PHYSICS ENABLED DATA-DRIVEN STRUCTURAL ANALYSIS FOR MECHANICAL COMPONENTS AND ASSEMBLIES

A Dissertation
Presented to
The Academic Faculty

By

Aarohi Shah

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology

August 2022

PHYSICS ENABLED DATA-DRIVEN STRUCTURAL ANALYSIS FOR MECHANICAL COMPONENTS AND ASSEMBLIES

Approved by:

Prof. Julian J. Rimoli
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Claudio V. Di Leo
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Graeme J. Kennedy
School of Aerospace Engineering
Georgia Institute of Technology

Prof. George Kardomateas
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Michael R. Tupek
Computational Mechanics
PTC

Date Approved: May 16, 2022

There’s a way to do it - find it.

– Thomas Edison

To my mother

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Prof. Julian J. Rimoli and the Aerospace engineering department at Georgia Tech. for providing me the opportunity to research and earn my doctoral degree. I am grateful to Prof. Julian J. Rimoli for his unwavering support throughout my doctoral journey. I am also thankful to the United States Army and NASA for funding my research through the course of my graduate school.

I am also very grateful to my committee members, Profs. George Kardomateas, Claudio V. Di Leo, Graeme J. Kennedy, and Dr. Michael R. Tupek for their guidance and valuable feedback to my dissertation.

I would like to express my gratitude to Profs. Dewey Hodges, J.V.R. Prasad, Massimo Ruzzene, Marilyn Smith, and Karthik Ramachandran for the enjoyable interactions and their mentorship.

I would like to thank my fellow group members and colleagues at Georgia Tech., Kip, Julie, Amir, Daniel, Chams, Katelyn, Yuan, Darshan, Saurabh, Prathik, Ameya, and my very good friend Hernan. Many thanks should go to all the friends I made at Georgia Tech., Arman, Donald, Luca, Harshit, Ruthvik, Korak, and Hanif without whom my graduate life and experience would have been extremely challenging. Special recognition to my dearest friends, Chetan, Ankit, and Shrunga. I will never forget all the time I have spent with every one of you and your unconditional support throughout this journey.

My graduate school life was greatly supported by my aunt and her family in Marietta, who did not let me feel homesick and were always present through the ups and downs of this journey. I am extremely grateful for their support.

I would like to express my deepest gratitude to my parents and sister, who has always been my foundation. They are the strong pillars who have continuously provided me with their support and love. They have always motivated me to work hard and strive for excellence. A special thanks to my mother- to whom this thesis is dedicated, for her unconditional support and commitment towards my entire life.

Finally, I would like to immensely thank my husband, Mohit Gupta, who is a friend, a colleague, and a true life partner. He has been on my side through the toughest times of this journey and has believed in me even when I doubted myself. His support and love have enabled me to get through it all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Analyzing structures that exhibit nonlinear and history-dependent behaviors is crucial for many engineering applications such as structural health monitoring, wave management/isolation, and geometric optimization to name a few. However, current approaches for modeling such structural components and assemblies rely on detailed finite element formulations of each component. While finite element method serves to be versatile and well-established for nonlinear and history-dependent problems, it tends to be inefficient. Consequently, their computational cost, becomes prohibitive for many applications when time-sensitive predictions are needed.

In the present work, we introduce a framework to develop data-driven dimensionally-reduced surrogate models at the component level, which we call smart parts (SPs), to establish a direct relationship between the input–output parameters of the component. Our method utilizes advanced machine learning techniques to develop SPs such that all the information pertaining to history and nonlinearities is preserved. Unlike other data-driven approaches, our method is not limited to any particular type of nonlinearity and it does not impose restrictions on the type of analysis to be performed. This renders its application straightforward for a diverse set of engineering problems, as we show through multiple case studies. We also propose a novel meta learning based approach to enable an extension of this approach to dynamic problems. In addition, we present several ways to enhance this approach in terms of precision and efficiency. Thus, the present work provides an approach that can dramatically boost the computational efficiency and simplicity to analyze large structures without sacrificing accuracy.

CHAPTER I

INTRODUCTION

1.1 Background and Motivation

The finite element method (FEM) is generally the method of choice when analyzing the mechanical response of structural components and mechanical assemblies. In its standard implementation, the method produces a full field solution, i.e., detailed displacement, strain, and stress fields within each component of an assembly. However, in many scenarios, this level of resolution is not required. For example, when dealing with mechanical assemblies, it is sometimes more important to establish the effective response of its components for the determination of global loads than details pertaining to stress concentrations within each of them. This renders the FEM computationally inefficient in such scenarios: a fine discretization of each part in the assembly leads to a large number of degrees of freedom (DOFs) for the entire model. This problem can become even worse when parts in the assembly have small features that need to be resolved by the mesh for FEM to work.

Extensive research efforts to overcome the challenges posed by FEM have been established [1] with a focus on reducing computational complexity. They usually leverage the fact that when designing a mechanical assembly, detailed meshes are many times not required and force-displacement relations may be sufficient to describe the interactions between components. An early and one of the most pioneering efforts in this direction is the static condensation method [2], in which the unloaded DOFs are expressed as a linear transformation of the loaded DOFs. However, this method is powerful only for linear problems wherein the force-displacement relations are fully captured by the stiffness of the component. Another alternative approach is the reduced basis (RB) method [3, 4, 5]. The main idea behind such a technique is the identification of a suitable problem-dependent basis which can represent parameterized solutions to partial differential equations. However, such a formulation is also geared towards linear applications. Further, in this domain, other related methods such as variational asymptotic beam theory [6], global-local approach [7] and, methods based on partition of unity [8] have also been developed to reduce the burden of complex geometric discretization. However, variational approach is limited to beam-like and plate/shell structures and cannot account for plastic deformations in the material. Moreover, the rest of the methods are based on modifying the solution space by making assumptions on it. For example in the global-local approach, the whole structural component or assembly is solved as a whole and local details are analyzed subsequently based on the results from the global analysis. This transition from global to local behavior or vice-versa entails specific formulations for different problems. Hence, such approaches suffer due to loss of generality. Also, when problems are nonlinear, there are not many established techniques for model order reduction, and those that exist generally cannot handle history-dependent behavior or the computational expense to do so becomes prohibitive. Along these lines, some efforts have been made to extend the RB method for nonlinear approximations [9, 10, 11, 12, 13] but with some limitations that generally compromise the computational efficiency, scope of application or numerical stability. For example, research has been established that focuses on careful selection of the reduced-basis functions for problems with geometric nonlinearity entailed by large deflection problems [9].

For model order reduction in dynamic problems [14] also, mathematical homogenization of the component has been established [15, 16, 17, 18]. Condensation techniques analogous to static condensation have been developed to include inertia effects in the formulation [19, 20, 21, 22]. Such a technique tends to smear the microscopic properties of the problem and provide only an averaged response to be passed at a global level. This (in addition to being limited to linear applications) leads to losses in the accuracy of dynamic simulations, in particular when the wavelength of deformation approaches the size of the heterogeneities. While the accuracy can be recovered to some extent using the improved reduced system [23, 24], enriched continuum methods [25, 26, 27], and advanced multiscale methods [28, 29, 30], the additional computational cost is inevitable. Alternatively, such enhancements entail compromised generality in terms of the type of unit cell and/or the underlying material model. Another prevalent alternative strategy for dynamic problems is the use of spectral methods [31, 32]. Such methods convert the problem from time to frequency domain thereby overcoming the problem of computational inefficiency. However, frequency-domain methods rely on superposing the effect of different frequencies and hence can be only applied to linear problems. Many advancements have been developed to apply spectral methods in nonlinear structural analysis [33, 34, 35, 36], by either solving the eigenproblem or updating the reduced basis at every time step. However, the effectiveness of such an approach for real-life applications is questionable due to the excessive computational time spent at every time step. Along similar lines, another approach has been developed, such that the stiffness remains constant, and the nonlinearity is included in the problem using nonlinear internal forces [37, 38, 39]. Though effective, such an approach proves to be accurate in problems that are only slightly nonlinear. The Bloch-Floquet theory [40] is also extensively used to provide a steady-state solution in any periodic heterogeneous problem by analyzing a single unit cell. But, such a method also can be used only for infinite periodic structures and cannot provide transient solutions for finite arbitrary structures.

Recent advances have explored data-driven (machine learning based) approaches to solve the challenges posed by traditional FEA. With the increase in the amount of data available as well as powerful GPUs, machine learning (ML) has demonstrated applicability to represent complex nonlinear mappings. ML is a widely used approach and is based on making a computer program learn and adapt to new inputs or situations without human interference. Due to its huge benefits in computation costs, structural and material scientists have found ML useful and have implemented it in different forms which can be found in early [41, 42] and recent [43] review papers. A few major advancements are reviewed in the following paragraph to highlight the tremendous potential of ML that has been harnessed by researchers in computational mechanics and materials.

Researchers have explored the application of ML in structural health monitoring for damage detection extensively since the 1990s [44, 45] using the large amount of data available. Due to the difficulty in defining analytical relations between damage patterns and measurements obtained during non-destructive evaluation of critical structures, extensive efforts have been made to simulate the inverse problem. This provides means to relate scattered surface displacement response from ultrasonic defect evaluation parameters. [46, 47, 48, 49, 50, 51]. Topology optimization [52, 53, 54, 55, 56, 57] and microstructure quantification[58, 59, 60, 61] are other areas where ML techniques have established promising improvements over traditional approaches. These research efforts have used and implemented a variety of ML approaches such as Bayesian algorithms, Monte Carlo simulations, k-nearest neighbors, support vector regression, and convolutional neural networks which indicate the possibility to explore many more areas for applications of ML.

Further, researches have been established to use data-driven based techniques to predict mechanical behavior of materials for problems that entail poor understanding to accurately describe constitutive equations [62, 63, 64, 65]. Data-driven modeling of nonlinear material behavior has also been established using various ML techniques such as auto-progressive [66], tensor based [67] neural networks and, symbolic regression [68]. Research effort presented by Kirchdoerfer et al.[69] shows the applicability of data-driven approaches for material modeling for cases in which empirical formulation requires extensive efforts. These models definitely provide alternate ways of describing constitutive relations but they do not overcome the challenge of high computational cost involved in predicting structural behavior at a macroscopic scale, which is required for analyzing mechanical assemblies.

Moreover, most of these efforts are derived from experiments that are inherently sensitive to the experimental design envelope and induced noise leading to significant uncertainty.

Considering the scope of the proposed work, at a macroscopic level, limited research has been established to compute data-driven computationally efficient high fidelity solutions [70, 71, 72, 73, 74]. A recent example is the data-driven selection of parameters for reduced order models (ROMs) [70, 71]. For example, in the digital twin [70] concept, traditional localization and optimization approaches of selecting an appropriate model from a previously developed library of ROMs for components is replaced with the Bayesian estimator. This method, though efficient compared to FEM, greatly suffers in nonlinear or history dependent problems due to the incapability or inefficiency of the underlying model. Smart Finite Elements (SFEs) are another example in this category, in which surrogate models are created via advanced ML techniques to describe structural behavior of complex macro elements [73]. However, even though SFEs works for nonlinear problems, they cannot handle history-dependence. For dynamic problems, a recent example is the estimation of influence function for a bond-based peridynamic model to study periodic structures. [74]. In this method, the influence function of a unit cell is estimated using linear regression. This method, though efficient compared to FEM and other numerical methods, cannot be used for nonlinear applications (when the kernel function evolves with time) and to model arbitrary structures (with more than one type of unit cell). As a final note, we use the term ROM for model order reduction techniques that are not data-driven. For cases when model order reduction takes place using ML, we use the term surrogate models or data-driven models and we follow the same terminology in the remainder of this thesis.

1.2 Scope and Organization

Through the present work, we propose a data-driven framework to analyze the structural behavior of mechanical components and assemblies in an efficient manner. To do this, we use surrogate modeling to represent dimensionally reduced data-driven models at the component level. The proposed approach enables the application of data-driven methods for nonlinear and history dependent problems such that all the information pertaining to these phenomena is preserved. That is, the surrogate model understands and learns this information during its development using a machine learning algorithm. This way, the developed component model can be used for efficient computations for various time-critical applications and studies.

We illustrate the proposed approach for different types of phenomena and nonlinearities such as structural health monitoring for components with localized damage, mechanical assemblies with contact, and wave propagation in large periodic/arbitrary finite structures. Further, we also propose several ways to enhance this approach in terms of precision and efficiency by including physics-based constraints during prediction to avoid violation of physical laws, developing a low noise sample generation technique for broad frequency spectrum applications, using a corotational formulation to enforce frame-indifference, and developing a bi-network approach for components that are operational for several hours in a dynamic environment.

In addition, we propose a novel meta learning approach: dual-step training approach to ensure prediction accuracy when there is a large spectrum of input parameters that need to be considered; for e.g. when a broad range of input frequencies need to be considered in the operational profile of a component.

Overall, the proposed work provides a general method that is computationally efficient in analyzing the nonlinear and history-dependent behavior at a component level and provides a promising path for the use of data-driven techniques to solve difficult nonlinear problems in a simple and concise manner.

The rest of this thesis is organized in the following way. In Chapter II, we introduce and develop our approach: smart parts, for using data-driven methods (machine learning based) to establish efficient input-output relationships for structural components. We illustrate our approach using multiple engineering applications that involve different types of nonlinearties. Then, in Chapter III, we extend the proposed method to solve dynamic (wave propagation) problems for finite structures made of arbitrarily repeating unit cells. Such an approach effectively homogenizes the dynamic behavior of a linear/nonlinear homogeneous/heterogeneous unit cell for a broad frequency spectrum. Finally, we conclude this thesis in Chapter IV by summarizing our main findings and defining future directions for our research.

CHAPTER II

SMART PARTS

2.1 Overview: Machine learning

In broad terms, machine learning refers to the application of an algorithm that enables a computer program to learn from past experiences and make predictions. There are mainly three types of machine learning techniques namely, supervised learning, unsupervised learning and reinforcement learning. The books in references [75, 76, 77] provide a good description for these methods along with various applications.

For our method, we use supervised learning and specifically artificial neural networks (ANNs). Mathematically, supervised learning makes use of an optimization algorithm to find a set of parameters of a function $\mathcal{F}$ that approximates the input-output relationship such that the approximation error E is minimized. It does this by a process called as *training*, in which the algorithm iteratively makes predictions and checks the approximation error with respect to the actual or truth outputs. One of the most commonly used supervised learning technique is linear regression, which can be used when the output parameter is a linear function of the input parameters. However, when complex and nonlinear relationships need to be represented, a more powerful technique such as ANNs is required. For the proposed work, we use a specific version of this technique called recurrent neural networks (RNNs), which is useful for sequence to sequence problems and enables history retention. For completeness, we provide a basic description for ANNs and RNNs in the subsequent paragraphs.

2.1.1 Artificial Neural Networks

In this section, we provide a short description of artificial neural networks (ANNs) for completeness. Details and theory can be found in the research established by Braspenning et.al [78] and LeCun et.al [79] for interested readers as a full explanation is beyond the scope of this chapter. In general, ANNs can be viewed as a mathematical function that consists of certain inputs, outputs, and weights. These weights form the mapping of the function from inputs to outputs through a hidden layer as shown in Fig. 2.1. The process of deriving these weights requires a large set of inputs and outputs (supervised learning) so that the domain of the problem is well defined.

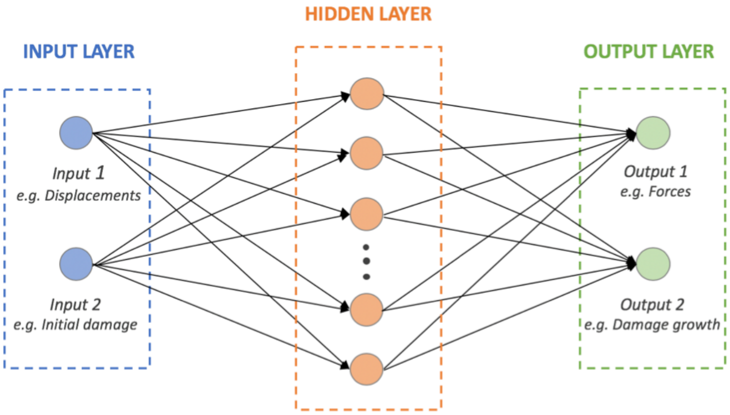


Figure 2.1: Fully connected three-layered artificial neural network linking nonlinear structural parameters

A single iteration through a traditional or feedforward neural network is as shown in Fig. 2.2 for an input signal through one neuron. The network first makes certain output predictions using initial weights and biases in the forward propagation step as shown in Eq. 2.1.

Consider the input to be denoted by x_i and the model parameters namely, weights and biases for input-to-hidden layer by w_{hi} and b_{h0} respectively. The biases are used as an intercept value to make the model more general and provide for more parameters in order to search for the best possible solution. The input to the hidden layer is the weighted

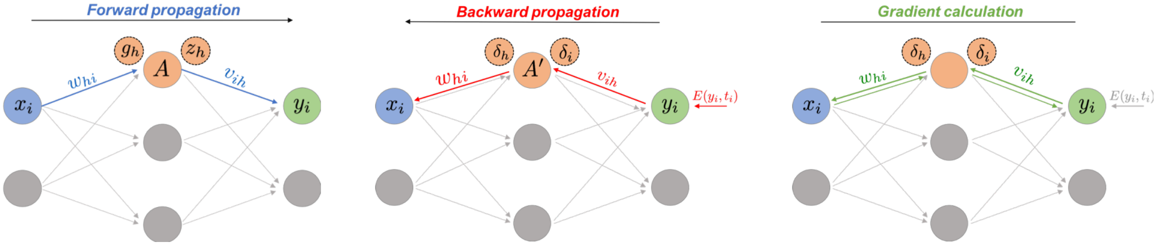


Figure 2.2: Single iteration procedure for multi-layer neural network

sum of the inputs which is then “activated” in the hidden layer. This is required since a network with only input-output layers can approximate linear functions and cannot be used for nonlinear mathematical mapping. Presence of a hidden layer makes the mapping nonlinear, however, more complex mappings can be obtained by using different activation functions (A) available in literature. The output from the hidden layer (z_h) serves as the input to the output layer (y_i) using different set of weights (v_{ih}) and biases (b_{i0}). It is important to note that, there can be an additional activation stage at the output layer if needed.

$$\begin{aligned} g_h &= \sum w_{hi} x_i + b_{h0} \\ z_h &= A(g_h) \\ y_i &= \sum v_{ih} z_h + b_{i0} \end{aligned} \tag{2.1}$$

Once the outputs y_i are predicted, they are compared with truth values (t_i) which are the correct outputs as per the data. Quantitatively this comparison is performed using an error or loss metric (E). The error is backpropagated towards the input as an error signal (δ_i) by passing it through the output layer weights (v_{ih}) and summing over all output nodes. This result is then passed through the gradient of the activation function at the hidden layer (A') to obtain the error signal at the hidden layer (δ_h) as shown in Eq. 2.2.

$$\begin{aligned} \delta_i &= E'(y_i, t_i) \\ \delta_h &= A'(g_h). \sum \delta_i v_{ih} \end{aligned} \tag{2.2}$$

These error signals and input-output parameters at different layers are used to calculate gradients of the error function with respect to the weights at each layer. These gradients can then be used to update the weights of the network as shown in Eq. 2.3 with a user-defined learning rate (η). This in general, constitutes a single iteration during the training process. Training continues until the error reduces to an acceptable value (E_{accept}) such that when the network is used to predict an unseen input, an accurate output can be expected.

$$\begin{aligned}\frac{\partial E}{\partial w_{hi}} &= x_i \delta_h \\ \frac{\partial E}{\partial v_{ih}} &= z_h \delta_i \\ w_{hi} &= w_{hi} - \eta \frac{\partial E}{\partial w_{hi}} \\ v_{ih} &= v_{ih} - \eta \frac{\partial E}{\partial v_{ih}}\end{aligned} \tag{2.3}$$

It is important to note that, traditional or feedforward neural networks cannot be used for sequence to sequence problems. This is because, in traditional neural networks connections between different nodes do not form a directed graph, thus rendering them incapable to use internal states (memory) of the network to process sequence inputs. Hence, to address this issue we use recurrent neural networks.

2.1.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) address this issue and use the current and previous input (history information) to predict the current output. Hence, in RNNs there are connections that feed output values of a hidden layer into its own input/s of the preceding layers. This can be demonstrated in a simple manner as shown in Fig. 2.3, wherein the output of the hidden layer not only goes as input to the next layer but is also fed back as an additional input onto itself. This aspect enables the model to store relevant history information which allows us to use it for temporal (sequential) evolution.

Training the recurrent neural network is similar to traditional neural networks such that

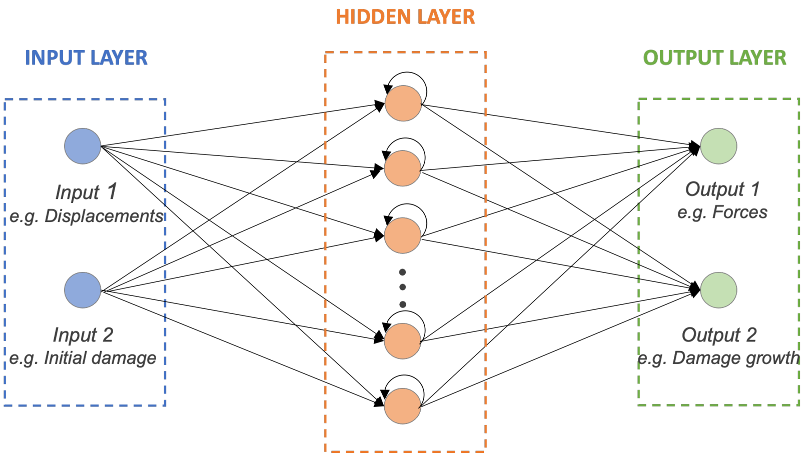


Figure 2.3: Example of a recurrent neural network with with output of hidden layer also used as its own input

each sample consists of a number of sequences for the input as well as the output. Note that, to apply the backpropagation algorithm, the libraries of the layers first *unroll*, that is, they expand the network, generating a copy of the network for each time increment and connecting the inputs and outputs of consecutive instances. Then, the weights and biases in the network are trained using traditional backpropagation on the unrolled network.

On the downside, feeding the output of a layer into its input vector leads to short memory spans and suffers from the problem of vanishing gradient when the gap between the previous relevant information and current step is too large. Hence, models with long dependencies such as the long short-term memory (LSTM) [80] models are extensively adopted by researchers to overcome the above problem. LSTM models are designed to have dedicated memory cell states to regulate the information flow through the network. Hence, in our approach we use LSTM models to design the neural networks that enable sequence-to-sequence prediction of structural parameters.

Finally, it is important to note that the main approach of smart parts, uses this recurrent neural networks for sequence regression. However, we also make use of other machine learning techniques such as logistic regression for classification of inputs and gaussian processes for sample generation. This overview only contained the minimal information needed for understanding and implementing the proposed approach. Details for other techniques that are used for specific tasks are explained during the description of those tasks later in this and the next chapter.

