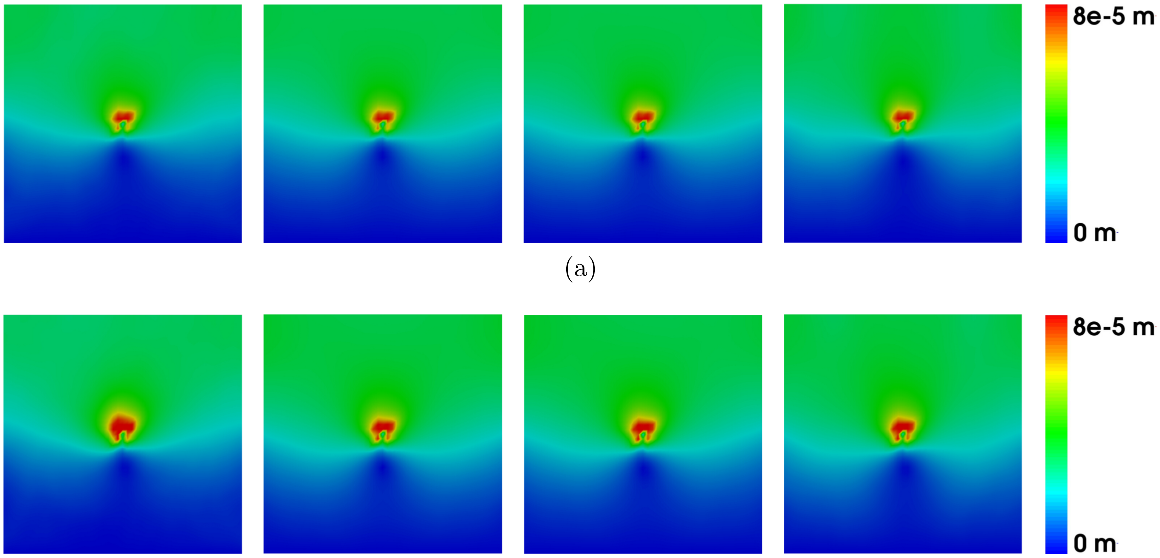


Figure 4.4: Schematics of (a) the first mode (b) the second mode (c) the third mode corresponding to the first three largest eigenvalues in descending order.

selected, were tested by two tests with randomly generated responses at two damage locations X = 0.29m, Y = 0.87m and X = 0.46m, Y = 0.64m. The comparisons of displacement field was plotted as shown in Figure.4.5, the L_2-error was shown in Table.4.3 and the comparisons of the nodal displacement values at two time instants for the second random test was shown in Figure.4.6. As illustrated in Figure.4.5, the displacement field reconstructed from ROM matched the full-order model displacement field response very well. However, from Table.4.3, it is seen that the L_2-errors of each random test at four time instants are relatively high and in some cases, the L_2-errors of using 10 POD basis members are higher than the L_2-errors of using 5 POD basis members. This phenomenon is probably due to the relatively inaccurate reconstruction of the low nodal displacement amplitude part of the whole plate, which is illustrated in Figure.4.6. From this picture it is seen that high amplitude part of the reconstruction, which is much more important, matches the full-order model results very well, while the low amplitude part, which is much less important, does not match each other nicely. Therefore, the POD basis and coefficients could be concluded as eligible.

After the validations, four sets of damage parameter vectors were randomly generated with the damage locations at $X_1 = 0.600$, $Y_1 = 0.900$, $X_2 = 0.127$, $Y_2 = 0.913$, $X_3 = 0.278$, $Y_3 = 0.547$, $X_4 = 0.632$ and $Y_4 = 0.097$. Then the SMARS algorithm was applied to solve the inverse characterization problem by using the ROM consist of 3, 5 and 10 POD basis members, respectively. The random search boundaries of the damage location were within



(a)

(b)

Figure 4.5: Comparison of the full-order model displacement field and the low-dimensional approximation with 3, 5 and 10 POD basis members used at time instant 800 μs and damage location (a) x=0.29m y=0.87m (b) x=0.46m y=0.64m for POD projection test with randomly generated responses.