2.2 Proposed approach

2.2.1 Concept

FEM is used in stress analysis to compute approximate full-field solutions for stresses and strains within mechanical components, which could be either isolated or part of a larger structural or mechanical assembly. Even though this approach can be simplified via model order reduction for linear and some nonlinear problems, the full-field FEM solution is still needed when problems are nonlinear and history-dependent. In this scenario, we typically need three main equations to compute the solution (finding displacements: $\underline{u}$) of the boundary value problem, namely, equilibrium, kinematic and constitutive, as described below:

$$\nabla \cdot \underline{\underline{\sigma}} + \rho \underline{b} = 0 \tag{2.4}$$

$$\underline{\underline{\epsilon}} = \frac{1}{2}\big(\nabla\, \underline{u} + \nabla\, \underline{u}'\big) \tag{2.5}$$

$$\underline{\underline{\sigma}} = \underline{\underline{\sigma}}[\Delta \underline{\underline{\epsilon}}(\Delta \underline{u}_j), Q] \tag{2.6}$$

These equations describe the behavior of a component B in an assembly subjected to body forces $\underline{b}$ along with displacement boundary conditions $\underline{u}^*$ on the surface $\partial_1 B$ and tractions $\underline{t}^*$ on the remaining surface $\partial_2 B$. For a material exhibiting path-dependence, the obtained displacement is a function of its history and hence the stress ($\underline{\underline{\sigma}}$) can be expressed as a functional (Eq. 2.6) which depends on the strain increment ($\Delta\underline{\underline{\epsilon}}$) and internal state variables (Q). These state variables [81], e.g. equivalent plastic strain, introduce path-dependence into problem.

In general, the FEM approach is very powerful and can handle a large variety of problems, but becomes unnecessarily detailed when analyzing assemblies with multiple components, for which we may not need to know details related to the internal stress state of their components. Instead, assembly analysis could be performed with relatively simple relations between inputs (displacements at contact locations between components) and outputs (corresponding forces at such points). That is, despite being nonlinear and history-dependent, the input-output structural parameter relationships can be represented as concise and simple mappings at these representative locations in various components, which could then replace the cumbersome finite element analysis (FEA) of the entire assembly.

In this work, we propose to generate such mappings by making use of *artificial neural networks* (ANN) trained with detailed finite element models of individual components. In this way, the ANN acts as a dimensionally-reduced surrogate model capable of predicting the corresponding nonlinear and history-dependent mechanical behavior.

For the purpose of illustration, let us consider a landing gear assembly as shown in Fig. 2.4. The simplified assembly consists of three discretized components or sub-assemblies attached to each other using appropriate kinematic constraints. Note that in this case, the existence of small features such as holes, requires a fine discretization for an accurate solution, thus increasing the computational cost. Using our approach, we can develop and use component surrogate models (*smart parts - SPs*) for components or sub-assemblies such that they can be assembled together using appropriate kinematic constraints, while capturing the effective history-dependent behavior of the component. The proposed approach is also general in terms of input-output combinations and types of nonlinearities. For example, it can be implemented to predict the deformation of the component under the effect of applied forces or vice-versa.

In general, data-driven models using ANNs are developed by making them learn from truth input-output data, the process commonly known as *training*. We train the SP using

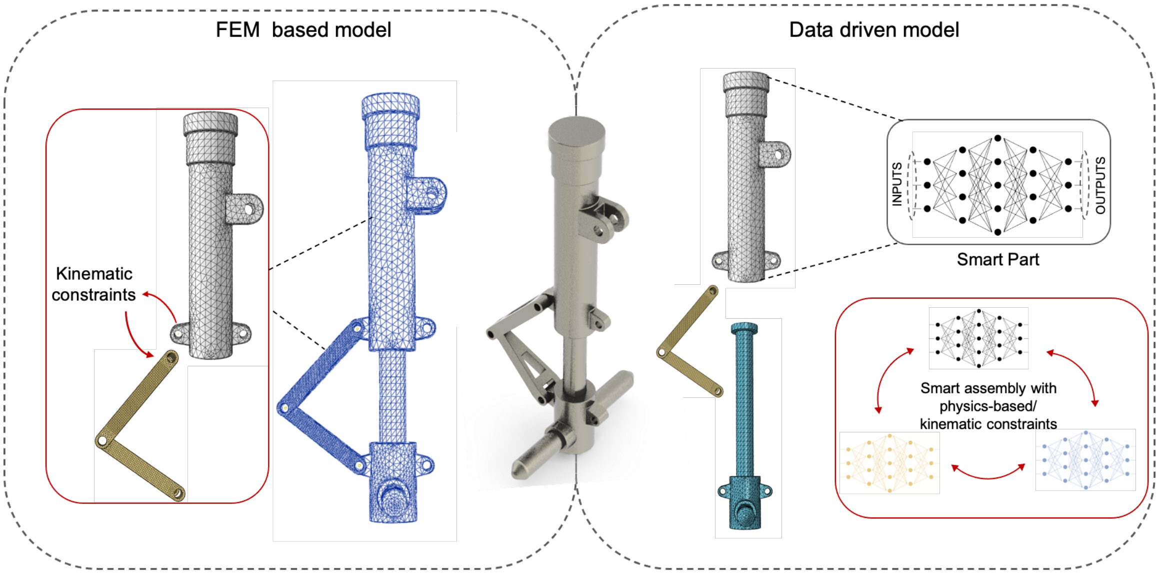
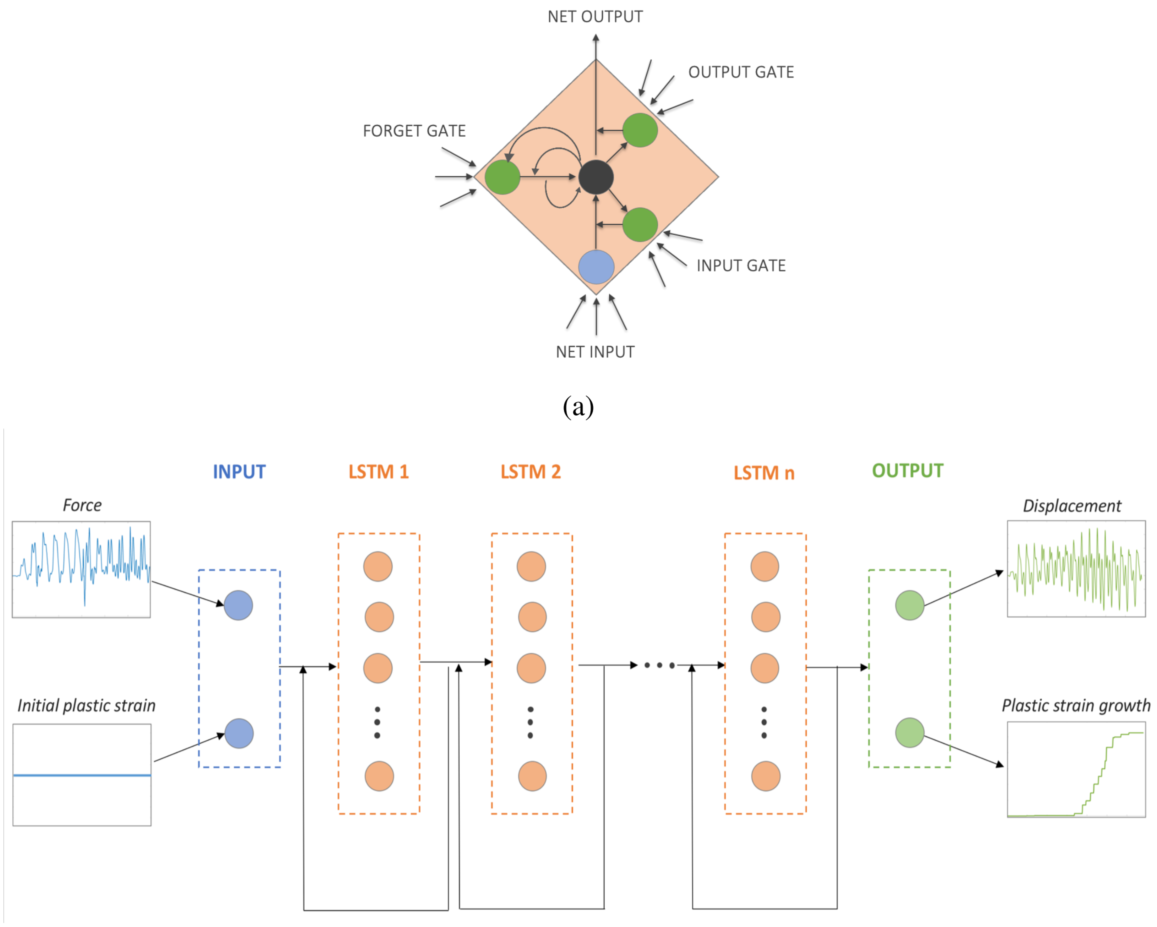


Figure 2.4: Motivation and approach schematic using a landing gear assembly

high fidelity solutions computed by mesh converged FE analysis to formulate a sim-to-sim approach. The training data thus retains all the nonlinear information such as contact, damage parameters, etc., which may exist in addition to the boundary conditions under which it operates. As explained earlier in Sec. 2.1, we use *long short-term memory* (LSTM) models, an RNN architecture to learn these nonlinear patterns for various parameters that may be history-dependent and finally use all this information for future predictions. Unlike simple RNN, LSTM neurons do not suffer from the problem of vanishing gradient due to short memory spans. Hence, it is useful for retaining information over larger time intervals. LSTM neurons enable history retention by using special memory cells [82] that are controlled by a forget gate as shown in Fig. 2.5a. These gates have internal states which represent the information used by the neuron to make a prediction. Hence, the output from each neuron is modulated by these internal states allowing the model to learn relevant information. Multiple such neurons and layers of neurons are needed to accurately map the nonlinear patterns between structural parameters. Fig. 2.5b shows a typical problem schematic to develop a SP that uses n LSTM layers with multiple neurons to map the inputs of force and initial plastic strain to displacement and plastic strain growth.



(a)

(b)

Figure 2.5: LSTM architecture schematics. (a) Single LSTM neuron for history retention, (b) Data-driven model schematic to develop history-dependent SP

2.2.2 Implementation

The overall implementation methodology of our approach, including from sample generation to model deployment for a SP, is depicted in Fig. 2.6. Each step can be performed in various ways to suit different kinds of problem definitions, as it will be shown through different case studies in Sec. 2.3. In general, we use the specific problem definition to generate samples using FEA due to its high accuracy in solving nonlinear problems. This enables us to obtain an accurate and sufficiently large training set that typically determines the strength of the domain knowledge for our ML model.

The samples are then split into a training set and a test set. The proportion of this split is such that 10 - 20$\%$ of the samples are used in the test set. This value is decided based on the size of the overall set and by ensuring that both the training and test sets represent the domain with similar distributions to avoid bias during training or validation.

The training set is used to train the selected model architecture which for the purpose of this study is LSTM sequential models. The model architecture includes parameters such as number of layers, number of neurons in each layer, batch size, activation functions in dense layers, etc. These parameters vary for different models and need to be selected based on the complexity of learning involved in the problem as well as the strength of domain knowledge. We select and fine-tune these parameters by conducting a sensitivity study to understand the effect of these parameters on model accuracy as illustrated in the example applications (Section 2.3). During training, the model is monitored for accuracy using a loss metric which acts as an optimization score function and defines how well the model learns after every epoch (number of times the learning algorithm sees the dataset). We use mean squared error (E_{MSE}) as the loss function, which is given by:

$$E_{\text{MSE}} = \frac{1}{n}\sum_{j}^{N}(\boldsymbol{y}_j^o - \boldsymbol{y}_j^p)^2 \tag{2.7}$$

where $\boldsymbol{y}_j^o$ and $\boldsymbol{y}_j^p$ represent the observed output and the predicted output by the network respectively and N is the number of samples used for training.

Once the model with a reasonably low loss is trained, it is used for the prediction of cases in the test set. This step may either result in finalizing the model and its deployment if the prediction is within the defined tolerance, or in optimizing the architecture of the model and re-training it. Once a SP is developed, it can be used with other such parts in an assembly to depict its structural behavior.

In the next and its subsequent section, we illustrate the proposed approach on realistic components and assemblies exhibiting fundamentally different problem definitions and combinations of nonlinearities. Further, we provide ways to enhance the data-driven mod-

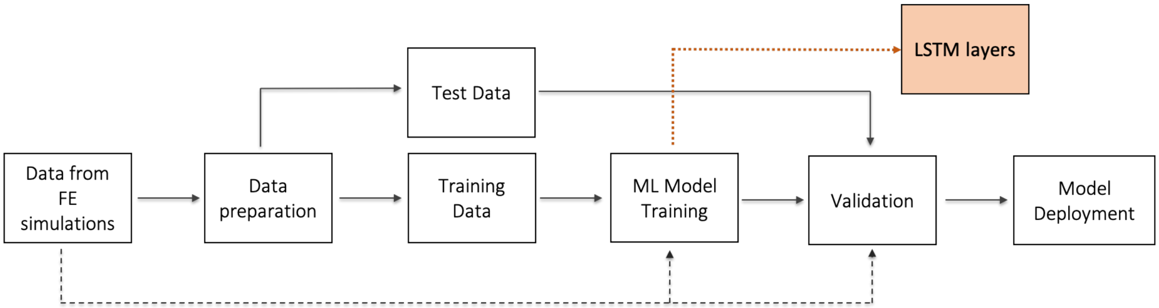


Figure 2.6: Overall implementation methodology

els so their capability or accuracy can be improved. These enhancements are meant to illustrate how the approach can (and should) be tailored depending on the problem at hand. Finally, we present a comparison for the computational cost required by the proposed approach and FEM.

2.3 Case studies

The overall approach for different problems remains consistent as explained in the previous section. However, depending on the problem at hand, one or more steps may require additional considerations as exemplified in the various case studies in this section. In the first study, we illustrate our approach using the helicopter pitch link since, life prediction of critical components subjected to dynamic loads has been a challenge due to the prohibitive computational cost associated with time-domain simulations. Current methods rely on sensor data gathered during operation which cannot be analyzed in real-time. These limitations generally lead to conservative component life estimates. We address this challenge in the first example to provide an approach that can monitor localized damage on a helicopter pitch link in real-time. In practice, predictions from such a model could be utilized with appropriate control schemes for component life extension applications. Further, we also illustrate specific enhancements to this application by developing a bi-network approach able to improve the precision and efficiency of the model.

Similarly, simulating contacts has been made possible with the use of FEM, but with significant computation cost. This cost worsens with the presence of small features such as holes. Hence, we demonstrate the capability of our approach to handle such a problem in an efficient fashion.

In general, ROMs can only establish input-output relationships in components/ assemblies. Hence, such models do not provide the means to monitor localized parameters such as internal damage. So, we use the growth history of maximum equivalent plastic strain PEEQ$_{max}$ as an output to include this capability for both examples. PEEQ$_{max}$ acts as a localized damage parameter and helps in monitoring the critical damage-causing value even if the rest of the component behaves in a linearly elastic manner. This parameter serves to be crucial especially in predicting or identifying structural failure which may not be evident due to the elastic behavior of the overall component. For the purpose of this work, we use classical metal plasticity model with isotropic hardening [83] to simulate the post-yield behavior during the sample generation process in FEA to represent material nonlinearity.

In case, other structural parameters are of interest, they can be included as additional features in the ANN while training the SP. Also, note that, even though we consider material nonlinearity and contact in the problems under consideration, nothing prevents this approach to include other types of nonlinearities, such as geometric nonlinearity, or to consider other nonlinear material models, e.g., nonlinear elasticity, etc.

2.3.1 Helicopter pitch link

Many real life structural problems include dynamic effects and hence require solving cumbersome dynamic equations to describe the behavior of the component or system under consideration. Moreover, reduced order modeling techniques for dynamic problems such as modal superposition can be applied only when the problem is linear and hence face limitations in applications. This can be avoided by the presented approach. To establish the capability of SPs for structural health monitoring applications, we use a helicopter pitch link [84, 85] to predict displacements and damage evolution from applied dynamic forces at its attachment points.

Pitch link problem definition

We develop a representative model of the pitch link, utilizing properties for 7075-T6 Aluminum in order to carry out the FEA. The mass and length of the pitch link were estimated utilizing data obtained through a NASA Contractor Report [86] carried out by Sikorsky Aircraft and the radius of the pitch link was calculated using these parameters. The pitch link is connected to the pitch horn, as shown in Fig 2.7a, via a pin joint. Hence, to depict the behavior accurately, the dynamic loads are applied at the center point in the pin-hole (reference node A) which gets distributed on the internal surface so that the motion of the internal surface is governed by that of the reference node as shown in Fig 2.7b. We restrict reference node A for displacements in X and Z directions and rotations in X and Y directions. The other reference node (reference node B) is restricted for displacements in X, Y, and Z directions and rotations in X and Y directions.

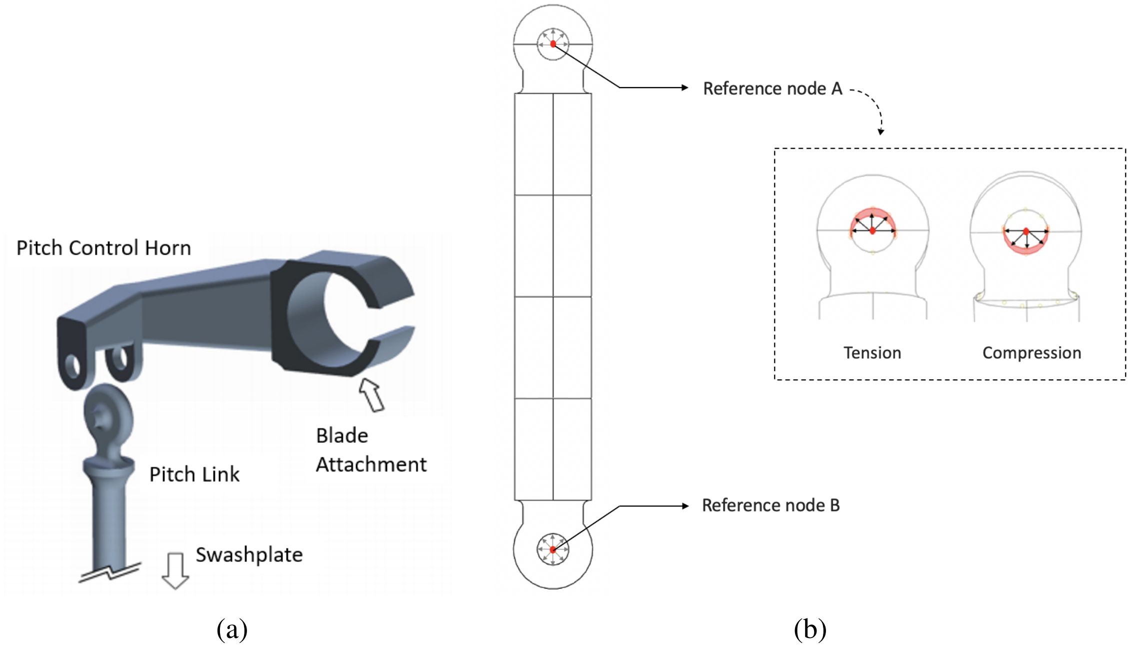


Figure 2.7: Boundary condition set-up for pitch link (a) Pitch link and pitch horn asssembly of a rotor [87], (b) Schematic of pitch link model for FEA

Training sample generation

We develop a full dynamic time marching analysis using 3-D FEA to generate training samples. Dynamic forces generated during a maneuver such as cyclic longitudinal doublet were used as inputs. We implement a Linear Time Invariant (LTI) scheme coupled with a Linear Quadratic Estimator (LQE) [88] to estimate real-time pitch link loads. A representative profile for control input profile with the corresponding estimated loads in time domain along with its frequency content is shown in Fig. 2.8. The frequency content is obtained using a fast fourier transform (FFT), which is then normalized with the rotating frequency (4.3 Hz) of the pitch link. Dominant peaks at 1, 2, 3, and 4/rev are observed in the estimated family of dynamic loads. Then, we generate multiple such profiles using different control input parameters to generate the training samples. For the considered family of maneuvering loads, two control input parameters are primarily of interest: the aggressiveness of input and percentage magnitude of the command input. The aggressiveness of the input is represented by the rise time of the input such that a rise time variation of 0.1s in the control input resembles close to a step input (more aggressive) compared to 0.5s which has a smooth ramp. The percentage magnitude of the control input is a measure of the movement of the control stick from its neutral position and hence in a way reflects the magnitude of the maneuver and the corresponding dynamic loads. For this work, we use a rise time range of 0.1s to 0.5s and a percentage control input magnitude range of 5 to 50%.

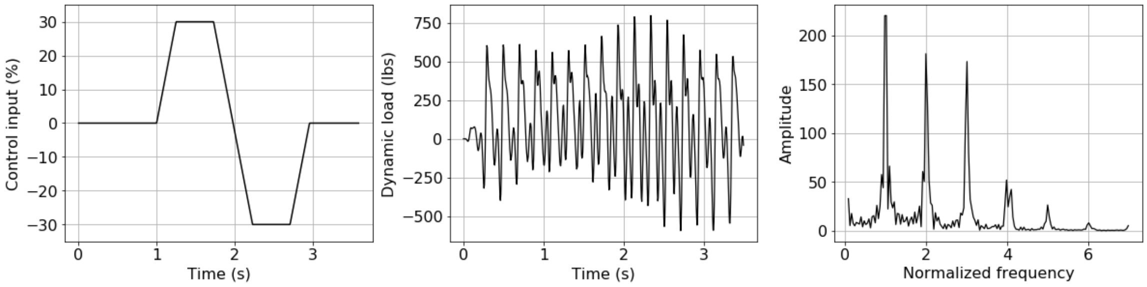


Figure 2.8: Representative sample of dynamic loads (from left to right): control input profile, dynamic loads in time domain, frequency content of dynamic loads

Note that there may be cases such that the initial plastic strain is non-zero when the pitch link is under operation due to the presence of defects or due to previous history. Hence, to construct a well-represented training set, we need to have training samples that represent cases with preexisting localized plastic strain in the smart pitch link. We do this using ten different and randomly generated load profiles, connected in series to form a *parent profile* for the dynamic FE analysis. After the analysis, we divide the parent profile back into load profiles of the initial length along with their respective outputs (displacements and maximum equivalent plastic strain). This way we generate 700 parent profiles which result in 7000 training samples spread over a range of control inputs and initial plastic strains. We show the corresponding sample distribution in Fig. 2.9 as a function of command/ control magnitude and rise time such that the intensity of the color in the plot represents the density of samples. Univariate distributions of the samples are also shown on the top and right side of the plot for magnitude and rise time respectively. We observe that the final training sample set has a fairly uniform distribution in terms of magnitude as well as rise time. This is important, since a biased sample set in terms of the domain it needs to represent may lead to a biased SP. Note that in Fig. 2.9 we do not show any information about the location of a particular sample in the parent profile, that is whether the dynamic load corresponding to that sample is applied on a pristine pitch link or one which has some preexisting localized damage. We only show that a sample with a particular magnitude and rise time is considered somewhere in the training. Hence, the neural network has to learn these aspects and should be able to accurately represent the pitch link for any such operating condition.

Neural Network Architecture and Results

We train two separate models, one with force-displacement and the other one with force and initial plastic strain-maximum equivalent plastic strain growth as the input-output combinations. This was done to have two independent models that can be used separately for

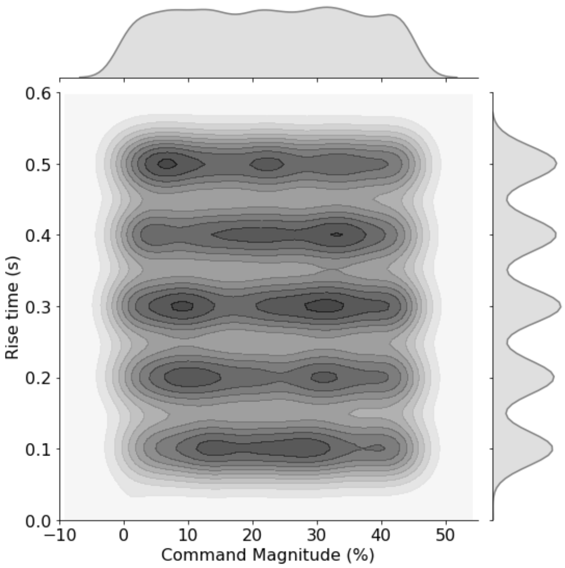
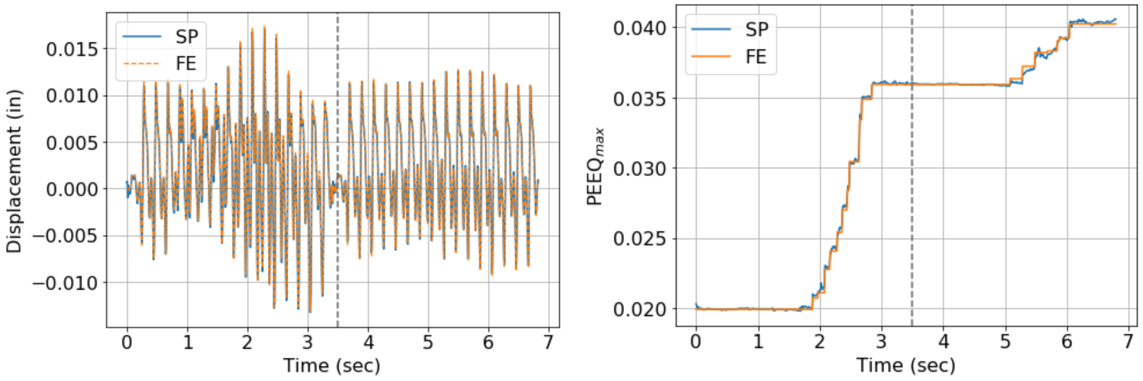


Figure 2.9: Sample distribution for pitch link

predictions if needed. It also relaxes the training complexity due to the significant difference in the frequency content of the two outputs, both of which need to be predicted with the same input. Moreover, note that the precision required for the two output features, namely displacement and maximum equivalent plastic strain (damage growth) is different. That is, we need more precision for the predicted displacements in its temporal evolution when compared to that needed for the damage growth prediction wherein it is sufficient for the prediction to be in a certain acceptable range. Hence, for this case, a combined model will not only entail the use of a highly complex model architecture but will also cause an imbalance in the precision of the predicted output features. A combined model can also be developed if needed with a deeper network and larger hyper-parameters.

It is important to realize that the orders of magnitude for each of the input features and each of the output features might be considerably different. Hence, we need to scale the features so that the variance of all features is in the same range. If not, the feature with a higher variance dominates the training and hence the predicted output of the model. We use the *min-max scaling technique* which brings all feature values in $[0, 1]$ range. We train 1 LSTM layer with 2 dense layers each of size 200 for the force-displacement model and 3 LSTM layers with 3 dense layers of size 600 for the damage evolution network. The latter model is more complex due to the highly nonlinear nature of mapping that the statistical model needs to infer from the high-frequency dynamic loads to the step-like function represented by the localized damage growth. For both the cases we use the Adam optimizer [89] with a validation split of 20% and mean squared error (Eq. 2.7) as the loss function for training. We test the trained model with unknown cases fed to the model to validate the developed smart pitch link. Further, the model was sequentially called as a function so that multiple random profiles can be tested in series with information of the end plastic strain from the preceding profile. Figures 2.10a and 2.10b show a comparison between the predicted outputs (SP) versus the computed outputs (FE) from 3-D FEM based approach. The vertical line in these results separate the consecutive prediction profiles. The results show comparable predictions with 3-D FEA including in the region of transition from one profile to another.



(a)

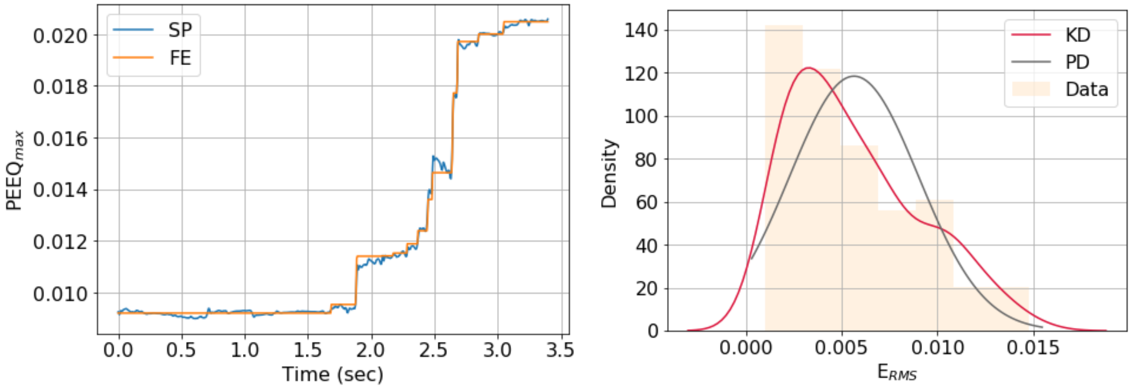
(b)

Figure 2.10: Comparison of predicted outputs for two sequences applied in series to smart pitch link. (a) Displacement response, (b) Damage evolution

This serves as an impetus for realistic application in structural health monitoring and life extension of critical dynamic components. It is therefore, essential to relate damage parameters with the *harmonic content* of the applied dynamic loads. We develop a smart pitch link to map the harmonic content of the input dynamic forces to the maximum equivalent plastic strain (indicates localized damage) growth. We obtain the time varying harmonic content of the input forces using the moving window FFT approach. The size of the sampling window is 1 revolution and the sampling frequency is $1/\Delta t$ where Δt is the time increment considered for the history. We develop and train this model using 3 LSTM layers with 800 neurons and 2 dense layers of size 800. Results indicate good accuracy when matched with outputs obtained from FEA in Fig. 2.11a. Figure 2.11b shows the residual error distribution of root mean square (RMS) error (E_{RMS}) for 100 unknown samples computed using:

$$E_{\text{RMS}} = \frac{1}{max|f_a|}\sqrt{\frac{1}{l}\sum_{j=1}^{l}(f_{aj} - f_{pj})^2} \tag{2.8}$$

where, f_a is the truth output value obtained from FEA, f_p is the output value predicted by the SP and, l is the length of the sequence. The density of errors can be visualized as a continuous function using the kernel density estimation (KD) [90]. This estimate is higher for densely located points, which in this case lies in the neighborhood of 0.25% error. A probability density function (PD) is also fitted for the E_{RMS} values, which result in a mean error of 0.55% and maximum error of 1.5%. These low error values thus establish that the model is well-trained and has good accuracy.



(a) (b)

Figure 2.11: Harmonics based model results. (a) Localized damage prediction, (b) Residual error histogram and distribution

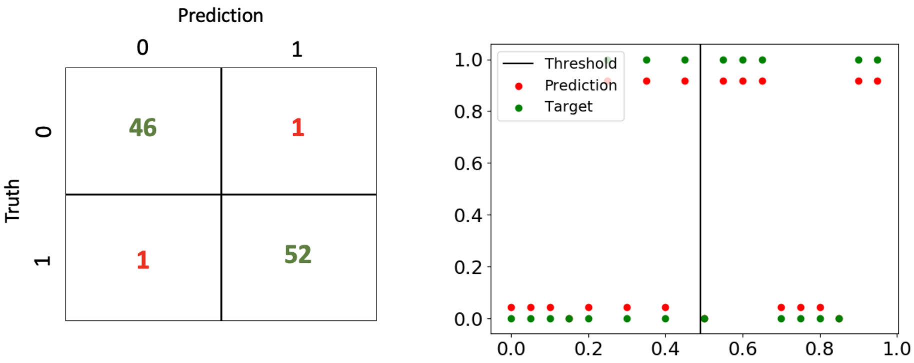
Helicopter pitch link Enhancements: Bi-network approach

In most flight conditions, the pitch link operates at relatively low loads for long periods of time, resulting in no damage growth. On the other hand, some maneuvers may impart loads large enough to trigger damage growth. One potential pitfall of our previous approach is that an *always on* model might accumulate small errors in localized damage growth over long periods of time, resulting in an underestimation of component life. To address this, we developed a *classification model* that acts as a *switch* and determines if the damage evolution model needs to be triggered or not. We train the classification model using a combination of dynamic loads which may or may not result in localized damage growth as inputs and binary outputs (switch on: 1 and switch off: 0). We utilize 1 LSTM layer of size 100 with an output dense layer activated by a sigmoid function [91]. Further, we adopt the stochastic gradient descent optimizer with a fixed learning rate of 1, since the output feature is not a sequence and a momentum of 0.1 to damp noisy predictions during training. Since this model is a classification model, we use binary cross entropy (E_{CE}) as shown in Eq. (2.9) as the loss function and also observe accuracy (Eq. (2.10)) of the model as an additional metric. In Eq. (2.9), y_{ai} is the binary label (1 or 0), y_{pi} is the predicted probability of the sample i and, n_s is the number of samples used for training.

$$E_{\text{CE}} = -\frac{1}{n_s}\sum_{i=1}^{n_s} y_{ai}\, log(y_{pj}) + (1 - y_{ai})\, log(1 - y_{pj}) \tag{2.9}$$

$$\text{Accuracy} = \frac{\text{No. of correct predictions}}{\text{Total no. of predictions}} \tag{2.10}$$

We obtain an accuracy of 94% for the switch model during training with 98% correct predictions when tested with 100 test samples as illustrated through the confusion matrix in Fig. 2.12a. Of those, we select 20 random test samples to compare the target and prediction results in Fig. 2.12b.



(a) (b)

Figure 2.12: Switch model accuracy. (a) Confusion matrix, (b) Prediction for 20 test samples

Further, to understand the effect of individual harmonic components of the dynamic load on the damage parameter, we compute gradients of the output plastic strain history with respect to the input harmonics. Results for the gradients along with the combined model framework for the smart pitch link are shown in Fig. 2.13. Observations from Fig. 2.13 indicate that the switch is *on* from 0 sec to 7 sec, that is when dynamic loads capable of growing localized damage are detected. It turns *off* at 7 sec after the maneuver is complete, which precisely depicts an accurate functioning of the overall model. The damage evolution model is again triggered at 14 sec when the switch model predicts a *yes*, to predict the maximum equivalent plastic strain growth. More interestingly, Fig. 2.13 also shows the computed gradients of the output feature with respect to individual harmonics. As can be observed for this sample case, 1/rev, 2/rev, and 3/rev harmonics are contributing more towards damage during different periods of operation. This information can be used in conjunction with an appropriate controller to reduce their effect and increase component life in addition to monitoring the health of the component. Hence, these enhancements overall aid in structural health monitoring by reducing the overall prediction error along with determining more damaging inputs.

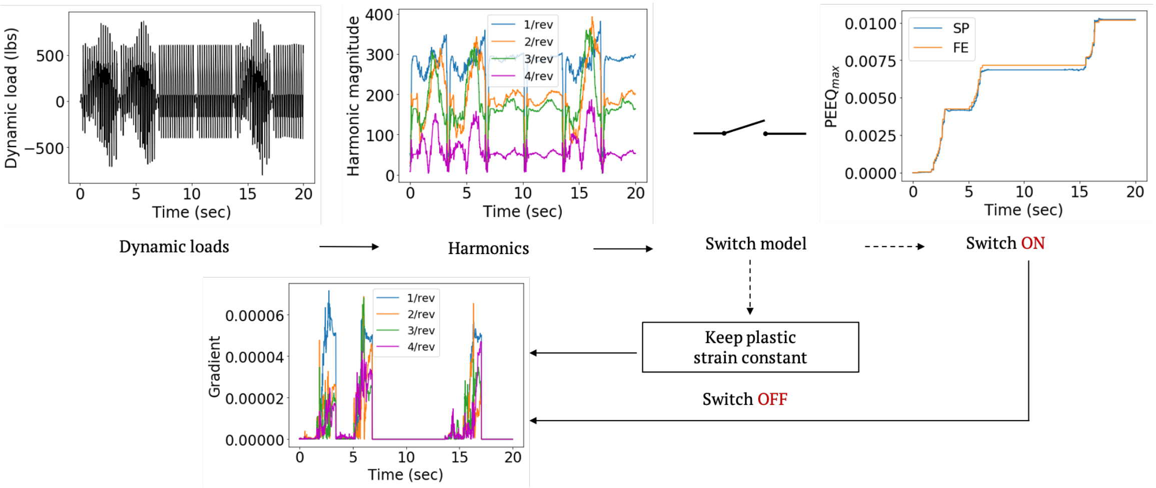


Figure 2.13: Schematic and results for enhanced model

2.3.2 Lap-joint

We illustrate our approach using a lap joint that is used in a wide spectrum of components ranging from pressure vessels to construction bracing and even as fastening joints in aerospace structures. The lap joint design and detailed finite element mesh modeled using FEA is shown in Fig. 2.14a and the analogous surrogate model schematic in Fig. 2.14b. The design makes use of two rectangular flanges connected with an M8 bolt. In addition to the contact nonlinearity present in the lap joint, we include material nonlinearity using plasticity data [92] for industry grade stainless steel 304L.

Training samples generation

We design a full 3-D finite element model of a lap-joint operating under quasistatic conditions for training, which takes displacements as the input degrees of freedom to compute forces and maximum equivalent plastic strain as outputs. We prescribe displacements at the eight corner nodes shown in Fig. 2.14b. To obtain displacements, we generate Gaussian processes and apply corotational formulation as explained in the subsequent paragraphs.

Since we are interested in history-dependent problems due to the high computation cost associated with them, we represent the displacement input at each corner node as a function

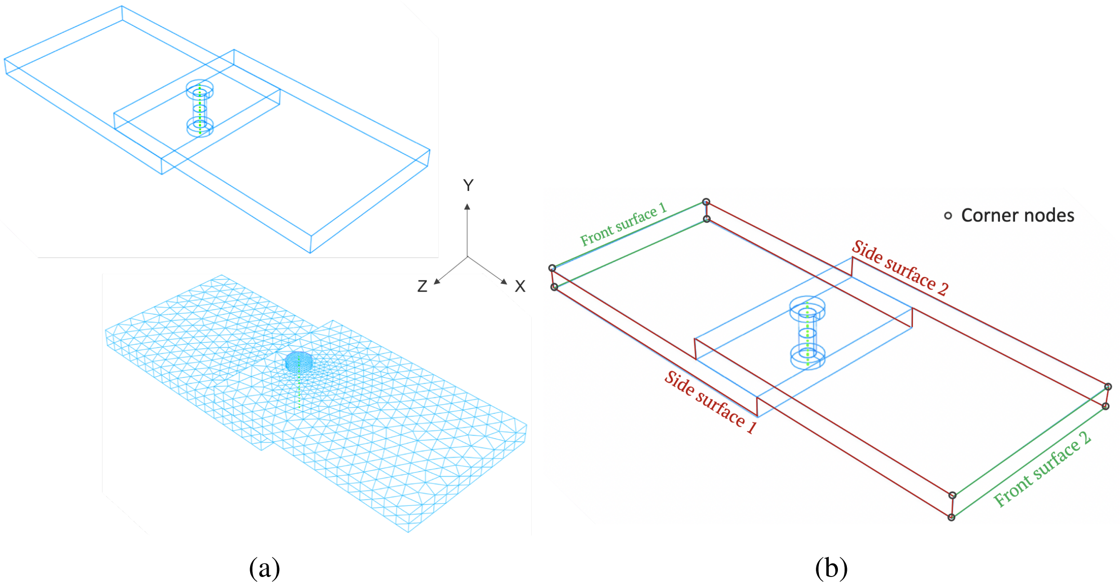


Figure 2.14: Lap joint model details. (a) Geometry and Mesh for Lap joint, (b) Boundary surfaces and nodes for Lap Joint model

of time to depict a deformation path that the SP might take. We use *Gaussian processes* to generate these input sequences with known input properties. Gaussian Processes are the multivariate case of a Gaussian distribution [93]. Hence, each random variable follows a Gaussian distribution and their joint distribution is also Gaussian. These processes are most commonly used in supervised learning to solve regression problems due to their ability to generate smooth and continuous sequences. Moreover, they have been used to generate samples in the form of strain invariants to develop smart constitutive laws [94] with very good sample representation.

To generate the input sequences, we first need to define certain input properties such as the mean or expected value, covariance, frequency, and amplitude of the function along with the initial conditions of the system. Unless there is a prescribed movement of the part that we know apriori, the mean of the function is assumed to be zero. The covariance function/kernel is the next parameter to be decided. This decision is important since it determines the smoothness of the generated function by ensuring the values that are close together in the input space will produce output values that are close together. There are many established kernel functions such as the Constant, Radial basis function (RBF), Matern, Rational Quadratic, etc. Readers are suggested to explore each of these kernel functions to analyze and decide based on the problem at hand [93]. For the lap joint problem considered here, we select the RBF kernel described in Eq. (2.11) parameterized by a length scale parameter, $l > 0$ which is of the same dimension as the input x. This type of kernel is a stationary kernel and hence depends on the distance between two data points and not on their absolute values $k(x_i, x_j = k(d(x_i, x_j))$, rendering them useful to represent nonlinear trends.

$$k(x_i, x_j) = exp\bigg(- \frac{1}{2} d(x_i/l, x_j/l)^2 \bigg) \tag{2.11}$$

We define the rest of the input properties based on the prescribed input limits for the problem. The frequency of the function is defined depending on the number of cycles of deformation that the part observes under operation. Similarly, the amplitude can be directly prescribed if known beforehand. If it is unknown, a value that results in a strain/stress close to the ultimate values (for this case 95% of ultimate values) can be defined so that the samples are well represented. Finally, we consider all deformation paths to start from an initial rest condition and impose zero initial conditions. As a result, we use the kernel with the all the defined parameters to create sequences for 24 independent variables of the training set representing the displacement of corner nodes in three directions. Figure 2.15 shows one such example of input sequences generated for all corner nodes.

The Gaussian processes generate the displacement input history at the four corner nodes of a surface. However, in a realistic scenario, the side and input surfaces (as defined in Fig. 2.14) will not be free surfaces and hence require a prescribed input. We use *bilinear interpolation* in space to obtain the input displacements at all the boundary surface nodes. For accurate interpolation such that it covers the entire side surface, we assume a bigger (rectangular) space (Fig. 2.16) and use the known values of displacement ($f(x_1, y), f(x_2, y), f(Q_{21}), f(Q_{12})$) to find the unknown ones ($f(Q_{11}), f(Q_{22})$) using

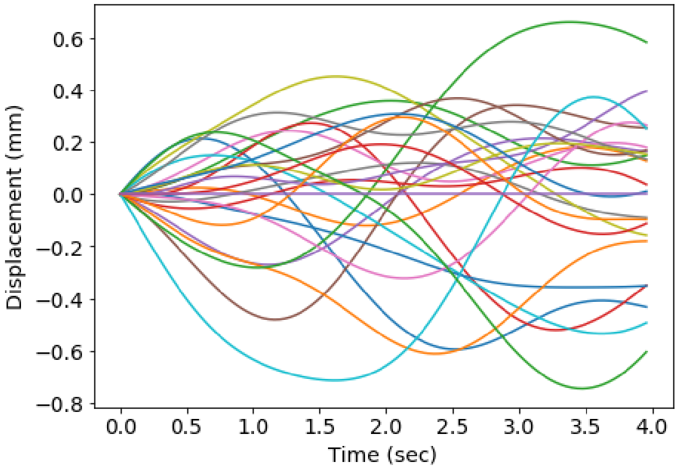


Figure 2.15: Sample Gaussian processes generated for corner nodes

Eq. (2.12). Interpolation equation in space (Eq. (2.13)) can then be applied using the coordinates of all nodes that occupy that projected space.

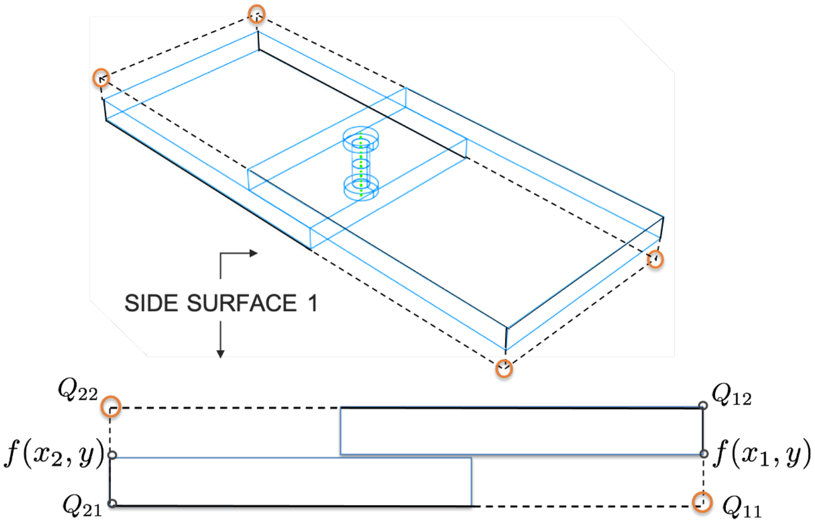


Figure 2.16: Bilinear Interpolation as applied on a side surface

$$
\begin{aligned}
f(x_1, y) &= \frac{y_2 - y}{y_2 - y_1} f(Q_{11}) + \frac{y - y_1}{y_2 - y_1} f(Q_{12}) \\
f(x_2, y) &= \frac{y_2 - y}{y_2 - y_1} f(Q_{21}) + \frac{y - y_1}{y_2 - y_1} f(Q_{22})
\end{aligned}
\tag{2.12}
$$

$$f(x,y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(Q_{11}) & f(Q_{12}) \\ f(Q_{21}) & f(Q_{22}) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix} \tag{2.13}$$

We enforce frame indifference for the SP by using a *corotational formulation* for the prescribed displacements on the lap joint. In addition to ensuring a frame indifferent SP by construction, using corotational displacements helps in significantly reducing the size of the training set. This is due to two reasons: (1) there is no need to represent the large range of displacements entailed by rigid body motion and (2) we don't have to include cases wherein a small change in the input displacements leads to a significant change in the output forces. Hence when we use corotational displacements, the range of samples is defined based on the allowable range of deformation, which is drastically smaller than that of rigid body motion. This renders a training process that is more computationally efficient.