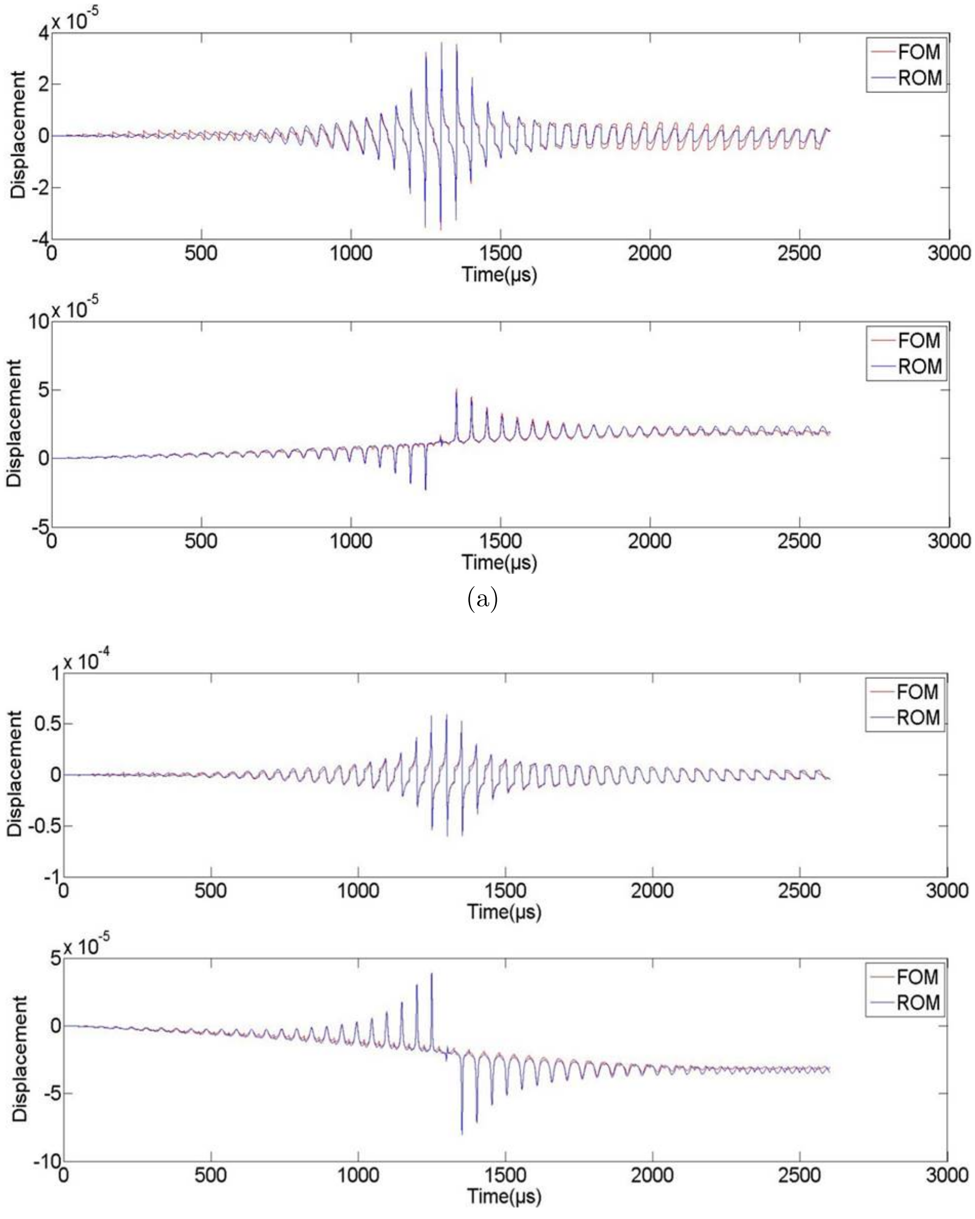
Table 4.3: The errors of POD projection test with randomly generated responses at four time instants by the ROM of 3 POD basis members, 5 POD basis members and 10 POD basis members of (a) test 1 (X = 0.29m, Y = 0.87m) and (b) test 2 (X = 0.46m, Y = 0.64m).