For this work, we do this by creating a local reference frame that is attached to the SP such that it rotates and translates with the SP. By doing this, we can decompose the motion of the SP into its rigid body motion and deformation. Since internal forces are independent of the rigid body motion when measured in the corotational reference frame, we can use only the deformation component of the displacement to predict the internal forces. This particularly helps in accounting for geometric nonlinearities induced due to the rigid body motion in the transformation matrix that relates the local (corotational) frame and the global (fixed) frame. Since the deformation component is known to be much smaller than the rigid body motion, a linear geometrical theory can then be used in the corotational frame. For implementation of the above mentioned method, we use the study by Battini [95] and extend it for a 3-D element. The displacement of the SP (or the element e in direction i) u_i^e has contributions from rigid body motion u_i^{erb} and deformation u_i^{ed}. In order to remove the rigid body motion from the total displacement component, we need to describe its translation (u_i^{et}) and rotation (u_i^{er}). The translation contribution can be easily obtained by taking the average of the displacement values at all 8 corner nodes from the bigger brick as considered for the bilinear transformation. For the rotation contribution, we need to compute the rotation matrix (R). We do this by defining the displacement field of the element e as a function of interpolation or shape functions as $U_i = \sum_j^8 u_{ij}^e N_j$ where $N_j = N(A, B, C)$ from Fig. 2.17. Using the isoparametric formulation of the displacement, we compute the displacement gradient $\left(\frac{\partial U}{\partial x}\right)$ and consequently the deformation gradient (F_c) at the center of the element that is then used to compute the rotation matrix. Equations (2.14) and (2.15) depict the procedure used.

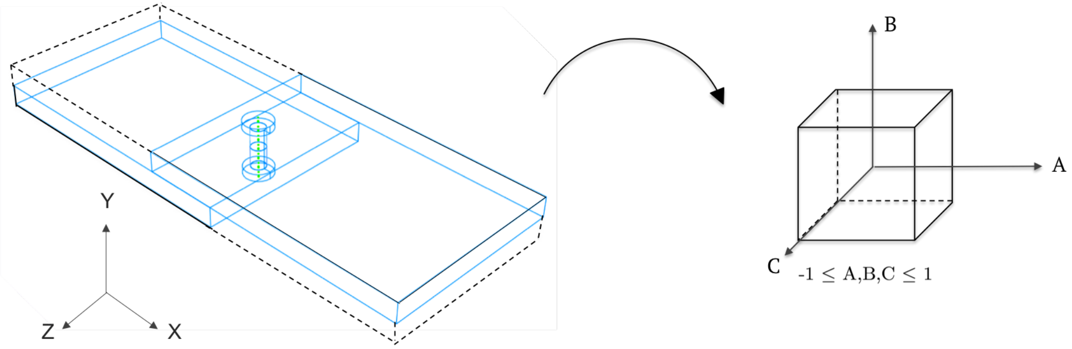


Figure 2.17: Isoparamteric mapping of the SP

$$\frac{\partial U_i}{\partial x_k} = \sum_j^8 u_{ij}^e \frac{\partial N_j}{\partial x_k}$$

$$\text{where,} \begin{bmatrix} \frac{\partial N_j}{\partial X} \\ \\ \frac{\partial N_j}{\partial Y} \\ \\ \frac{\partial N_j}{\partial Z} \end{bmatrix} = \begin{bmatrix} \frac{\partial X}{\partial A} & \frac{\partial Y}{\partial A} & \frac{\partial Z}{\partial A} \\ \\ \frac{\partial X}{\partial B} & \frac{\partial Y}{\partial B} & \frac{\partial Z}{\partial B} \\ \\ \frac{\partial X}{\partial C} & \frac{\partial Y}{\partial C} & \frac{\partial Z}{\partial C} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial N_j}{\partial A} \\ \\ \frac{\partial N_j}{\partial B} \\ \\ \frac{\partial N_j}{\partial C} \end{bmatrix}, X_i = \sum_j^8 X_{ij}^e N_j \tag{2.14}$$

$$F_c = \frac{\partial U_i}{\partial x_k} + I$$

$$U_c = \sqrt{F_c^T F} \tag{2.15}$$

$$R = F_c U_c^{-1}$$

With the computed translation displacement u_i^{et} and rotation matrix R, we extend the formulation presented by Battini [95] for 3-D displacement to compute the deformation component u_i^{ed} using Eq. (2.16).

$$u_i^{ed} = R(u_i^e - u_i^{et} + X_i^e - X_c^e) - (X_i^e - X_c^e) \tag{2.16}$$

Hence, from Eq. (2.16) we obtain all three components of the corotational displacements for all corner nodes of the brick. We then implement the bilinear interpolation method described earlier on these displacements to obtain displacements on all surface nodes to represent the boundary conditions for the quasistatic analysis. The output variables are forces at the corner nodes and the maximum equivalent plastic strain in the entire lap joint. Finally, to obtain the forces in the global reference frame, we pull back the forces from the local reference frame by multiplying them with R^T.

Neural Network Architecture and Results

The parameters of interest for this case are the internal forces at the corner nodes and the localized internal damage represented by the maximum equivalent plastic strain within the model, irrespective of its location. Hence, we design the neural network to have displacements and initial plastic strain as the input features and forces and maximum equivalent plastic strain as the output features. Note that the maximum equivalent plastic strain is used for damage evolution but is not needed for the temporal evolution of the SPs. The prescribed displacement inputs are for the eight corner nodes and all three directions. Further, we use initial plastic strain as an input feature in order to enable the model to learn the difference between a pristine case (such that no internal damage is present) and an evolved case (has non-zero localized internal damage parameter). It is important to note that these two input parameters, namely, the displacements and initial damage are sufficient to accurately predict and hence monitor the evolution of the quantities of interest as observed in the subsequent paragraphs. Similarly, the output of the model consists of 24 features for the nodal forces and 1 feature for the maximum equivalent plastic strain growth.

It is important to realize that the orders of magnitude for each of the input features and each of the output features might be considerably different. Hence, we need to scale the features to a range which is centered around zero. This is done so that the variance of all features is in the same range. If not, the feature with a higher variance dominates the training and hence the predicted output of the model. We use the *standardization technique* for scaling which replaces the values by the Z scores, thereby, redistributing the features with a mean of 0 and standard deviation of 1.

As mentioned in Sec. 2.2, we use LSTM architecture for the proposed work. We train this case with 3 LSTM layers of 200 neurons followed by a dense layer of size 25 (same as no. of output features). We train this model with 5000 samples and a validation split of 20%. Further, we use the Adam optimizer [89] which is based on an adaptive gradient descent method with mean squared error (Eq. 2.7) as the loss function. During training, we evaluate the model performance using validation loss, which is computed for inputs unseen by the model during training. It is worth mentioning that the model architecture and size of the training set play an important role in the obtained model accuracy. To study this effect, we conduct a sensitivity analysis. In this analysis, we compare the minimum or best validation loss obtained for different architectures as indicated by different colors and different number of samples in Fig. 2.18. Each case is set to be trained for a maximum of 1000 epochs. However, in order to avoid overfitting of the model, we impose an *early stopping* training condition. This condition allows us to stop training when the model performance stops improving for the validation set and only improves for the training set, by defining a patience parameter. For the purpose of all cases in this work, we use a patience of 10, which implies that the training is stopped as soon as 10 iterations pass wherein the validation loss doesn't improve and the training loss improves. If such a situation does not arise, the training continues through the entire amount of specified epochs. Hence, we obtain the minimum validation loss until the training is stopped or finished. Further, we test for architectures by not only varying the number of neurons in each layer, but also the number of layers. In Fig. 2.18 we show the minimum validation loss obtained for 100 (orange) and 200 (blue) neurons in each layer with number of layers as 1, 2, and 3. We observe that as we increase the number of samples, the loss decreases in general. However, for the case of 200 neurons, the loss seems to stabilize beyond 4000 samples. Also, increasing the number of layers, decreases the loss, thereby improving model accuracy but this change is not significant. Based on these results, we select the model that has a validation loss in the $\mathcal{O}(10^{-5})$. Note that for all cases presented in this thesis, a network architecture study has been performed to obtain an optimal network and determine a sufficient sample set size.

In general, conservation of linear and angular momentum of a component is dictated by the equilibrium of forces and moments which can be written using Eq. (2.17).

$$B_r^e(f^e) = 0 \tag{2.17}$$

where B_i^e represents the function for internal forces and moments and r denotes the number of relationships. All these relationships need to be balanced for the SP to be in equilibrium. However, due to the small errors that arise during prediction, the predicted internal forces may not satisfy these relations, leading to unbalanced forces and moments, which violate conservation principles if left unchecked. We guarantee that the SP satisfies the *conservation of linear and angular momentum* by adapting a method originally developed for smart

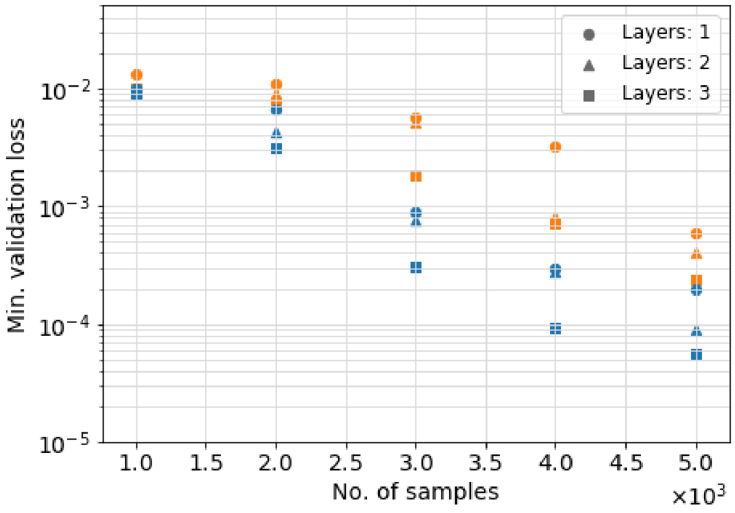


Figure 2.18: Network architecture study with minimum validation loss as a function of size of training set for different architectures

elements [73] to smart parts. Consider the forces $\bar{f}^e$ contain a prediction error $\bar{e}^e$, then Eq. (2.17) becomes,

$$B_r^e(\bar{f}^e + \bar{e}^e) = 0$$

Since, we cannot solve this system of equations directly, we assume $\bar{e}^e = \sum_i^r \alpha_i \bar{e}_i^e$, where α_i are the unknown coefficients. This may represent uniformly distributed forces or moments (pairs of forces) and helps us obtain an approximate solution. As a consequence, we reduce the error in the model and ensure the conservation of linear and angular momentum.

It is imperative to check the model accuracy with inputs which are not used for training. Results in Fig. 2.19 shows the prediction accuracy for all output parameters (forces on all 8 nodes: f_* and localized information: PEEQ$_{max}$) when validated for a test case *unknown* to the trained network. We compare the SP predictions with high fidelity FE results obtained from solving a full 3-D model. These figures indicate good agreement for the forces as well as maximum equivalent plastic strain growth. To generalize the accuracy of the model quantitatively, we use 100 unknown cases (n_s) with the developed model and compute the average RMS error ($\hat{E}_{\text{RMS}}$). RMS error can be computed using Eq. (2.8). We obtain an average error ($\hat{E}_{\text{RMS}} = \sum E_{\text{RMS}}/n_s$) of less than 2% for all the primary variables represented by the 24 output force features as seen in Fig. 2.20. Moreover, we observe that the same error metric for the plastic strain output has a value of 0.15%, which shows a remarkable agreement indicating that the model is able to resolve the evolution of the localized information accurately.

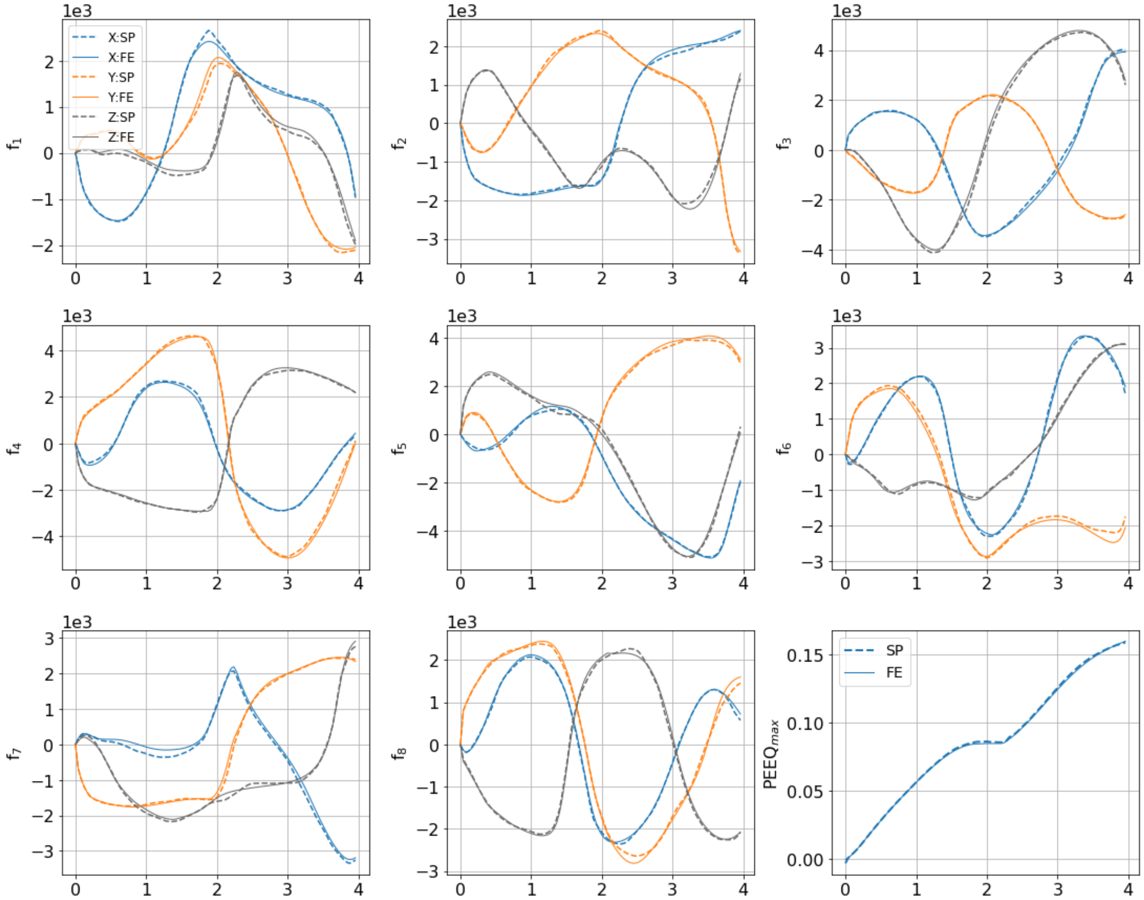


Figure 2.19: Evaluation of SP (data-driven) model by comparison of predicted output with high fidelity (FE) results

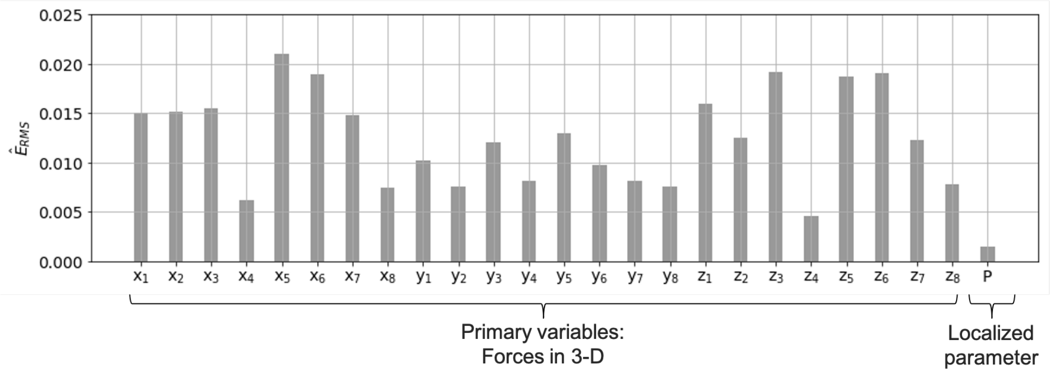


Figure 2.20: Average RMS errors for output forces at each node

2.4 Applications of Smart Parts

2.4.1 Smart Assemblies using Smart Parts

The lap joint SP developed in the previous section can be utilized in conjunction with other structural components to form assemblies. In this section, we exemplify how to use our approach to analyze an assembly composed of two lap joint SPs in series. Such a *smart assembly* can be part of a bigger component or structure and analyzed accordingly. Figure. 2.21 shows the schematics of the assembly, as well as the corresponding FEA and SP-based setups. We adopt the same lap joint design discussed earlier in this study, joining two of them together such that they share a common face. For the FE based approach, a tie constraint physically enforces compatibility such that displacements of the common surface are equal in magnitude and direction for both the lap joints. The same configuration can be used with two SPs (cell 1 and cell 2) in series. In this case, we impose boundary conditions in the form of corotational displacements on the corner nodes of the two open sides, from which we compute displacements for adjacent surfaces using bilinear interpolation as described in Sec. 2.3.2. Hence, there will be 12 nodes of interest that define the overall assembly and 36 degrees of freedom associated with these nodes. Note that the forces at the interface nodes (common side) must satisfy appropriate constraints to maintain the integrity of the assembly. Hence, we include this constraint condition in the form of force equilibrium in the prediction algorithm. For the case of two cells connected in series, the forces in the common nodes will be the sum of forces predicted in these nodes from cell 1 and cell 2. For the purpose of this work, we use an iterative procedure: the nonlinear conjugate gradient (CG) formulation (Polak–Ribière [96, 97]) to ensure force equilibrium at each time increment.

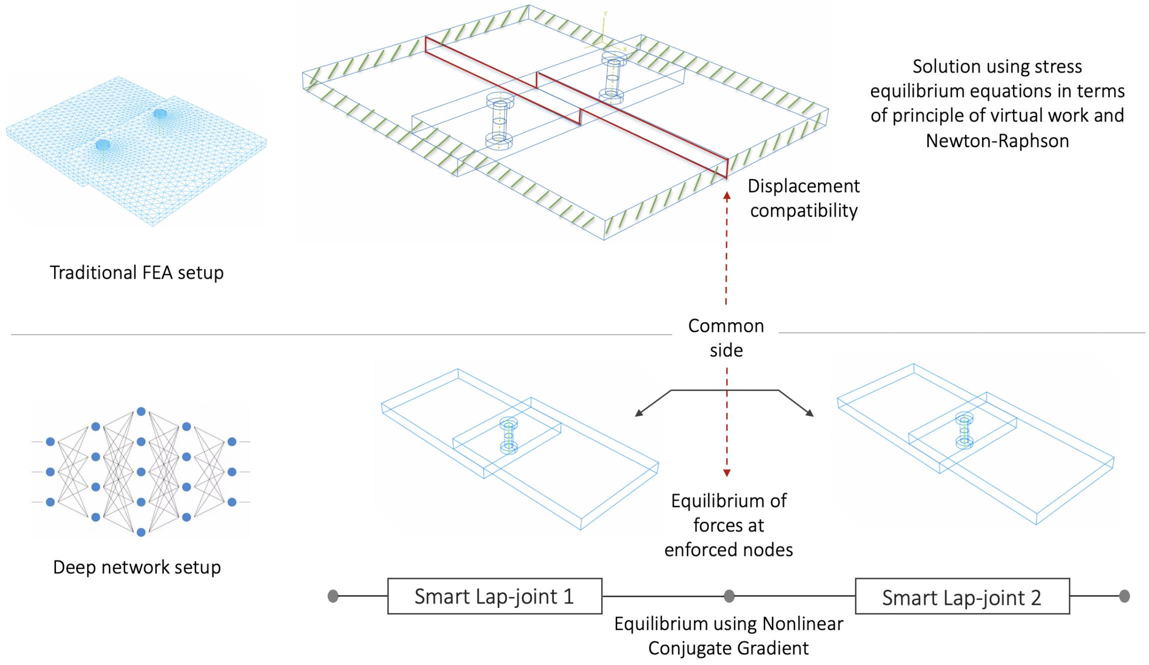

Solution using stress equilibrium equations in terms of principle of virtual work and Newton-Raphson
Displacement compatibility
Traditional FEA setup
Common side
Deep network setup
Equilibrium of forces at enforced nodes
Smart Lap-joint 1
Smart Lap-joint 2
Equilibrium using Nonlinear Conjugate Gradient

Figure 2.21: FEA (top) and SP (bottom) based representation for lap joint assembly

Note that we use the LSTM framework to ensure that the model retains history information, it is important to realize that after every prediction the LSTM internal states evolve for the next prediction. We can decouple the resetting of LSTM internal states from updates made to the model parameters of the network using the *stateful* mode. Stateful mode of LSTM thereby, allows us to obtain an output prediction at every time step without any loss of history information that is stored in the form of internal states. Hence, it is important to note that after every prediction the LSTM internal states evolve for the next prediction. Since we enforce force equilibrium for the assembly case of lap joints in an iterative manner, these LSTM internal states require *careful management* [98] during sequence prediction as explained in the following paragraph.

We explain the entire prediction procedure for the assembly lap joint using Fig. 2.22 wherein Fig. 2.22b illustrates the accurate procedure with LSTM state management. In this, we reset the internal states to the states corresponding to previous converged output during the iterative nonlinear conjugate gradient. Once the equilibrium condition is satisfied, the current states are updated to be used for the next time step prediction. Otherwise, as shown in Fig. 2.22a the states keep evolving, even within the nonlinear conjugate gradient procedure (Fig. 2.22c) that aims at finding an equilibrium solution at each time step. This eventually leads to erroneous results or divergence. Thus it is imperative that state management is necessary whenever there is an iterative procedure required within the prediction algorithm to avoid incorrect accumulation of LSTM states. To provide an analogy, a similar procedure is performed in elastic-plastic simulations at the constitutive level when an iteration is needed: that is, the effective plastic strain (state) is updated only after the iteration has converged.

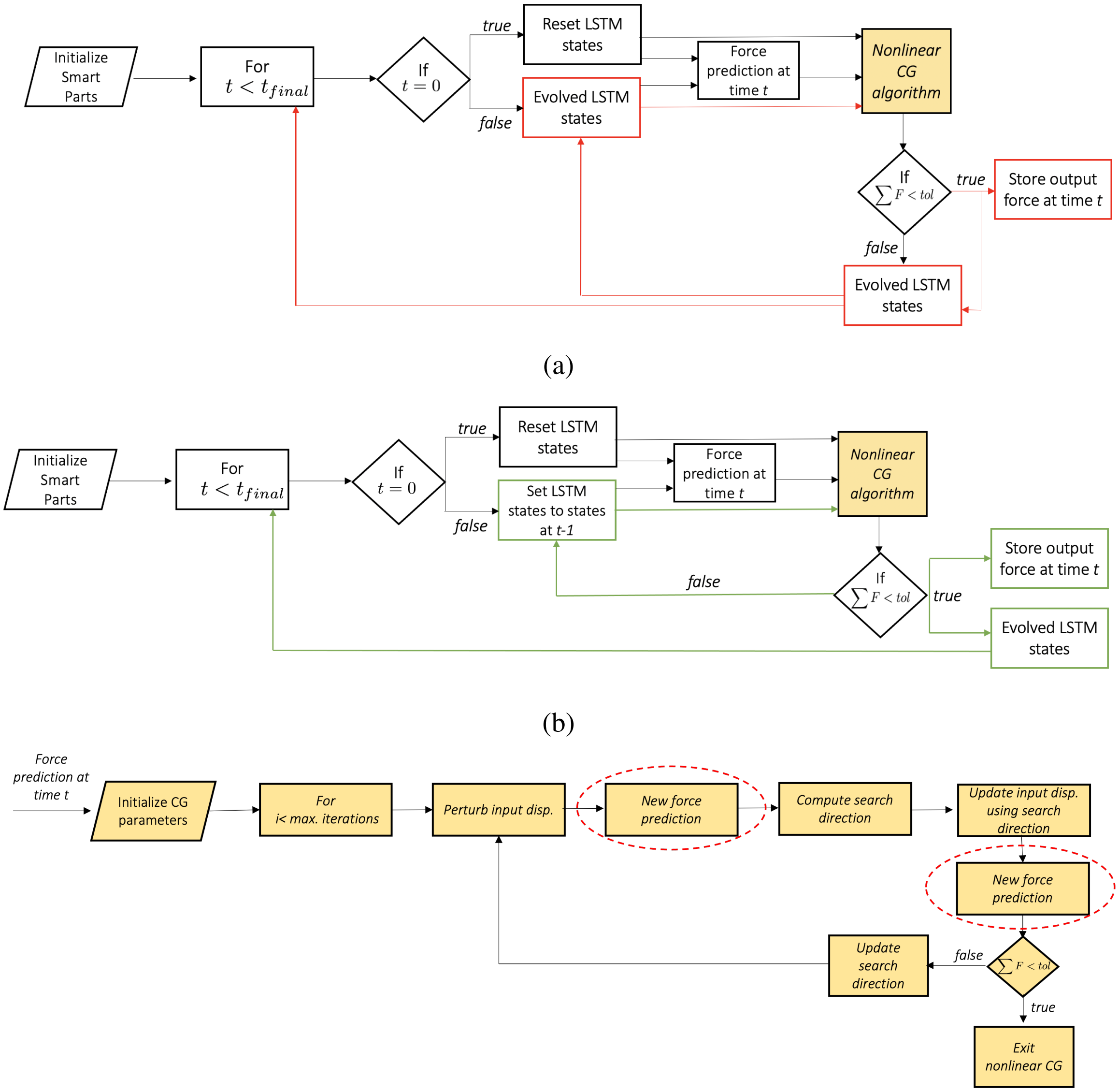
Results

Prediction using the proposed approach is obtained by implementing the trained smart lap joint in series and using nonlinear conjugate gradient in the algorithm as explained earlier. The analogous high fidelity assembly is modeled using 3-D FEA. A few test cases unknown to the trained model are simulated using both approaches to compute/predict forces and maximum equivalent plastic strain inside the domain of each lap joint. Comparison of results for one such case is shown in Fig. 2.23. Fig. 2.23a shows the comparison of predicted forces with those computed using FEA for the entire assembly.

Further, we quantify the accuracy using the the coefficient of determination, R^2 score given by:

$$R^2 = 1 - \frac{\sum_j^l (f_{aj} - f_{pj})^2}{\sum_j^l (f_{aj} - f_m)^2} \tag{2.18}$$

where f_a is the truth output value obtained from FEA, f_p is the output value predicted by



(a)

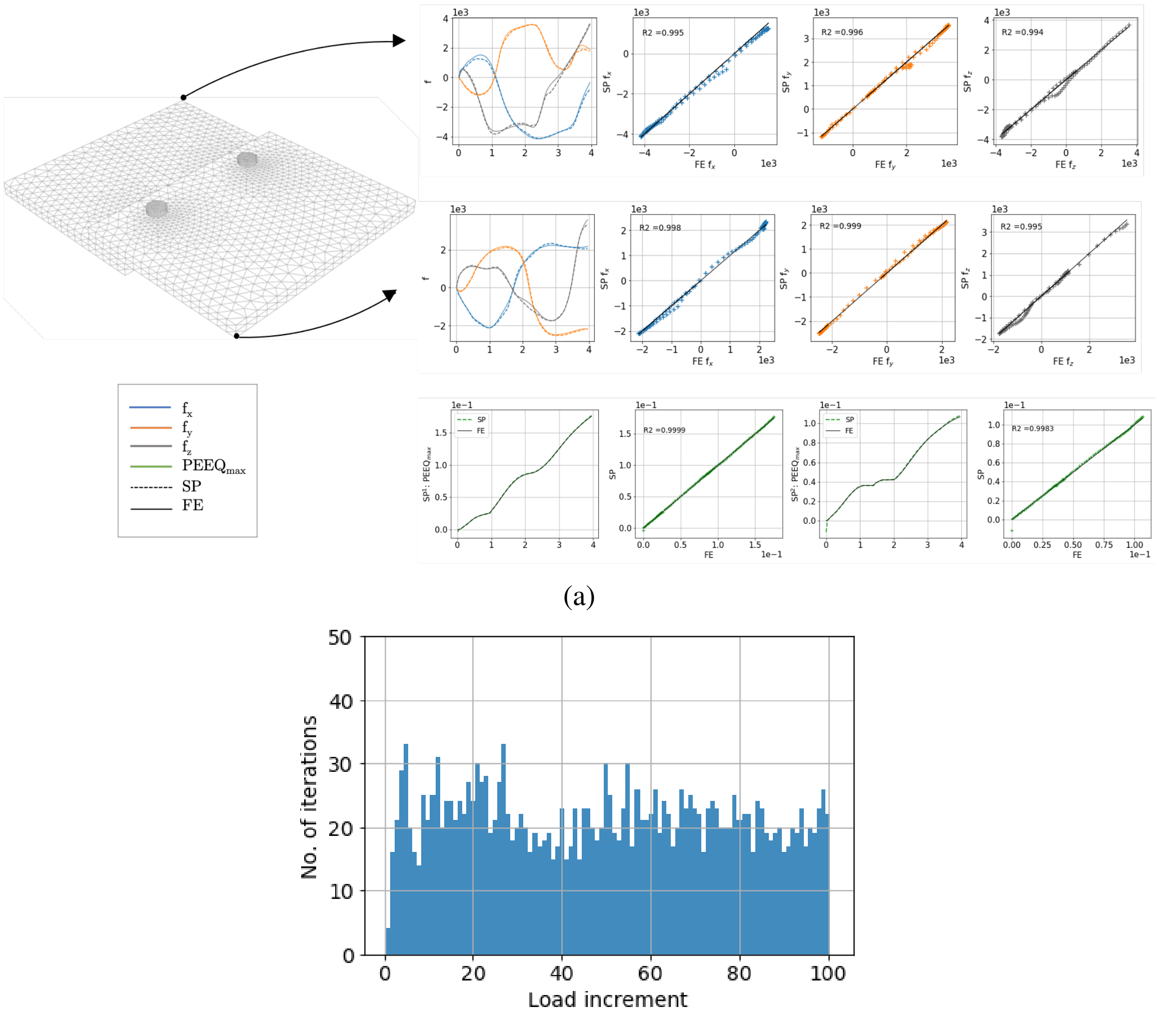
(b)

(c)

Figure 2.22: Implementation procedure for assembly of lap joints. (a) Equilibrium approach without state management, (b) Equilibrium approach with state management, and (c) Nonlinear conjugate gradient procedure

the SP based assembly, f_m is the mean of observed or truth data and, l is the length of the sequence. Hence, alongside the force comparison, we plot the predicted value (SP f$_*$) versus the truth value (FE f$_*$) of the output parameter such that each point corresponds to a predicted value over the time history of loading. A perfect correlation ($R^2 = 100\%$) is indicated by the line $f_a = f_p$. Hence, as the prediction points get close to this line, the accuracy of the model increases.

We show R^2 scores for forces in all three directions for the selected nodes. We obtain a minimum R^2 score of 99% for each of these parameters indicating very good accuracy. Remarkably, similar accuracy is also observed in maximum equivalent plastic strain for both the cells also shown in Fig. 2.23a. These plastic strains can either be used separately to further analyze the localized damage or as a measure of maximum plastic strain (damage) that exists in the entire assembly to predict failure.



(a)

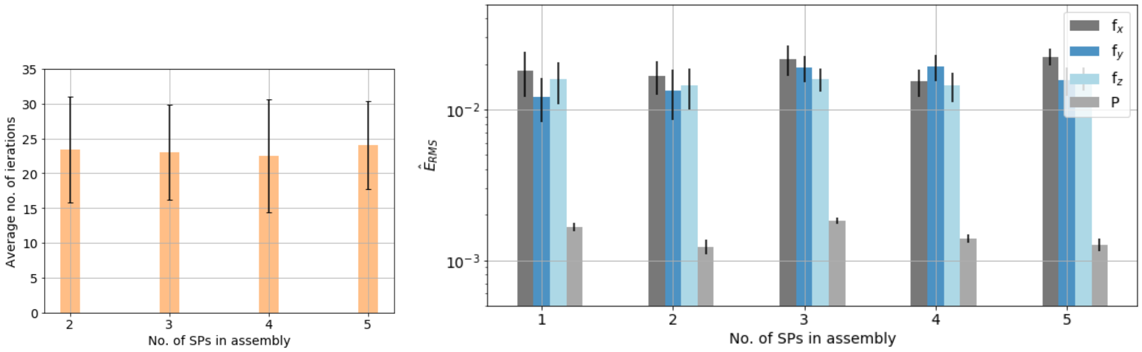
(b)

Figure 2.23: Lap joint assembly results for a test case with two SPs. (a) Lap joint assembly result comparison for various nodes along with corresponding R^2 scores, (b) Required iterations over entire deformation path

During the prediction algorithm, we also record the number of iterations that are needed at each increment for convergence of forces at the interface. This, in other words gives us an idea of the prediction quality from both the SPs that constitute the assembly and also allows us to check and identify, if there are any specific regions over the deformation path which take longer to converge. For the considered case in Fig. 2.23a, we show the required number of iterations as a function of the load increment in Fig. 2.23b. Despite the fact that the maximum allowable iterations are defined as 500 for each loading increment, we obtain a converged solution in ≈ 20 iterations. We also do not observe any abrupt changes in the iterations except at the beginning. This is attributed to the sudden increase in the feature value that occurs at the start when the deformation goes from zero to non-zero.

Parametric study for larger assemblies

We perform a parametric study to investigate the robustness of the proposed method. This helps us understand how the assembly behaves when we add more components. We conduct a study to model lap-joint assemblies with upto 5 SPs. As in the previous case with two SPs, we model these assemblies by placing SPs side by side. The boundary conditions are also imposed in a similar fashion on all open surfaces in the form of corotational displacements on the corner nodes and their bilinear interpolation. We show cumulative results for these studies in Fig. 2.24, wherein we predict the forces and localized information using 5 unknown cases for each assembly. We first analyze the required number of iterations as a function of the number of constituting SPs in Fig. 2.24a, because an increase in the number of SPs results in an increase in the number of interface surfaces and consequently the number of interface nodes. We observe no significant trend in the average number of iterations required as well as the average standard deviation (shown as the black stem in the plot). These findings also provide an insight into the computational cost for these predictions. The computational cost is associated with the number of times a prediction is made for each load increment and hence, the number of iterations. Since increasing the number of SPs in an assembly does not increase the required number of iterations for convergence, the computational cost also remains bounded. Corresponding to these cases, we show the average RMS error for different output features in Fig. 2.24b. In this plot, we also include the errors for a single lap-joint (1 SP in assembly case) as a reference for comparison. We can see that the errors are bounded as we increase the SPs in an assembly for all outputs. In fact, we also do not observe any significant increase or decrease in errors as well as dispersion for any particular force component, implying good overall representation. As a final note, it is worth mentioning that the localized information, not only has low errors but also very low standard deviation in these errors implying good accuracy across the operational domain.



(a)

(b)

Figure 2.24: Cumulative results for lap-joint assemblies with varying number of SPs. (a) Required convergence iterations, (b) Average RMS errors for output parameters

2.4.2 Localized damage driven control strategy for critical helicopter components

It is well recognized that component life extension using control schemes, such as but not limited to load alleviation and load limiting, involves a careful trade-off between vehicle maneuver performance and the impact of ensuing maneuver on a component life usage. This is more so for helicopter rotor components, such as rotating pitch links, that are routinely subjected to varying harmonic loads. In such cases, effective component life extending control schemes are predicated on the availability of effective metrics in their evaluations. Perhaps more importantly, the development of load alleviation/ limiting control schemes becomes challenging as it needs to consider the accumulation of this localized damage in its approach which is typically not accounted for in traditional fatigue analysis. It is also worthwhile to emphasize that both: fatigue analysis and localized damage accumulation need to be considered in the development of control strategies for accurate and effective component life extension.

Extensive research efforts have been made to extend component life by developing different types of control strategies. One of the most prevalent approach is the Load Alleviation Control (LAC), wherein the flight control system is modified to reduce component level peak-to-peak dynamic load and consequently reduce the peak-to-peak stress in the hope to reduce fatigue damage [99, 100]. Such modifications can take the form of either the addition of feedback signals or the optimization of flight control system gains. Despite being effective in many applications, LAC suffers from two major limitations. The first issue is in its inability to discern between aggressive versus non-aggressive maneuvers and second in its total negligence of fatigue damage induced by specific harmonic loads. To circumvent these limitations, a Load Limiting Controller (LLC) has been proposed [101]. The proposed LLC scheme limits the pilot command inputs during aggressive maneuvers using model predictive control to limit specific component level harmonic loads. Hence, a clear distinction in the aggressiveness of the maneuver is taken into account, and further, fatigue damage introduced by harmonic loads is not neglected. However, in its current form, the LLC strategy relies on an ambiguous selection of the harmonic that needs to be limited and the extent to which it is limited. Reducing arbitrary harmonic loads to arbitrary values does not provide for an optimal strategy as it can lead to significant effects on or sacrificing the maneuver performance of the vehicle, which in most cases is not desirable. Hence, it would be beneficial to limit harmonic loads based on their effect on both localized damage and maneuver performance.

A potential approach is to target and directly limit/reduce the harmonic that relates most to damage growth, and consequently, to the life of the component. To do this, there is a need for a real-time decision-making tool that can assist the life extending control schemes (such as the LLC scheme) in deciding which harmonic loads need to be limited and by how much during flight, based on their effect on both fatigue damage and maneuver performance. Currently, there is no established strategy/tool to execute this. However, in this section, we demonstrate how the pitch link smart part can enable such a less conservative control strategy to extend the component operational life.

The smart part approach is used to develop the localized damage prediction model for the pitch link as described in Sec. 2.2. It is important to note here that damage in this example refers to the localized plasticity that is predicted by the smart pitch link, which if allowed to grow in an environment with cyclic loads can become crucial for the component life. Also, the helicopter model used to generate the load in this study is different than the one in Sec. 2.3.1. With the objective of coupling the damage model with an appropriate control scheme, we use a helicopter model that uses the bare-airframe model in conjunction with a flight controller to generate the pitch link harmonic loads. The data-driven model maps these harmonic loads along with the preexisting damage information (input features) to the localized damage parameter growth (output feature) that is the maximum equivalent plastic strain growth, during training and uses all this information for future predictions as shown in Fig. 2.25. Details for training such a model have been explained in Sec. 2.3.1. As mentioned earlier, we adopt the LSTM architecture for training, which for this case uses 1 LSTM layer and 1 Dense layer with 600 neurons and a total of 5000 samples. To validate the model, we test it with samples unknown to the network. We show one such case in Fig. 2.26, wherein we see a good match between the predicted damage growth (DG) and the computed output using a full 3-D FE model (Fig. 2.26a). We also quantify the error by computing the normalized residual values over the sample's time history. Figure 2.26b shows the occurrence of the computed residual errors with a peak at 0.005% and maximum error of 0.07%, indicating good model accuracy.

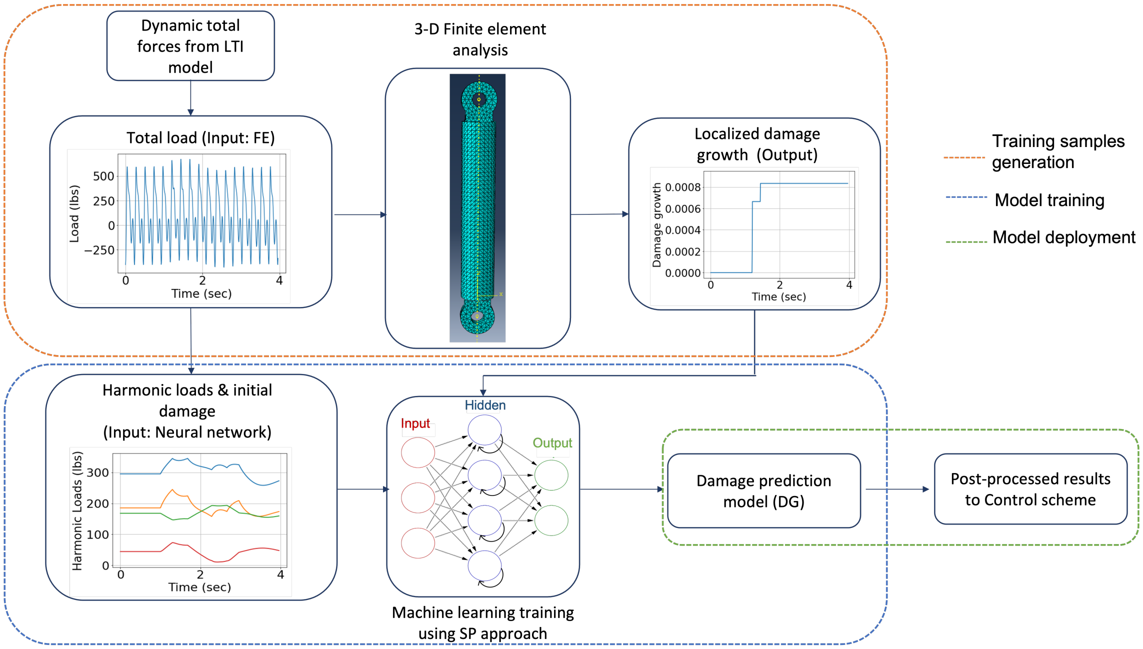
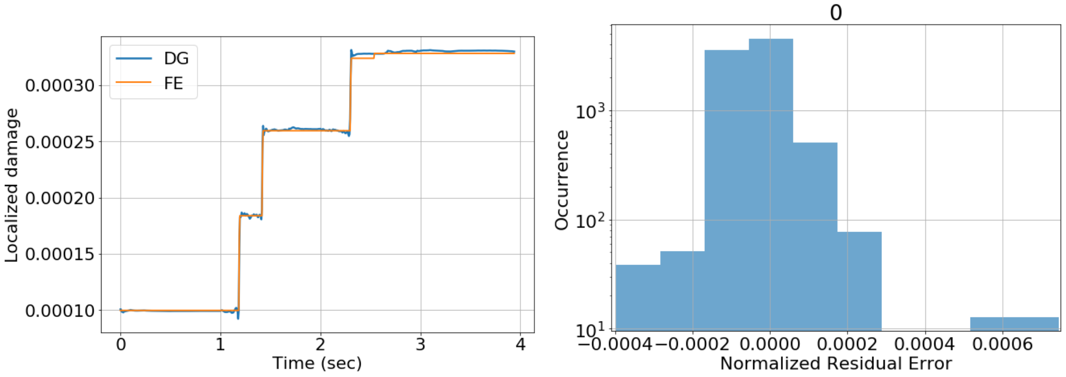

Dynamic total forces from LTI model
3-D Finite element analysis
Total load (Input: FE)
Load (lbs)
500
250
0
−250
0
2
4
Time (sec)
Localized damage growth (Output)
Damage growth
0.0008
0.0006
0.0004
0.0002
0.0000
0
2
4
Time (sec)
Training samples generation
Model training
Model deployment
Harmonic loads & initial damage (Input: Neural network)
Harmonic Loads (lbs)
300
200
100
0
0
2
4
Time (sec)
Input
Hidden
Output
Machine learning training using SP approach
Damage prediction model (DG)
Post-processed results to Control scheme

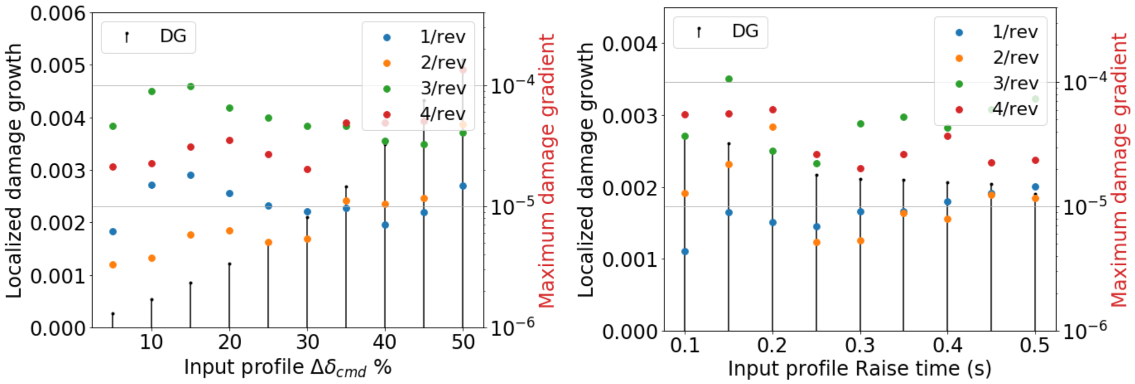
Figure 2.25: Overview of damage prediction model development and implementation

To use the localized damage parameter prediction to aid the control scheme, it is necessary to study its behavior with control input parameters. We conduct a sensitivity analysis for the longitudinal cyclic doublet maneuver on two parameters: percentage magnitude of the control input and rise time which represent the magnitude and aggressiveness of the maneuver as explained in Sec. 2.3.1. We vary the magnitude in the range of 5% and 50% to predict the corresponding damage growth by keeping the rise time at a constant value of 0.25s. Similarly, we vary the rise time from 0.1s to 0.5s for the sensitivity study. For this, we use a constant magnitude value of 30%. The results of these studies are shown in Fig. 2.27 with the end value of the predicted damage growth (DG) as the quantity of interest. We observe intuitive results, since as the magnitude of the input increases, the damage also increases, whereas when the input becomes less aggressive the damage decreases. However, the rate at which this change occurs is nonlinear in both cases, due to the nonlinear nature of the problem itself, thereby having a similar effect on operational component life.