(a)

Time instant(μs)	400	800	1200	1600
3 POD basis members	0.127	0.074	0.086	0.082
5 POD basis members	0.112	0.075	0.078	0.101
10 POD basis members	0.127	0.078	0.066	0.109

(b)

Time instant(μs)	400	800	1200	1600
3 POD basis members	0.176	0.153	0.179	0.282
5 POD basis members	0.166	0.139	0.160	0.265
10 POD basis members	0.204	0.137	0.148	0.268



(a)

(b)

Figure 4.6: Comparison of nodal X displacement values and Y displacement values of the first POD projection test with randomly generated responses at (a) the $400th$ time instant and (b) the $800th$ time instant

$\vec{\epsilon_i} \in [0,1] \times [0,1]m$. The stopping criteria for the SMARS algorithm was set to a maximum of 400 finite element analyses for all 12 models. The results of these 12 cases are shown in Table.4.4, Table.4.5 and Table.4.6.

As seen in the comparisons, in spite of the relatively high error showed in second validation test of the POD basis, this NDT inverse characterization strategy that used 5 and 10 POD basis members for the POD ROM gave a good estimation of the damage locations for all four tests and only the results of test2 and test4 when using 3 POD basis members for the POD ROM had an accurate estimation. The results of using 5 POD basis members for the POD ROM had a conspicuous improvement compared to the results of using 3 POD basis members for POD ROM for two random test1 and test3, while maintained the accuracy of the other two tests. The results of using 10 POD basis members have a small improvement compared to the results of using 5 POD basis members for test1 and test3, while maintained the accuracy of the other two tests as well. Therefore the relatively high error of the second validation test didn't affect the capability of ROM for inverse characterization. Although when using 10 POD basis members for the ROM gave the best result of the damage location, using 5 POD basis members for the ROM already provided a satisfying estimation of the damage location. Therefore, using 5 POD basis members to form the low-dimensional approximation of the random displacement field snapshot is enough. It is not necessary to use more than 5 POD basis members to pursue a better result yet increasing the computational cost at the same time.

The computation time of the full-order model and the POD ROM by using 3, 5 and 10 POD basis members are shown in Table.4.7. As seen of the comparisons, it saved 53.8% of computation time 25.7% of computation time when using 3 POD basis members ROM and 5 POD basis members ROM compared to the full-order model, respectively. However, When the 10 POD basis members ROM was used, the computation time was dramatically increased because of a more complicated system of ordinary differential equations.

Table 4.4: Four random tests included three trial solutions of each with unknown damage region(ϵ^x, ϵ^y) and the corresponding values estimated by the inverse characterization process with 3 POD basis members used for the POD ROM, and the resulting relative L_2-error and $L_\infty-$error of the estimated Young's Modulus distribution with their average values for example-aluminum thin plate

Test		ϵ_1^x	ϵ_1^y	L_2-error	Ave.	$L_\infty-$error	Ave.
1	Target	0.600	0.900	-	-	-	-
	Trial1	0.415	0.901	0.095	-	0.169	-
	Trial2	0.477	0.914	0.072	0.087	0.124	0.153
	Trial3	0.421	0.904	0.094	-	0.165	-
2	Target	0.127	0.913	-	-	-	-
	Trial1	0.118	0.904	0.007	-	0.013	-
	Trial2	0.134	0.918	0.005	0.006	0.009	0.010
	Trial3	0.124	0.906	0.004	-	0.007	-
3	Target	0.278	0.547	-	-	-	-
	Trial1	0.299	0.391	0.085	-	0.112	-
	Trial2	0.163	0.308	0.111	0.090	0.173	0.133
	Trial3	0.238	0.421	0.075	-	0.113	-
4	Target	0.632	0.097	-	-	-	-
	Trial1	0.620	0.095	0.008	-	0.013	-
	Trial2	0.625	0.094	0.005	0.009	0.009	0.015
	Trial3	0.653	0.097	0.014	-	0.023	-

Table 4.5: Four random tests included three trial solutions of each with unknown damage region(ϵ^x, ϵ^y) and the corresponding values estimated by the inverse characterization process with 5 POD basis members used for the POD ROM, and the resulting relative L_2-error and $L_\infty-$error of the estimated Young's Modulus distribution with their average values for example-aluminum thin plate