(a) (b)

Figure 2.26: Results of damage prediction model for a sample case; (a) Comparison with high-fidelity solution, (b) Residual error distribution



(a) (b)

Figure 2.27: Sensitivity studies for damage prediction model using sensitivity parameters: (a) Percentage magnitude of command input, (b) Raise time

Moreover, since time-varying harmonic loads (1, 2, 3, and 4/rev) are used as inputs for the damage prediction model, we can study their effect on the damage parameter. We adopt this strategy from the smart helicopter pitch link in Sec. 2.3.1, to numerically compute instantaneous gradients of the predicted damage with respect to the input harmonics. We perform the same sensitivity study as described above to investigate the maximum damage-causing harmonic as shown in Fig. 2.27. Our findings are counter-intuitive as we observe that *1/rev* is not necessarily the most damage causing harmonic, despite being the dominant one in terms of harmonic magnitude. In other words, there might be situations, when the damage growth could be reduced by attacking or limiting the harmonic to which it has maximum sensitivity. This is a key aspect that the smart parts approach provides for in enabling operational component life extension without significant impact on the maneuver performance.

Load limiting Control (LLC) scheme

The LLC scheme is briefly explained in this section to enable a complete understanding of this study. Details for this model can be obtained from previous established works [101]. Such a control strategy is based on a receding horizon model predictive control formulation. Using an on-board model that provides a mapping between the pilot command input and the harmonic pitch link loads, the LLC scheme solves the optimization problem shown in Eqns. 2.19 and 2.20 in order to generate estimates of available command margin. Using these command margin estimates, the input to the controller is constrained in order to keep harmonic pitch link load within a desired maximum value. A block diagram representation of the load limiting scheme integrated within a dynamic inversion controller is shown in Fig. 2.28.

$$min_{\delta\text{cmd}}[J], J = \int_{t_0}^{t_o+T_p} L(\|Y_{\text{harm}}\|_2, \delta\text{cmd})dt \qquad (2.19)$$

$$\text{such that, } \|Y_{\text{harm}}\|_2 \leq y_{\text{max}}$$

Figure 2.29 shows a sample result wherein the LLC scheme is designed to limit the magnitude of 1/rev pitch link load [101]. We can see that the LLC scheme keeps the load within the desired user-selected maximum value of $350\ lbs$.

Integrity ratio: Damage mitigation control metric

The developed damage prediction model is capable of predicting damage growth using the harmonic loads resulting from a maneuver. These results can then be post-processed

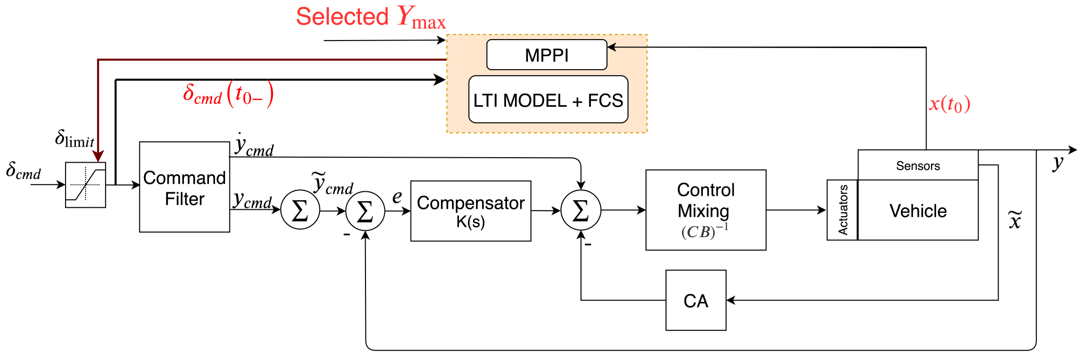


Figure 2.28: Load Limiting Control Scheme.

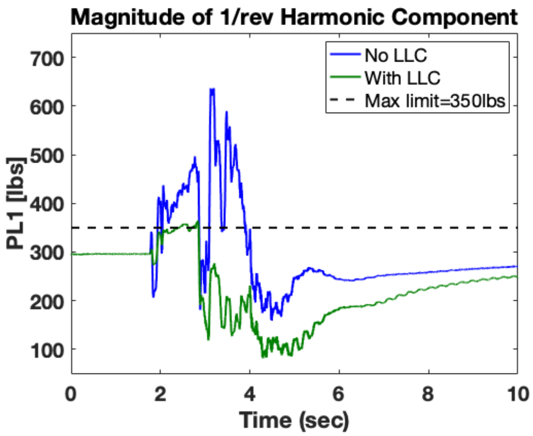


Figure 2.29: Magnitude of 1/rev pitch link load

to identify the sensitivity of damage with different harmonics as illustrated earlier in this example. Despite being immensely beneficial, it is challenging to use this information while implementing the LLC scheme. This is because the dynamics of the entire system are coupled and limiting a particular harmonic leads to significant changes in other harmonics as well, and hence the overall maneuver performance. This necessitates an approach that includes this aspect in making useful predictions about limiting a particular harmonic.

To do this, we introduce a metric that can aid in the control decision process, which we call the *integrity ratio* and is given by:

$$I = \frac{\Delta D}{\Delta P} \tag{2.20}$$

where, D and P correspond to the damage parameter and maneuver performance measure respectively. For the purpose of this study, we compute the changes in these quantities using open-loop simulations and compare the values obtained without and with LLC. Quantitatively, we represent ΔD as $\frac{D_{\text{no llc}} - D_{\text{llc}}}{D_{\text{no llc}}}$ such that D_{llc} is the damage parameter predicted by limiting a particular harmonic by a certain amount. Analogously, ΔP is given as $\frac{P_{\text{no llc}} - P_{\text{llc}}}{P_{\text{no llc}}}$ wherein, we select the maneuver performance measure (P) as the maximum pitch rate value attained. The aim is to, therefore, *maximize* this integrity ratio which includes maximizing damage reduction and minimizing the change in maneuver performance.

The integrity ratio helps us identify *which* harmonic should be limited with minimum effect on maneuver performance as shown in Fig 2.30. But, in its current form, we use an arbitrarily selected threshold value ($54\ lbs$ above trim) for this example. Also, note that while considering the $3/rev$ LLC case, the integrity ratio is computed with maximum P attained during pitch down as the limiting for this case occurs during the pitch down portion of the maneuver. On detailed investigation, we can observe that this threshold value results in a ΔD that is significantly lower for $4/rev$ as compared to $1/rev$ and $2/rev$ cases. This is attributed to the magnitude range of a particular harmonic. Hence, limiting with a threshold of $54\ lbs$ above trim significantly affects the magnitude when limiting $1/rev$ as compared to $4/rev$. This may lead to conservative estimates and requires us to use a threshold selection strategy that represents and can leverage the entire magnitude spectrum for a particular harmonic such that the effect on maneuver performance is within allowable limits. To study this further, we conduct threshold sensitivity studies for the integrity ratio as described in the subsequent paragraphs.

We perform the threshold sensitivity study in three ways by using the threshold measures (TM) shown in Eqns. 2.21-2.23: (1) varying a percentage of the trim value of each harmonic (2) varying a percentage of the maximum value of each harmonic and (3) varying a percentage of the range between the trim and maximum value of each harmonic. This way we can include a sufficiently large set of cases to understand the behavior of the integrity

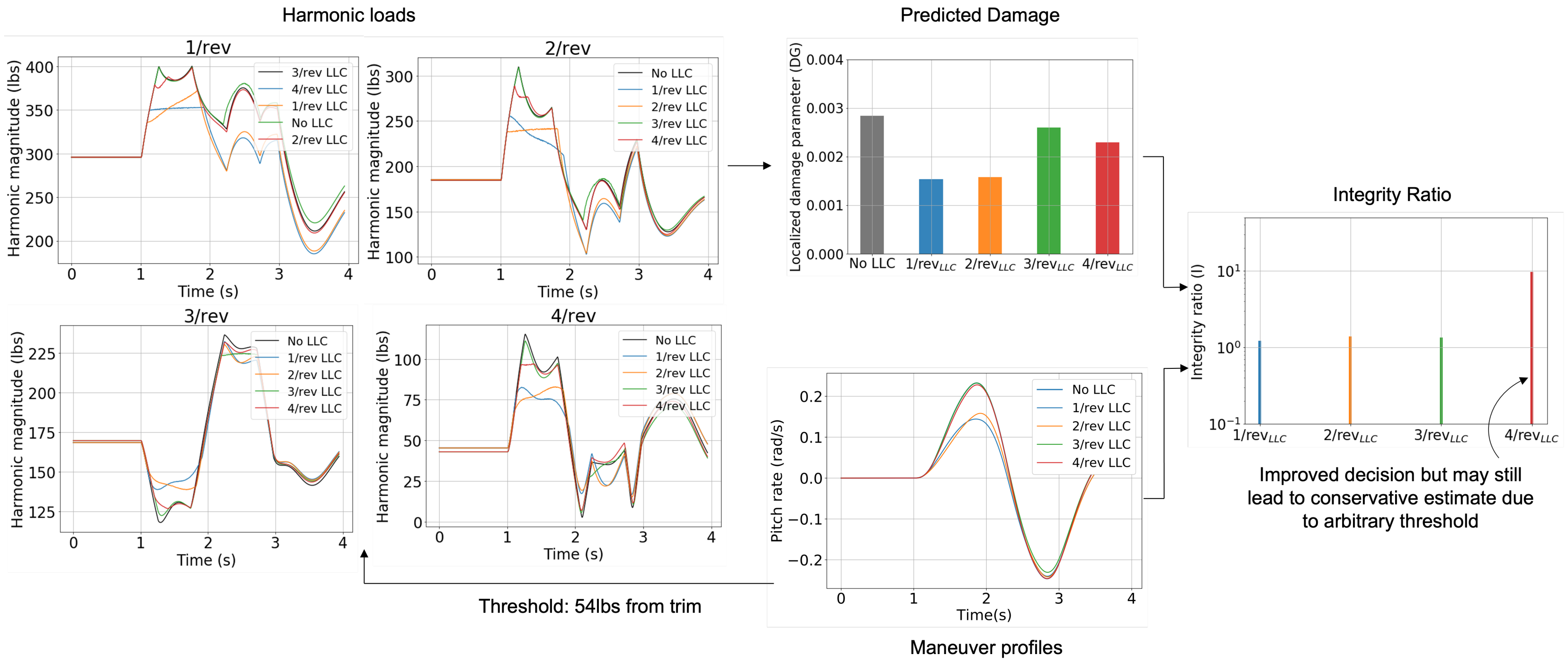


Figure 2.30: Sample result for integrity ratio with arbitrary threshold

ratio.

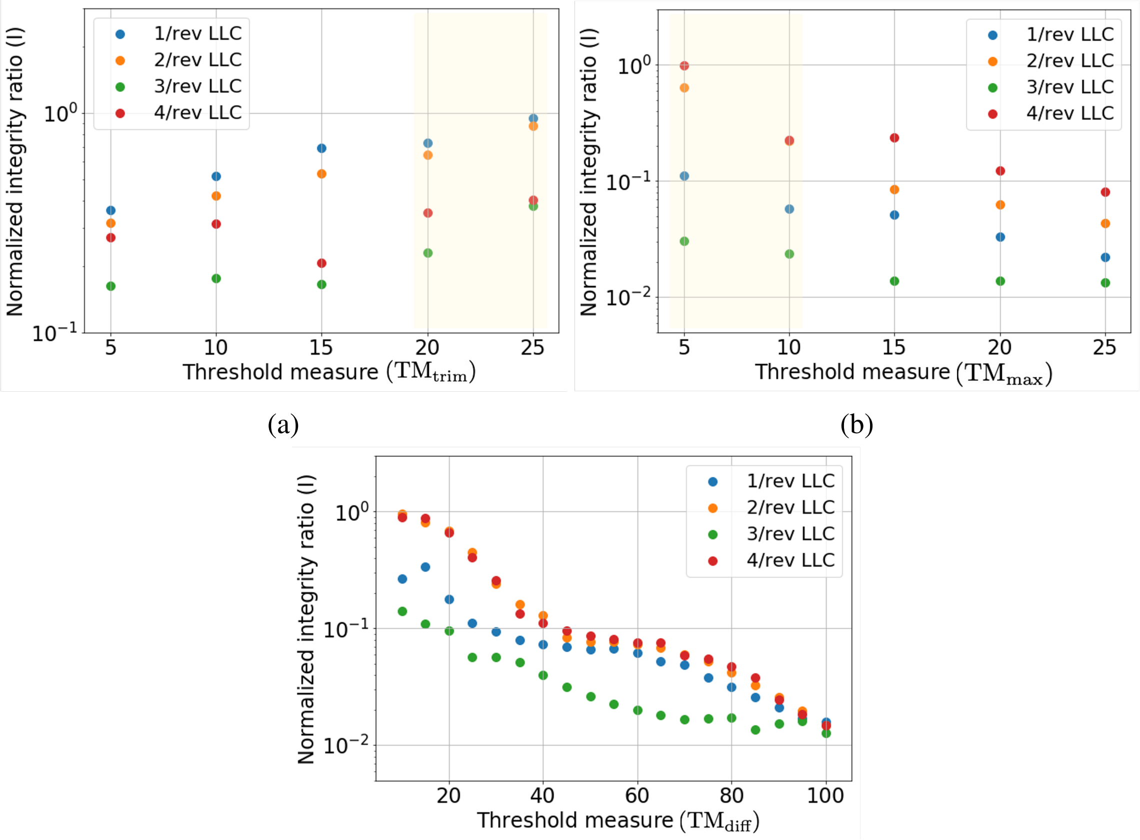
$$\text{TM}_{\text{trim}} = \text{Trim}_i + \frac{R}{100}(\text{Trim}_i) \tag{2.21}$$

$$\text{TM}_{\text{max}} = \text{Max.value}_i - \frac{R}{100}(\text{Max.value}_i) \tag{2.22}$$

$$\text{TM}_{\text{diff}} = \text{Max.value}_i - \frac{R}{100}(\text{Max.value}_i - \text{Trim}_i) \tag{2.23}$$

In Eqns. 2.21-2.23, R is the varying percentage value and i represents the particular harmonic being limited, that is $1/rev, 2/rev$, etc. The computed integrity ratios are shown in Fig. 2.31, such that within each case, the results are normalized by the maximum value of integrity ratio obtained for that case. Overall, we observe that for TM_{trim} and TM_{max}, as we increase the TM, the integrity ratio increases and decreases respectively. This implies, that as we move from the maximum value towards the trim value for the load threshold, the effect on the maneuver performance plays an important role and is reflected in the integrity ratios. Hence, we ideally would want to impose controls in the shaded portions of Figs. 2.31a and 2.31b.

Also, it is worth mentioning that a range more than $5\% - 25\%$ cannot be considered for these TMs due to different magnitude ranges for different harmonic loads. This makes these threshold measures slightly biased: TM_{trim} is biased towards reducing damage and TM_{max} towards preserving the maneuver. For the case with TM_{diff}, we use 20 equally spaced intervals between the maximum and trim value of each harmonic, thereby enabling an unbiased representation of results. Our findings indicate that even in this case (Fig. 2.31c), as we increase the TM, the integrity ratio decreases. This reduction is dramatically more after 50%, implying a significant effect on the pitch rate and hence the maneuver performance. The best-case scenario (largest integrity ratio) obtained suggests that $2/rev$ should be limited by 5% which corresponds to ΔD : 2.5% and ΔP : 0.15%. Closely following, the second-best option is to limit $4/rev$ by 5% with a ΔD : 9% and



(a)

(b)

(c)

Figure 2.31: Sensitivity studies for threshold showing integrity ratios for (a) $\mathrm{TM_{trim}}$, (b) $\mathrm{TM_{max}}$ and (c) $\mathrm{TM_{diff}}$ measures

ΔP : 0.7%. Hence, with less than 1% reduction in maneuver performance, the localized damage growth can be reduced by 10% by using our proposed strategy in harmonic and threshold selection for load limiting. Moreover, selecting $4/rev$ for limiting, implies that the maneuver can be conducted, but at a slower rate and hence might not significantly affect the handling qualities. Overall, depending on the allowable effect of maneuver performance, the best choice in terms of damage mitigation can be made for which harmonic to limit along with the percentage that needs to be limited, thereby leading to component life extension. Thus, the smart parts approach when used for a dynamic component such as the helicopter pitch link, can aid in the control decision by leveraging the information pertaining to localized damage growth as well as maneuver performance which enables a less conservative strategy to extend the operational life of the component.

2.5 Computational Cost

The main benefit entailed by our proposed approach is its capability to provide a reduced-order model to solve nonlinear structural problems in real time with accuracy comparable to that of 3-D FEM. As discussed in the different case studies we presented, the accuracy of our methodology is not sacrificed when compared to high fidelity models. Furthermore, our approach entails a key advantage when compared to high fidelity models: the associated computational cost required to produce predictions is significantly lower.

Table. 2.1 shows this comparison for the cases studied in this paper. We report the computational time as the average of that obtained over 100 simulations for each study. Results did not show significant dispersion. We can see that, depending on the case, FEM computations take between one and a half and three hours to solve the corresponding problem, whereas the proposed approach can deliver results on the order of a few seconds while ensuring very low errors in the solution and preserving the required localized information.

Table 2.1: Computational time comparison between 3-D FEA and SP approach

Case	3-D FEA (hr)	Data-driven: SP (sec)	
		Local	GPU
Lap joint	1.5	3.2	1.7
Lap joint assembly	2.1	40.2	26.6
Pitch link (without switch)	2.8	1.9	0.5
Pitch link (with switch)	2.8	1.0	0.3

It is worth noting that the comparison above was performed on different hardware due to the nature of each implementation. The finite element simulations ran on our Intel Xeon e5 2.5 GHz cluster. Each lap-joint simulation was executed on 8 cores in parallel, whereas the pitch link ones on 16 cores. On the other hand, we executed the SP simulations on two different architectures, as shown in the table. The *Local* cases corresponds to a MacBook Pro laptop with 6 computing cores. In this case, simulations took advantage of 2 of them. The *GPU* cases correspond to a GPU-based machine with an NVIDIA accelerator that allows for parallelization by using CUDA LSTM layers. As can be seen from Table. 2.1, this leads to a further reduction (at least 40%) in computational time. Overall, we can see that the SP implementation, when running either on a laptop or on a dedicated accelerator, resulted in a drastic reduction in required computational time. This could be key in enabling real-time computations in time-critical applications, for example, health monitoring of mechanical assemblies.

It is worth mentioning that there is some upfront computational cost associated with the generation of training samples that may seem substantial, however, this is a constant cost. Once trained, the benefits of the SP keep improving as the number of simulations that need to be run increases, thereby offsetting the upfront cost. For example, for the helicopter pitch link, we simulate 7000 (700 parent profiles $\times$ 10 load profiles per parent) load profiles to generate the training set. Further, each load profile has ≈ 690 timesteps, thereby resulting in a total of ≈ 4.83 million timesteps (7000 load profles $\times$ 690 timesteps per load profile) that are simulated. However, it is important to note that one load profile of ≈ 690 timesteps represents only ≈ 4sec of helicopter flying time, which is an extremely small fraction of the total flying time that a helicopter encounters over its entire operational life. Hence, the resulting number of required simulations is very large in a typical helicopter flight, which offsets the upfront cost of developing the SP.

CHAPTER III

DYNAMIC HOMOGENIZATION VIA META LEARNING

3.1 Introduction and Motivation

The accurate prediction of stress wave propagation in structures are of great importance in numerous engineering applications such as but not limited to non-destructive testing [102, 103, 104, 105] and earthquake analysis [106, 107, 108]. However, most of the unique and interesting wave propagation related phenomena occur when there are material or structural heterogeneities or when there is a significant modulation in stiffness or inertial properties across the structure. Consequently, the dynamic behavior of heterogeneous media is greatly researched in the form of phononic crystals [109, 110] and acoustic wave management [111, 112]. For such problems as well, FEM becomes computationally expensive [113]. This is because extremely fine meshes are needed to resolve small features which when considered from the perspective of a large global structure results in a large number of elements and degrees of freedom. This problem is further exacerbated if short wavelengths need to be captured which leads to the need for a very fine discretization of the temporal domain. Hence, as discussed in Chapter I, when problems become nonlinear or need to capture a broad frequency spectrum (including extremely short wavelengths), there are not many established alternatives to FEM.

In this Chapter, we address this gap and propose a novel approach for data-driven homogenization via meta-learning to describe the dynamic behavior of a unit cell and subsequently large organized arbitrary structures. In this approach, we make no assumptions about the type of material as well as the heterogeneities that may be present in the unit cell. To demonstrate this we develop a unit cell with and without heterogeneity with linear and nonlinear material models. Perhaps more interestingly, we show how these data-driven unit cells can be implemented to study and analyze wave propagation in larger arbitrary structures. We further enhance our approach by developing and illustrating an effective approach for low noise sample generation for a broad frequency spectrum.

3.2 Problem Statement

Let us consider a wave propagation problem in continuous media defined through the governing differential equation

$$\mathcal{M}[\ddot{\mathbf{u}}(x,t)] + \mathcal{K}[\mathbf{u}(x,t)] = \mathbf{f}(x,t) \tag{3.1}$$

where $\mathcal{M}$ and $\mathcal{K}$ are spatial differential operators specific to each kind of problem, $\mathbf{u}(x,t)$ is the displacement field, x is the position over the spatial domain Ω, $t \in [0,T]$ is the time and $\mathbf{f}(x,t)$ is the forcing term. Note that we denote the time derivative of a variable by using a dot over that variable. We essentially want to find the solution $\mathbf{u}(x,t)$ to the differential equation shown in Eq. 3.1 subject to certain boundary conditions over the considered domain Ω. Moreover when considering continuous media such that the domain can be represented by repeating unit cells (of one or more kind), the behavior of each unit cell can also be represented by Eq. 3.1 but with appropriate boundary conditions (forces from neighboring unit cells).