Test		ϵ_1^x	ϵ_1^y	L_2-error	Ave.	$L_\infty-$error	Ave.
1	Target	0.600	0.900	-	-	-	-
	Trial1	0.568	0.904	0.021	-	0.035	-
	Trial2	0.599	0.908	0.005	0.011	0.007	0.019
	Trial3	0.590	0.907	0.007	-	0.014	-
2	Target	0.127	0.913	-	-	-	-
	Trial1	0.137	0.920	0.007	-	0.012	-
	Trial2	0.118	0.910	0.006	0.007	0.009	0.011
	Trial3	0.116	0.908	0.007	-	0.012	-
3	Target	0.278	0.547	-	-	-	-
	Trial1	0.255	0.536	0.017	-	0.029	-
	Trial2	0.265	0.501	0.030	0.018	0.046	0.029
	Trial3	0.288	0.553	0.008	-	0.013	-
4	Target	0.632	0.097	-	-	-	-
	Trial1	0.652	0.098	0.013	-	0.022	-
	Trial2	0.627	0.101	0.004	0.008	0.007	0.014
	Trial3	0.620	0.093	0.008	-	0.014	-

Table 4.6: Four random tests included three trial solutions of each with unknown damage region(ϵ^x, ϵ^y) and the corresponding values estimated by the inverse characterization process with 10 POD basis members used for the POD ROM, and the resulting relative L_2-error and $L_\infty-$error of the estimated Young's Modulus distribution with their average values for example-aluminum thin plate

Test		ϵ_1^x	ϵ_1^y	L_2-error	Ave.	$L_\infty-$error	Ave.
1	Target	0.600	0.900	-	-	-	-
	Trial1	0.584	0.904	0.011	-	0.018	-
	Trial2	0.603	0.892	0.005	0.007	0.009	0.013
	Trial3	0.592	0.906	0.006	-	0.012	-
2	Target	0.127	0.913	-	-	-	-
	Trial1	0.138	0.888	0.016	-	0.026	-
	Trial2	0.132	0.925	0.007	0.008	0.012	0.013
	Trial3	0.126	0.913	0.001	-	0.001	-
3	Target	0.278	0.547	-	-	-	-
	Trial1	0.280	0.543	0.003	-	0.005	-
	Trial2	0.287	0.524	0.016	0.008	0.025	0.014
	Trial3	0.285	0.554	0.006	-	0.011	-
4	Target	0.632	0.097	-	-	-	-
	Trial1	0.650	0.099	0.012	-	0.020	-
	Trial2	0.633	0.104	0.004	0.008	0.006	0.014
	Trial3	0.618	0.096	0.009	-	0.015	-

Table 4.7: Average one-time computation time of full-order model, ROM of 3 POD basis members, ROM of 5 POD basis members and ROM of 10 POD basis members of 400 times computations for example-aluminum thin plate

	full-order model	3 members ROM	5 members ROM	10 members ROM
Time(s)	12.239	5.659	9.093	22.946

5.0 CONCLUSION AND FUTURE DIRECTIONS

A strategy for using reduced-order modeling, in particular the proper orthogonal decomposition (POD) model reduction approach in inverse material characterization problems is presented in this work. The approach contained an implementation of POD basis for low-dimensional Galerkin finite element modeling of solid mechanics. A case study of an aluminum plate was presented and this POD ROM strategy for inverse material characterization was proved to be capable of identifying material parameters of a system accurately and efficiently with minimal cost.

This strategy showed potential for decreasing the computation cost of finite element modeling while maintaining a good accuracy of material characterization for basic solid material structure based on transient dynamic testing. However, more complicated structures need to be tested in order to show a more significant decreasing of computational cost. In addition, more damage parameters such as depth and reduction of the Young's modulus, as shown in Eqn.4.1, need to be considered and the wave also can be stimulated as a high frequent tone burst wave to make the the simulation closer to realistic experiment.

BIBLIOGRAPHY

[1] C Louis. Nondestructive testing: radiography, ultrasonics, liquid penetrant, magnetic particle, eddy current. *ASM Int*, 1995.

[2] William Rundell and Heinz W Engl. *Inverse problems in medical imaging and nondestructive testing*. Springer-Verlag New York, Inc., 1997.