To compute the solution for a finite structure, the discretized version of Eq. 3.1 is typically obtained and used through a refined mesh of the unit cell. But, it is clear that, when performing dynamic simulations over a finite domain that consists of a large number of unit cells, the computational cost associated with these simulations makes such studies highly inefficient. This problem worsens if local features in the unit cell need to be resolved. This

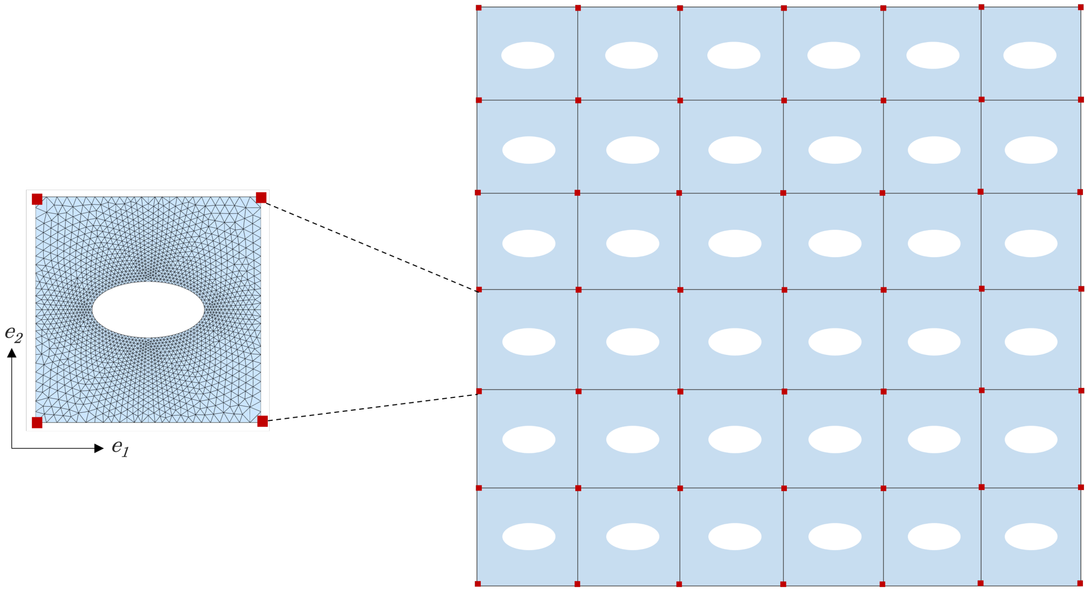


Figure 3.1: Periodic assembly with mesh driven by local heterogenity in every unit cell

is because a highly refined local mesh is needed which in turn leads to models with a large number of degrees of freedom for large assemblies of unit cells. We illustrate this using a fine discretization of the unit cell in Fig. 3.1 along with a finite structure with multiple such unit cells, potentially leading to a highly refined mesh over the entire domain and correspondingly a very large model that needs to be solved. Note that in the remainder of this Chapter, a finite structure is denoted using the number of unit cells that repeat in each direction (e_1 and e_2 defined in Fig. 3.1). So, a finite structure of size $m \times n$ denotes, m unit cells in e_2 direction and n in e_1 direction. Also, within a finite structure, a particular unit cell is denoted using $\mathrm{C}_{p,q}$, such that $p \in [1, m]$ and $q \in [1, n]$.

There are numerous reduced order modeling or homogenization alternatives to FEM which have been established to overcome computational inefficiency such as static condensation, spectral methods to name a few. However, as mentioned in the introduction, it is very challenging to find an efficient and accurate method to homogenize the unit cell (spatial domain) which is capable to operate for a broad frequency range or for nonlinear material response. In the following section, we introduce our approach to homogenize the unit cell by developing a data-driven surrogate which can then be used to study the dynamic behavior of arbitrary large structures.

3.3 Homogenizing via surrogate modeling of unit cell

Let us consider the problem of homogenizing the dynamic behavior expressed in Eq. 3.1 for the unit cell illustrated in the previous section. To model such a periodic structure, we propose an approach that, given a sufficient number of numerical experiments that represent the unit cell's global dynamic behavior, we can make a machine learning algorithm learn this information and use it to model larger finite structures. Hence, this approach, even though is demonstrated for a 2-D unit cell, is not limited to it and can be used to model periodic structures which are discrete, continuous, and/or include nonlinearity of some form, as exemplified later in Sec. 3.4. Since we are interested in modeling the dynamic behavior, we need to consider sequences representing time-domain data for all input (displacements) - output (forces) parameters. Our approach, hence, adopts ML techniques that enable the unit cell to learn such sequential data or mechanical behavior and uses information pertaining to its previous time-step to predict output parameters for the current time step.

More specifically, the surrogate modeling of a unit cell involves two major steps as shown in Fig. 3.2 such that the first step involves *training set generation*. In this, we generate various displacement sequences ($\mathbf{u}_*$), which the unit cell will see as inputs. Then, we develop a finite element model to represent the unit cell in a periodic structure and compute the nodal forces ($\mathbf{f}_*$) in time domain from the corresponding input displacements. While generating training samples, we simulate a single cell: $1c$ (1×1) case and a $3c$ (3×3) finite structure case. This is done so that, during training, the model can learn the important difference between a start/end cell versus a cell that may be in between two cells. Following this, in the second step, we utilize the training samples set and *train* the selected neural network architecture. Finally, we evaluate the model performance by implementing it with displacement inputs unknown to the model for a finite periodic structure having multiple unit cells. It is important to note that this is the key benefit of our approach since traditional methods based on analyzing a unit cell cannot study the dynamic behavior of finite periodic structures. We explain each of these steps in a detailed manner in the remaining portion of this section using the unit cell shown in Fig. 3.1.

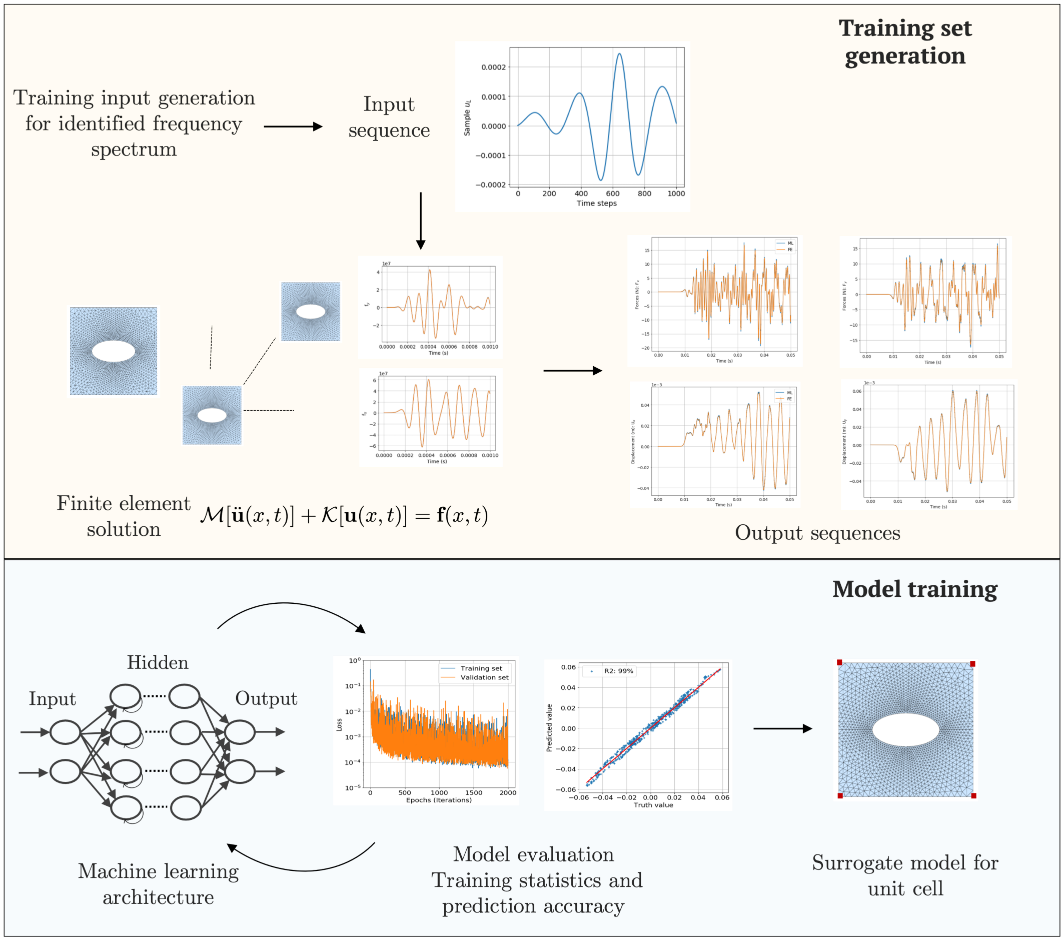
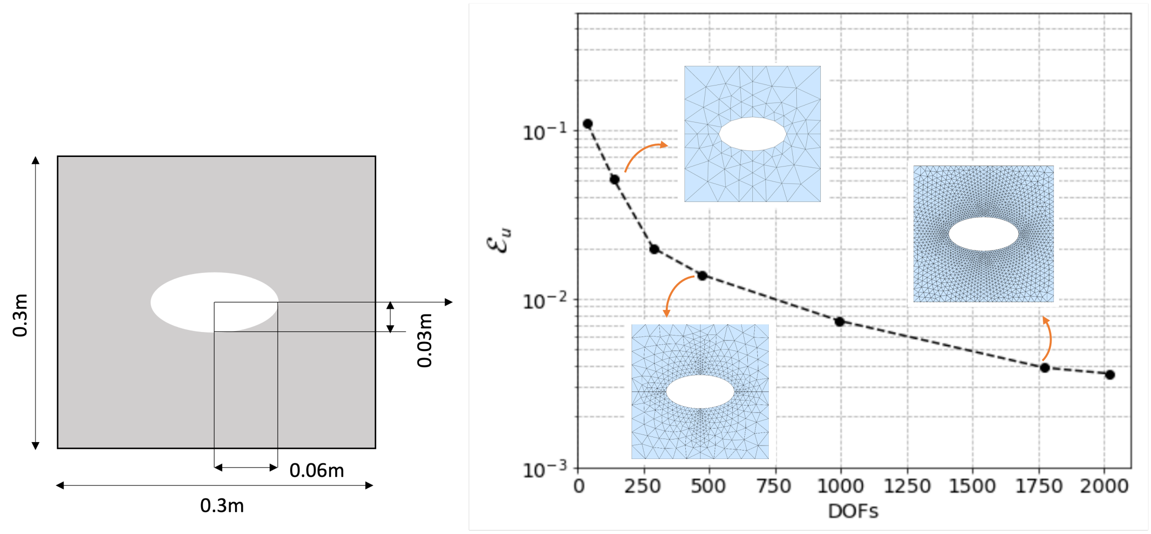

Training set generation
Training input generation for identified frequency spectrum
Input sequence
Sample u_i
Time steps
Finite element solution
$\mathcal{M}[\ddot{\mathbf{u}}(x,t)] + \mathcal{K}[\mathbf{u}(x,t)] = \mathbf{f}(x,t)$
Output sequences
ML
FE
Time (s)
Model training
Input
Hidden
Output
Machine learning architecture
Training set
Validation set
Loss
Epochs (iterations)
R2: 99%
Predicted value
Truth value
Model evaluation Training statistics and prediction accuracy
Surrogate model for unit cell

Figure 3.2: Methodology for developing surrogate model for a unit cell

3.3.1 Training sample generation

For illustration purposes we first assume linear elastic behavior of the unit cell in this section and extend the approach for nonlinear materials in Sec. 3.4.3. The geometric definition of the unit cell with a heterogenity is shown in Fig. 3.3a and the material considered is Aluminum with a modulus of elasticity (E) of 69GPa, Poissons ratio (ν) of 0.33, and density (ρ) of 2700kg/m^3. Further, displacement is prescribed as an input at the left end global nodes ($\mathbf{u}_L$) of the unit cell. The displacement at the right end ($\mathbf{u}_R$) and subsequent nodes are derived quantities from the forces generated due to $\mathbf{u}_L$.



(a) (b)

Figure 3.3: Development of unit cell (a) Unit cell design, (b) Mesh convergence

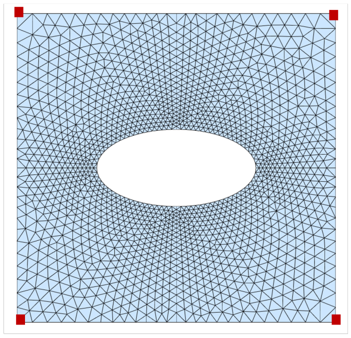


Figure 3.4: Finalized mesh for unit cell

We conduct numerical analyses using different local discretizations by increasing the number of elements used to resolve the local features. For each discretization we measure the error norm of the maximum displacement obtained at the right end of the unit cell as shown in Fig. 3.3b. We compute the error norm given by,

$$\mathcal{E}_{\mathcal{U}} = \frac{|\mathcal{U}_f - \mathcal{U}_n|}{|\mathcal{U}_f|} \tag{3.2}$$

in which $\mathcal{U}_f$ denotes the numerical maximum displacement computed using an extremely fine finite element mesh with 1841546 elements and 923379 nodes, and $\mathcal{U}_n$ is the maximum displacement obtained for various cases with different meshes. This comparison helps us finalize the local mesh needed to ensure high-fidelity results. The converged (finalized) mesh for the considered unit cell as shown in Fig. 3.4 has 1773 nodes and resulted in $\mathcal{E}_{\mathcal{U}} = 0.3\%$. It is important to note that this mesh is used to generate the training samples. However, during training, displacement and forces at the global nodes (red nodes) are used to represent the global behavior of the unit cell. This, thereby homogenizes the unit cell behavior by reducing the degrees of freedom while retaining sub-element characteristics during the analysis of large structures.

In general, the accuracy and reliability of ML-driven surrogate models rely heavily on the underlying data using which they were trained. Consequently, it is essential to identify the relevant space of interest for a particular problem to ensure that the input spans that space. For this work, the input parameter represents the prescribed displacement at the left end of the first cell. Hence, this displacement should be such that it spans over all the possible excitation scenarios (different excitation frequencies) for the considered periodic structure. For the considered unit cell, we use a frequency range of 10kHz to 50kHz. This frequency range includes very high frequencies, corresponding to very short wavelengths. It is important to emphasize here that for this unit cell the maximum frequency that can be resolved using static condensation [19] is 14.5 kHz; thus, the considered frequency range for training samples is 3 times this frequency.

One possible approach to generate training inputs is to use harmonic inputs such as sinusoidal or superposition of sinusoidal signals. However, considering that ML methods interpolate solutions for any signal seen outside the training sample, such an approach defeats the purpose of developing a general unit cell capable of systematically representing any type of input signal. Another approach prevalent in the literature is the use of Gaussian processes to generate smooth and continuous sequences for structural problems [94, 114].

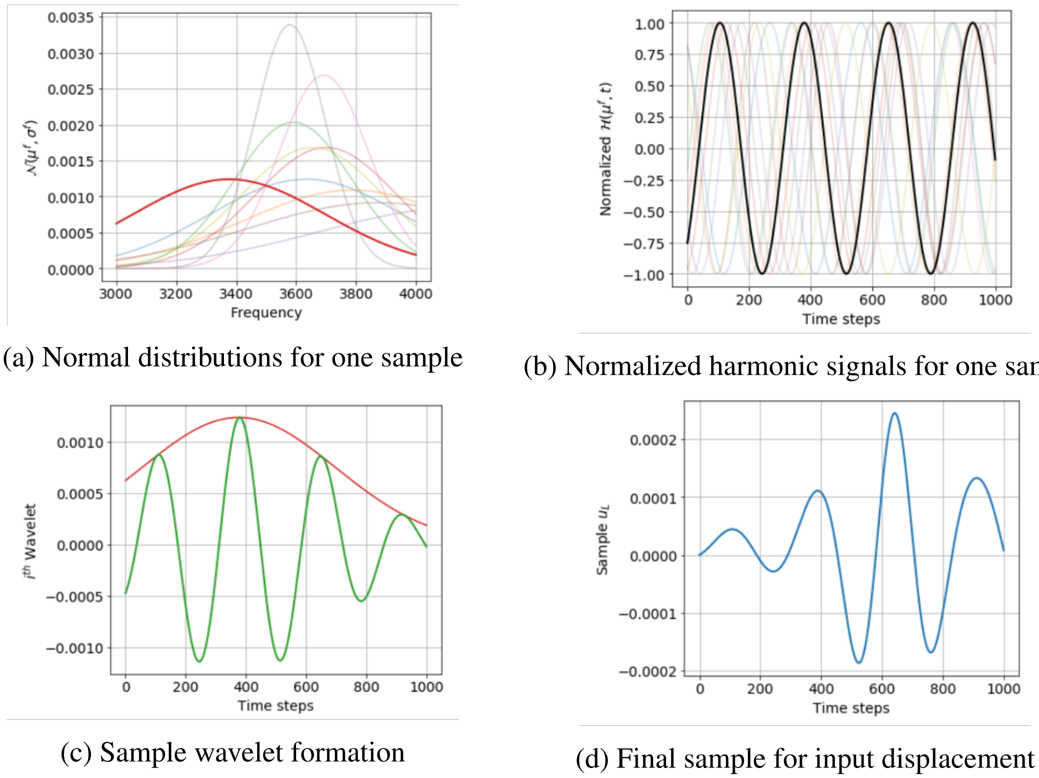
Though extremely powerful, this method suffers from large noise to signal ratio when considering a wide frequency spectrum, causing issues in generating samples that have high frequencies with the dominant magnitude.

To overcome these problems, we propose a method that is general in a way that, it is independent of the frequency spectrum that needs to be considered. Our solution makes use of wavelets [115] which we form by using a Gaussian distribution and a systematically randomized harmonic signal. A superposition of multiple such wavelets generates one input signal for the surrogate model, which is mathematically given by:

$$\mathbf{u}_L(t) = \sum_i \mathcal{N}(\mu_i^f, \sigma_i^f)\mathcal{H}(\mu_i^f, t) \tag{3.3}$$

where, $\mathcal{N}(\mu_i^f, \sigma_i^f)$ is a normal distribution function with mean frequency, μ_i^f and standard deviation, σ_i^f and $\mathcal{H}(\mu_i^f, t)$ is the harmonic signal contribution expressed as $A_i cos(2\pi\, \mu_i^f t - \phi_i)$. The mean and standard deviation in $\mathcal{N}$ for each wavelet is extracted from a uniform distribution defined for the frequency range under consideration. Note that, this frequency range is user-defined and consequently gives control over the noise to signal ratio. In other words, if the frequency range is high, there is more probability of having a diverse set of frequencies in the final sequence, whereas an extremely narrow frequency range results in a displacement signal with only one harmonic. Hence, depending on the type of structure being studied, an appropriate range can be selected. We also introduce randomized phase (ϕ_i) and amplitude (A_i) in the harmonic contribution to the final sequence, ranges for which can be selected based on the system parameters. Finally, we ensure that all input samples start from an initial rest condition to depict a realistic scenario. Overall, this method serves to be randomized yet systematic, enabling a good and unbiased representation of sample sequences.

In Fig. 3.5, we show the steps to generate one displacement sample. We first generate multiple (10 in this case) sequences that represent normal distributions followed by harmonic signals for these cases as shown in Fig. 3.5a and 3.5b respectively. Note that, the harmonic signals shown are normalized by the their corresponding maximum amplitude for good visualization. Then, we utilize these two sequences to generate the wavelet as shown in Fig. 3.5c. Finally, we can see the result of 10 wavelets that form a single sample to represent the prescribed displacement at the front/left node in Fig. 3.5d.



(a) Normal distributions for one sample

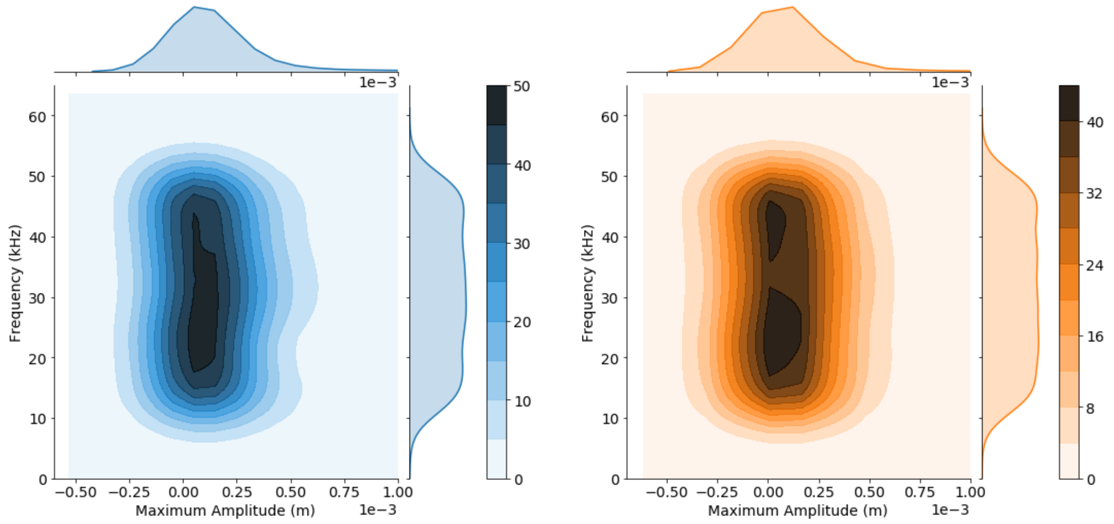
(b) Normalized harmonic signals for one sample

(c) Sample wavelet formation

(d) Final sample for input displacement

Figure 3.5: Schematic representation of method to generate sample input

For this work, we generate 2000 input displacement samples for each degree of freedom (displacement in e_1 and e_2 direction) and each global node at which displacement is prescribed (2 corner nodes at left end of the unit cell). This results in a total of 4000 samples for each degree of freedom. We show the distribution of all samples combined in Fig. 3.6. This plot helps in estimating the univariate distribution in terms of frequency and amplitude as shown in the margins (right and top) along with their bivariate distribution (center) using a kernel density estimate [90]. For this plot, we obtain the frequency data by computing the Fast Fourier Transform (FFT) of the sample and selecting the frequency which has the dominant magnitude. Analogously, for the amplitude, we compute the maximum absolute amplitude of that sample in time domain. We can observe that a fairly uniform distribution is obtained for both variables in their respective defined ranges. This exercise is important as it serves as a way to validate if the generated samples represent the interested space for a study in an appropriate manner. Otherwise a biased sample distribution may lead to a biased model during training and its subsequent application.



(a)

(b)

Figure 3.6: Sample distributions in terms of dominant frequency and maximum amplitude for prescribed displacements in (a) e_1 direction and (b) e_2 direction

In order to generate the complete training set, we conduct FE analysis for the unit cell by applying the input displacements obtained using Eqn. 3.3 on the left end of the unit cell. Further, it is important for the model to learn the difference between a cell which may be at the boundary (right or left) of a finite structure and a cell that may be in between other cells. To ensure this, we conduct FE analysis for a finite structure with a total of 9 cells with a 3×3 configuration. Note that this *serves as a minimum requirement in order to account for different unit cell behaviors* for the considered case of 2-D periodicity. As a result, for each input sequence $\mathbf{x}(t)$, we obtain two (one for single cell $(1c : 1 \times 1)$ and one for three cell $(3c : 3 \times 3)$ analysis) corresponding output sequences $\mathbf{y}(t)$ which can be represented by:

$$\begin{aligned}\mathcal{S}^{1c} &= (\mathbf{x}_j(t), \mathbf{y}_j^{1c}(t)) \\ \mathcal{S}^{3c} &= (\mathbf{x}_j(t), \mathbf{y}_j^{3c}(t)) \text{ for } j = 1, 2, 3, ...n\end{aligned} \tag{3.4}$$

such that $\mathcal{S}$ denotes the training set with n observations. Also, $\mathbf{x}(t) = \mathbf{u}_L(t)$ denotes the input displacement at the left end and,

$$\begin{aligned}\mathbf{y}^{1c}(t) &= (\mathbf{u}_R(t), \mathbf{f}(t)) \\ \mathbf{y}^{3c}(t) &= (\mathbf{u}_R^{mn}(t), \mathbf{f}^{mn}(t)) \text{ with } m = 1, 2, 3 \text{ and } n = 1, 2, 3\end{aligned}$$

represents all the outputs obtained from the FE analysis wherein $\mathbf{f}(t) = (\mathrm{f}_L(t), \mathrm{f}_R(t))$ are the forces at global nodes and mn denotes a particular cell.