[3] DM McCann and MC Forde. Review of ndt methods in the assessment of concrete and masonry structures. *NDT & E International*, 34(2):71–84, 2001.

[4] Saleem Akhtar. Review of nondestructive testing methods for condition monitoring of concrete structures. *Journal of Construction Engineering*, 2013, 2013.

[5] Louis Cartz. Nondestructive testing. 1995.

[6] Alexander G Ramm. *Inverse problems: mathematical and analytical techniques with applications to engineering*. Springer, 2005.

[7] Miguel Alejandro Aguilo. *Inverse strategies for characterization of material properties*. 2011.

[8] M Kauer, V Vuskovic, J Dual, G Szekely, and M Bajka. Inverse finite element characterization of soft tissues. *Medical Image Analysis*, 6(3):275 – 287, 2002. Special Issue on Medical Image Computing and Computer-Assisted Intervention - {MICCAI} 2001.

[9] C. Hernandez, A. Maranon, I.A. Ashcroft, and J.P. Casas-Rodriguez. An inverse problem for the characterization of dynamic material model parameters from a single {SHPB} test. *Procedia Engineering*, 10(0):1603 – 1608, 2011. 11th International Conference on the Mechanical Behavior of Materials (ICM11).

[10] ZEA Fellah, Farid G Mitri, Mohamed Fellah, Erick Ogam, and Claude Depollier. Ultrasonic characterization of porous absorbing materials: Inverse problem. *Journal of Sound and Vibration*, 302(4):746–759, 2007.

[11] Aurelio L Araújo, Cristovao M Mota Soares, Carlos A Mota Soares, and Jose Herskovits. Characterisation by inverse techniques of elastic, viscoelastic and piezoelectric properties of anisotropic sandwich adaptive structures. *Applied composite materials*, 17(5):543–556, 2010.

[12] John C Brigham and Wilkins Aquino. Inverse viscoelastic material characterization using pod reduced-order modeling in acoustic–structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 198(9):893–903, 2009.

[13] Wilkins Aquino. An object-oriented framework for reduced-order models using proper orthogonal decomposition (pod). *Computer Methods in Applied Mechanics and Engineering*, 196(4144):4375 – 4390, 2007.

[14] Hung V Ly and Hien T Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and computer modelling*, 33(1):223–236, 2001.

[15] Troy R Smith, Jeff Moehlis, and Philip Holmes. Low-dimensional modelling of turbulence using the proper orthogonal decomposition: a tutorial. *Nonlinear Dynamics*, 41(1-3):275–307, 2005.

[16] Z Ostrowski, RA Białecki, and AJ Kassab. Solving inverse heat conduction problems using trained pod-rbf network inverse method. *Inverse Problems in Science and Engineering*, 16(1):39–54, 2008.

[17] Michael D Graham, Ioannis G Kevrekidis, JL Hudson, Götz Veser, Katharina Krischer, and Ronald Imbihl. Dynamics of concentration patterns of the no+ co reaction on pt: Analysis with the karhunen-loève decomposition. *Chaos, Solitons & Fractals*, 5(10):1817–1831, 1995.

[18] Nejib Smaoui and Ridha B Gharbi. Using karhunen–loéve decomposition and artificial neural network to model miscible fluid displacement in porous media. *Applied Mathematical Modelling*, 24(8):657–675, 2000.

[19] Charlene A Drumm and Michael D Morris. Microscopic raman line-imaging with principal component analysis. *Applied spectroscopy*, 49(9):1331–1337, 1995.

[20] H Berthiaux, V Mosorov, L Tomczak, C Gatumel, and JF Demeyre. Principal component analysis for characterising homogeneity in powder mixing using image processing techniques. *Chemical Engineering and Processing: Process Intensification*, 45(5):397–403, 2006.

[21] Changwoo Ha, Gwanggil Jeon, and Jechang Jeong. Contrast enhancement and noise elimination using singular value decomposition for stereo imaging. *Optical Engineering*, 51(9):090504–1, 2012.

[22] Anna Burvall, Harrison H Barrett, Christopher Dainty, and Kyle J Myers. Singular-value decomposition for through-focus imaging systems. *JOSA A*, 23(10):2440–2448, 2006.

[23] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, 78(7):808–817, 2000.

[24] L Cordier, M Bergmann, et al. Proper orthogonal decomposition: an overview. *Lecture series 2002-04, 2003-03 and 2008-01 on post-processing of experimental and numerical data, Von Karman Institute for Fluid Dynamics, 2008.*, 2008.

[25] YC Liang, HP Lee, SP Lim, WZ Lin, KH Lee, and CG Wu. Proper orthogonal decomposition and its applicationspart i: Theory. *Journal of Sound and Vibration*, 252(3):527–544, 2002.

[26] CG Wu, YC Liang, WZ Lin, HP Lee, and SP Lim. A note on equivalence of proper orthogonal decomposition methods. *Journal of Sound and Vibration*, 265(5):1103–1110, 2003.

[27] Philip Holmes, John L Lumley, and Gal Berkooz. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press, 1998.

[28] Mansoure Montahaei and Behrooz Oskooi. Magnetotelluric inversion for azimuthally anisotropic resistivities employing artificial neural networks. *Acta Geophysica*, 62(1):12–43, 2014.

[29] T HACIB, MR MEKIDECHE, and N FERKHA. Computational investigation on the use of fem and mlp neural network in the inverse problem of defects identifications. *IU-Journal of Electrical & Electronics Engineering*, 8(1):537–548, 2012.

[30] MH Moradi and M Abedini. A combination of genetic algorithm and particle swarm optimization for optimal dg location and sizing in distribution systems. *International Journal of Electrical Power & Energy Systems*, 34(1):66–74, 2012.

[31] T Hatami, MAA Meireles, and G Zahedi. Mathematical modeling and genetic algorithm optimization of clove oil extraction with supercritical carbon dioxide. *The Journal of Supercritical Fluids*, 51(3):331–338, 2010.

[32] Laurent Magnier and Fariborz Haghighat. Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network. *Building and Environment*, 45(3):739–746, 2010.

[33] Che Lin, Venugopal V Veeravalli, and Sean P Meyn. A random search framework for convergence analysis of distributed beamforming with feedback. *Information Theory, IEEE Transactions on*, 56(12):6133–6141, 2010.

[34] Lei Shi, RJ Yang, and Ping Zhu. A method for selecting surrogate models in crashworthiness optimization. *Structural and Multidisciplinary Optimization*, 46(2):159–170, 2012.

[35] Juliane Müller and Robert Piché. Mixture surrogate models based on dempster-shafer theory for global optimization problems. *Journal of Global Optimization*, 51(1):79–104, 2011.

[36] Peng Zhao, Huamin Zhou, Yang Li, and Dequn Li. Process parameters optimization of injection molding using a fast strip analysis as a surrogate model. *The International Journal of Advanced Manufacturing Technology*, 49(9-12):949–959, 2010.

[37] Thomas J. R. Hughes. *The Finite Element Method*. Prentice-Hall, 1987.

[38] Friedrich Moser, Laurence J Jacobs, and Jianmin Qu. Modeling elastic wave propagation in waveguides with the finite element method. *Ndt & E International*, 32(4):225–234, 1999.

[39] Victor Giurgiutiu, Matthieu Gresil, Bin Lin, Adrian Cuc, Yanfeng Shen, and Catalin Roman. Predictive modeling of piezoelectric wafer active sensors interaction with high-frequency structural waves and vibration. *Acta Mechanica*, 223(8):1681–1691, 2012.

[40] M Gresil, V Giurgiutiu, Y Shen, and B Poddar. Guidelines for using the finite element method for modeling guided lamb wave propagation in shm processes. In *6th European Workshop on Structural Health Monitoring*, pages 3–6, 2012.

[41] John C Brigham and Wilkins Aquino. Surrogate-model accelerated random search algorithm for global optimization with applications to inverse material identification. *Computer Methods in Applied Mechanics and Engineering*, 196(45):4561–4576, 2007.

[42] Michele L Joyner. A numerical study of the pod method in nde. *Applied mathematics and computation*, 174(1):732–754, 2006.

[43] Bahram Notghi and John C Brigham. Optimal nondestructive test design for maximum sensitivity and minimal redundancy for applications in material characterization. *Smart Materials and Structures*, 22(12):125036, 2013.