Further, the right end displacement (u_R) that is obtained as an output from the FE analysis is used as an input feature along with u_L during the training process, details for which as explained in Sec. 3.3.2. In summary, for s input samples, we obtain $10s$ total observations that represent displacement-force relations. For the considered unit cell, we generated 2000 samples for $\mathbf{u}_L$ as discussed earlier, resulting in a total of 20000 observations in the overall ($\mathcal{S}^*$ = $\mathcal{S}^{1c} \cup \mathcal{S}^{3c}$) training set.

3.3.2 Novel training process using meta learning

Through the training sample generation process, we obtain a set of observations that relate the input displacement at the left end to the displacements at all subsequent nodes and nodal forces of the periodic structure. At this point, nodes only indicate the global or corner nodes of the unit cell whose behavior includes information pertaining to all the internal features of the unit cell. So for the 2-D unit cell considered, the four global nodes represent the unit cell, and all the dynamic behavior related to the internal features such as the holes is learnt by the machine learning model through this surrogate representation. To train the surrogate model, we wish to establish a relationship between the nodal displacements and nodal forces. So, we use $\mathbf{u}(t) = (\mathrm{u}_L(t), \mathrm{u}_R(t))$ as the input features and $\mathbf{f}(t)$ as the output features for the machine learning model. We can observe that such a training set-up can be easily applied to different types of unit cells by simply changing the number of input and output features.

To include any kind of history dependence (as needed for dynamic problems), it is necessary to select a machine learning algorithm that can retain information of the previous state. We utilize recurrent neural networks to do so, and specifically its long short-term memory (LSTM) architecture [80, 114] as explained in Chapter. II. Since this architecture have special memory cells characterized by internal states and forget gates, the output from each neuron gets modulated by these internal states allowing the model to learn/retain relevant information. We make use of multiple such neurons and layers of neurons to accurately establish the complex mapping between the defined input-output parameters. In addition, we also utilize Dense layers [116] (in addition to the output Dense layer) with appropriate activation functions to improve the representation capacity of the overall network. These layers increase the trainable network parameters which focus on mathematical transformation (rotation, scaling, translation, etc.) of the vector to improve point to point accuracy. Together, the LSTM and Dense layers with *relu* activation enable learning of history and/or nonlinear information without sacrificing local accuracy.

During training, the defined network architecture maps the displacements to forces using the network's internal parameters (IPs) namely, weights and biases. The training process thereby, entails an optimization of these IPs until a reasonably good solution is achieved. The goodness of the solution at every iteration (epoch) during training is given by a loss function which acts as an error metric between the actual/observed and predicted output and essentially drives the optimization process. We use the Adam [89] optimizer to update the IPs along with the mean squared error (E_{MSE}) as the loss function given by:

$$E_{\text{MSE}} = \frac{1}{s}\sum_{j}^{s}(\mathbf{f}_{j}^{o} - \mathbf{f}_{j}^{p})^{2} \tag{3.5}$$

where $\mathbf{f}_{j}^{o}$ and $\mathbf{f}_{j}^{p}$ represent the observed output and the predicted output by the network respectively such that $j \in \mathcal{S}^{*}$ and s is the number of samples used for training.

The above approach is extremely useful, but may suffer from compromised local accuracy when considering a wide spectrum of frequency range. This especially happens while analyzing periodic structures for extremely low frequency inputs. To address this issue, we propose a physics-inspired meta learning: *dual-step training* approach that uses transfer learning [117] and a custom loss function which we define to include the *effective tangent stiffness* information during the optimization process. This custom loss function, which we call the tangent loss is given in Eqn. 3.6, where $D(\mathbf{z}(t)) = \mathbf{z}_{t+1} - \mathbf{z}_{t}$ is a mathematical operation applied on the input and output feature sequences and n is the number of training samples.

$$E_{\text{TMSE}} = \frac{1}{s}\sum_{j}^{s}\frac{(D\mathbf{f}_{j}^{o} - D\mathbf{f}_{j}^{p})^{2}}{(D\mathbf{u}_{j})^{2}} \tag{3.6}$$

We illustrate this approach in Fig. 3.7 wherein the left side shows the first or the *source* step and the right side is the second or *target* step. Typically, *transfer learning* is employed for repurposing the knowledge learned in the source model to learn a new task (different data and/or network architecture) that falls within a similar domain. This is not the case in our approach. We use the concept of transfer learning in the sense that the knowledge learned during source training is used as a starting point in target training. However, the network architecture and the training data remains unchanged through both the steps. More specifically, two modifications are made in the target training: (1) the loss metric which drives the optimization process is changed from MSE loss to a custom loss (E_c) that includes MSE loss and tangent loss given by:

$$E_c = \alpha(E_{\text{MSE}}) + (1 - \alpha)(E_{\text{TMSE}}) \tag{3.7}$$

where α is a scaling parameter used to ensure that both the metrics have comparable orders of magnitude, and (2) only the dense layer IPs are kept available to the optimization for fine-tuning. In other words, all the information learnt by the LSTM layer/s during source training is unchanged during target training. Hence, the tangent loss is used to help the dense layers represent the mathematical transformations in a better fashion. We show the benefit of using the dual step training framework by comparing the improvement in the loss function obtained during training in Fig. 3.8. We can observe that when using a 1-step approach, the loss function saturates after 500 epochs whereas for the 2-step (dual) approach the loss function keeps decreasing. This in-turn results in a decrease in the prediction errors and an increased model accuracy.

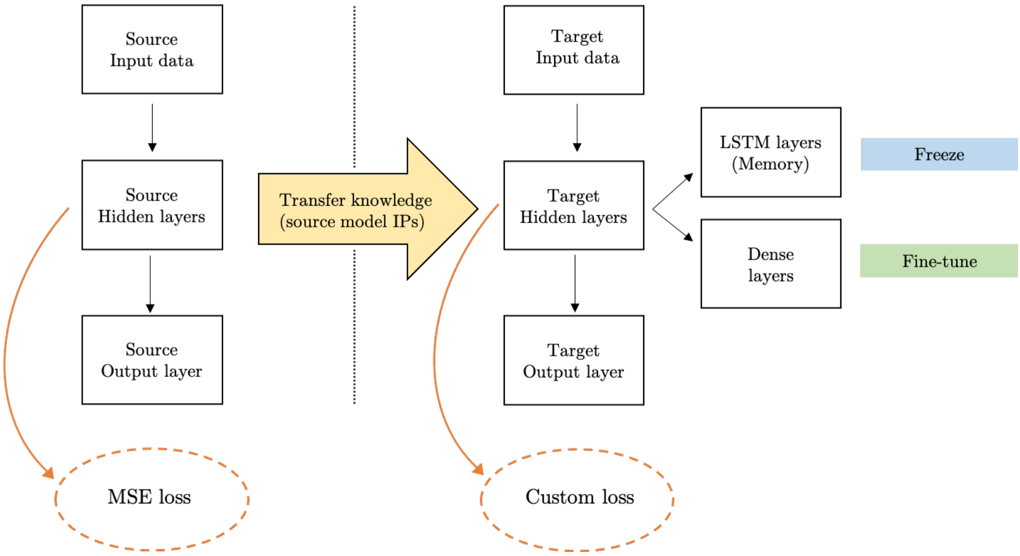


Figure 3.7: Dual-step training framework

It is important to mention that, the overall sample set is split into a training and a test set. During training the predictions for inputs in the test set (monitored as validation loss)

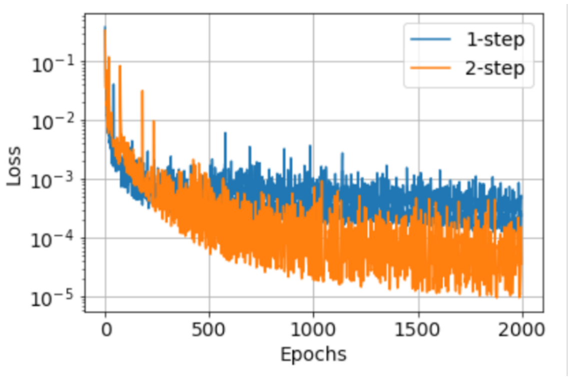


Figure 3.8: Loss function comparison for 1-step and 2-step training approach

are used to validate the current state of the network. The proportion of this split is typically 10-20% depending on the size and distribution of the overall set. We use a 20% split for this work, implying that 80% samples are used for training and 20% for validation. Note that, we also perform feature scaling on the sample data before training to allow a smooth and relatively fast gradient descent based optimization used by the neural network. To do so, we utilize the standardization technique where within each feature (input and output), values are normalized by their corresponding standard deviation as explained in Sec. 2.3.2.

Once trained, the model performance is evaluated using the validation loss, as this loss is computed for inputs unseen by the model during training. It is worth mentioning that the model architecture also plays an important role in the obtained model accuracy. To study the effect of model architecture, we conduct a sensitivity analysis. In this analysis, we compare the minimum or best validation loss obtained for different architectures as shown in Fig. 3.9 for the considered heterogeneous unit cell over both training steps. Each model architecture is trained for 1000 epochs in the source step followed by another 2000 epochs in the target step. However, in order to avoid overfitting of the model, we impose an early stopping training condition. This condition allows us to stop training when the model performance stops improving for the validation set and improves for the training set, by defining a patience parameter. For the purpose of all cases in this Chapter, we use a patience of 20, which implies that the training is stopped as soon as 20 iterations pass wherein the validation loss doesn't improve. If such a situation does not arise during training, the optimization process continues through the entire amount of specified epochs. Hence, we obtain the minimum validation loss until the training is either stopped or finished. Further, we test for architectures by not only varying the number of neurons in each layer, but also the number of layers. In Fig. 3.9, LSTM layers are represented by L and dense layers by D. So, the case L_1D_1 implies, that the architecture consists of one LSTM layer and 1 dense layer excluding the output dense layer. We consider cases such that the number of dense layers is less than or equal to LSTM layers ($D_\# \leq L_\#$), since the purpose of these layers is to improve local resolution for the information already learnt by the LSTM layers. Therefore, an excessive number of such layers is not required and if used may lead to over-representation of the model wherein some neurons may not even get activated.

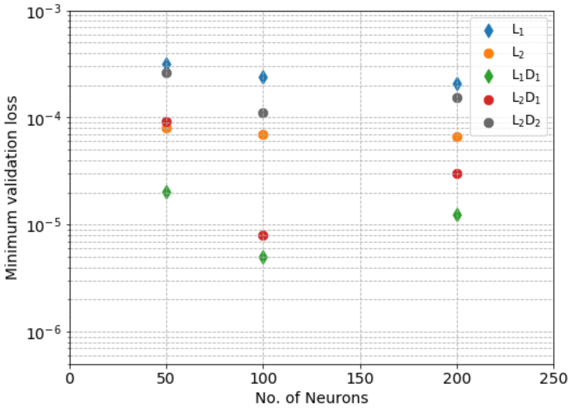


Figure 3.9: Network architecture sensitivity

We observe that, in general, adding a single dense layer leads to a better representation of the unseen data when compared to training only with LSTM layers. Also, an increase in the number of neurons in each layer improves the validation loss. This is typical, because less number of neurons implies less number of IPs are available to represent the input-output mapping. On the other hand, for cases with more neurons and more layers such as L_2D_2 with 200 neurons, a relatively higher loss is obtained. This can be attributed to the fact that a more complex architecture typically needs more iterations for improvement (since the convergence of IPs is slower) or that the framework detected overfitting and training was interrupted as explained earlier. We also obtain good accuracy with $\mathrm{L}_2\mathrm{D}_1$, but since the order of loss is similar for $\mathrm{L}_1\mathrm{D}_1$ case, we adopt the latter, which is a simpler model for implementation.

3.4 Example applications

We illustrate our approach using four examples of application. First, in Sec. 3.4.1, we explain the practical implementation of the developed surrogate model by utilizing the heterogeneous linear elastic 2-D unit cell which we introduced Sec 3.3. Second, we exemplify the modularity of the proposed approach for studying finite structures with more than one type of unit cell in Sec. 3.4.2. Then, we illustrate the generality of the approach by considering a nonlinear unit cell that exhibits hyperelastic behavior in Sec. 3.4.3. Finally, in Sec. 3.4.4, we demonstrate the case of a stiffened inclusion which can be used to study and predict bandgaps for finite structures. For all cases, we evaluate the performance of the surrogate model (DD) when implemented for large structures by comparison with corresponding high fidelity results from FE analysis.

3.4.1 Implementation of large structures via surrogate modeling of unit cell

In this case we consider the 2-D unit cell shown in Fig. 3.4. We develop the surrogate model using a total of 20000 samples that represent displacement-force relations and the proposed dual-step training approach as discussed in Sec 3.3. To generate the training samples, FE analysis is used which employs an explicit time integration scheme on Eqn. 3.8 wherein M is the mass matrix of the system and $\ddot{\mathbf{u}}^t$ is the acceleration at time t. Analogously, $\boldsymbol{f}^t$ is the force at time t, which based on our problem are the internal forces. This leads to finding the displacements at the right ends of all the cells from the computed end forces.

$$M\ddot{\mathbf{u}}^t = \boldsymbol{f}^t \tag{3.8}$$

It is important to note that the developed surrogate unit cell, is trained for displacements and forces that are represented as sequences in time domain. However, from an implementation perspective, we want to generalize the cell and simulate a periodic finite structure, such that only the displacement at the left end of the first cell is prescribed (known apriori). All the other quantities such as displacement at right ends and end forces are derived quantities. Consequently, the predictions from the periodic structure consisting of surrogate unit cells needs to be made at every instant of time. To do so, we use the *stateful* model of LSTM layers which decouples the internal state resetting from the updates made to the network IPs. This allows us to sequentially predict at every time step without losing history information along with giving us control over the update and reset of internal states. Finally, we use the predicted forces at each time step to compute the approximate velocity and displacement at all the nodes (where displacements are not prescribed) using an explicit scheme. These displacements then serve as an input for the surrogate model in the following time step. This entire solution scheme is depicted using a flowchart shown in Fig. 3.10.

Surrogate model evaluation for periodic structures

In this section, we analyze the behavior of the surrogate unit cell against input data unseen by the model during training. To do so, we test the surrogate model with pure harmonic signals, specifically sinusoidal displacements as test inputs to predict the forces and compute (by integration) displacements at the right end/s. Then, we compare these predictions, which we denote by data-driven (DD) results to those obtained by running a full FE analysis for the system. We also compute the results for static condensation to illustrate the benefits of the proposed approach over the former. We use a $1c$ (1×1) and $3c$ (3×3) configuration for preliminary validation since the training samples were generated for these configurations. Further, we conduct analysis across the frequency spectrum of the input considered, but with pure sinusoidal signals as inputs applied at the two left global nodes of $\mathrm{C}_{1,1}$ in both directions (e_1 and e_2). To test and analyze the behavior of the surrogate unit

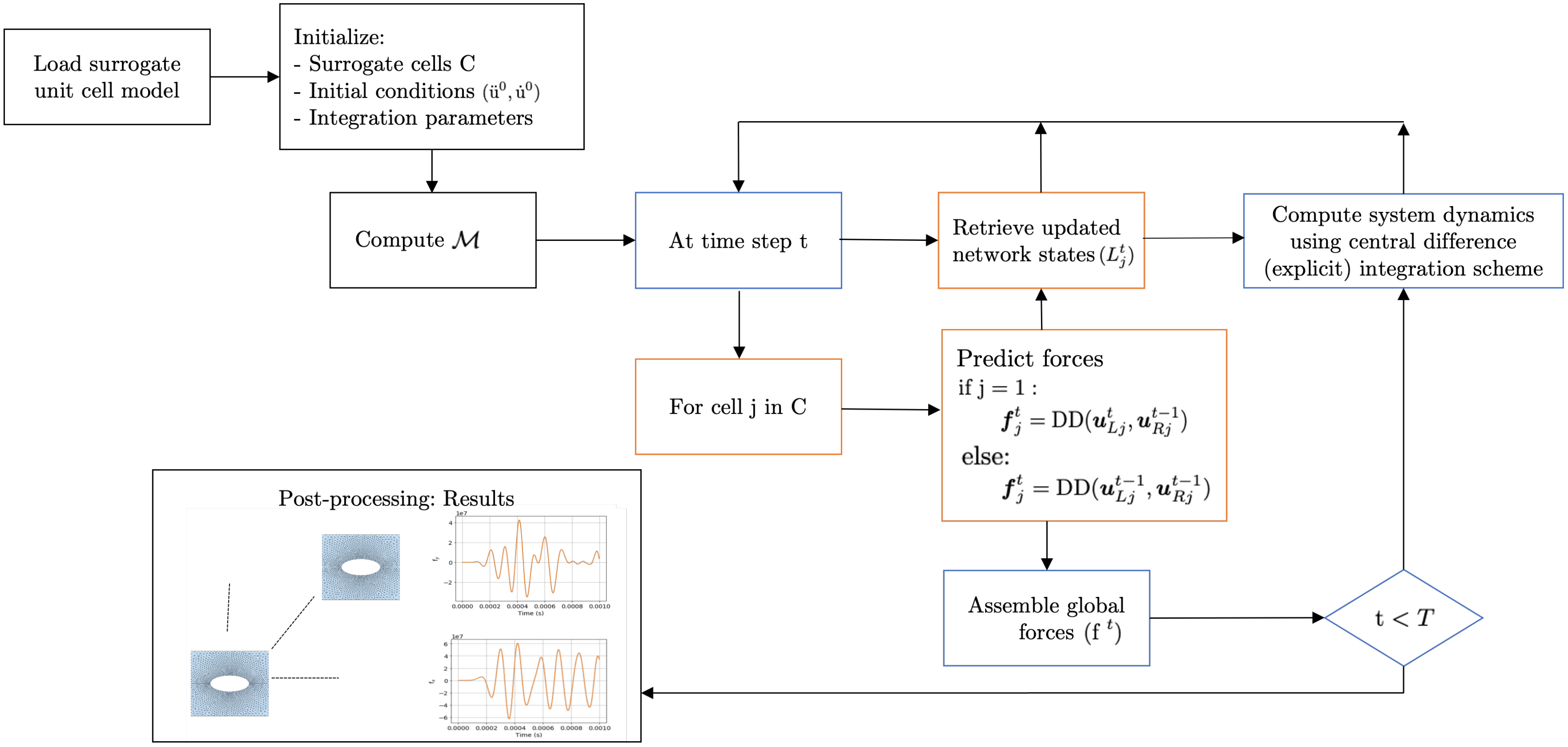

Load surrogate unit cell model
Initialize:
- Surrogate cells C
- Initial conditions $(\ddot{u}^0, \dot{u}^0)$
- Integration parameters
Compute $\mathcal{M}$
At time step t
For cell j in C
Retrieve updated network states (L_j^t)
Predict forces
if j = 1 :
$\boldsymbol{f}_j^t = \mathrm{DD}(\boldsymbol{u}_{Lj}^t, \boldsymbol{u}_{Rj}^{t-1})$
else:
$\boldsymbol{f}_j^t = \mathrm{DD}(\boldsymbol{u}_{Lj}^{t-1}, \boldsymbol{u}_{Rj}^{t-1})$
Compute system dynamics using central difference (explicit) integration scheme
Assemble global forces $(\mathrm{f}^{\,t})$
$\mathrm{t} < T$
Post-processing: Results
1e7
Time (s)

Figure 3.10: Implementation procedure for large finite structures

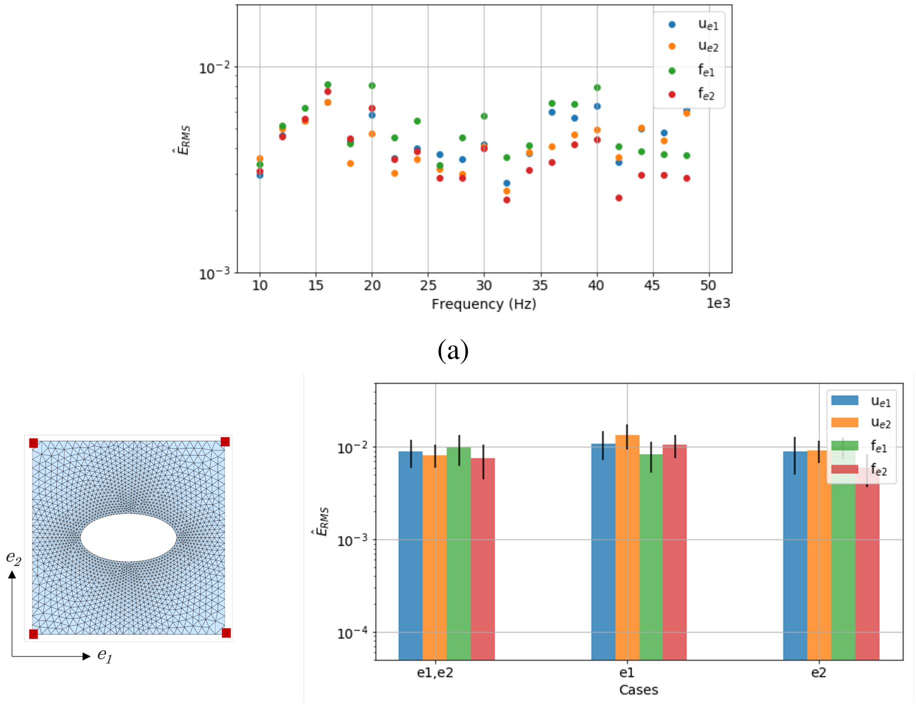
cell for a large number of predictions, we use a quantitative measure, namely the average root mean square error: $\hat{E}_{\text{RMS}} = E_{\text{RMS}}/(n_d \times n_f)$, such that n_d and n_f represent the total number of global nodes and number of cases (frequencies) at which the model accuracy is analyzed, and E_{RMS} for an output parameter $\mathbf{y}(t)$, is given by:

$$E_{\text{RMS}} = \frac{1}{\max(\mathbf{y}_t^o)} \sqrt{\frac{1}{T} \sum_{t=1}^{T} (\mathbf{y}_t^o - \mathbf{y}_t^p)^2} \tag{3.9}$$

wherein, $\mathbf{y}^o$ and $\mathbf{y}^p$ is output obtained from FE analysis and DD model respectively, and T is the total time steps for which we conduct the simulation. Note that for all the simulation results shown in this Chapter, we utilize a T which is atleast two times of that for which the model was trained. This way, during model evaluation we can ensure that the error does not propagate with time.

Figure 3.11a shows the errors obtained for predicted forces and the computed displacements in both directions over the entire frequency spectrum. We consider the forces predicted on all four global (corner) nodes and the displacements at the right end for computing these errors. We observe good accuracy for the predicted forces and computed displacements since the errors are less than 1%. The errors remain in the same range across the frequency spectrum which indicates good model learning and representation across the entire input spectrum. This particularly happens due to the development of the dual-step approach for training which in a way uses the information of different input frequencies via the tangent stiffness loss function.

We also analyze accuracy of the unit cell for different loading scenarios as shown in Fig. 3.11b. In this study, we consider different excitation scenarios, i.e. if the excitation is only in e_1 direction or e_2 direction or both. We observe that the model is able to capture similar accuracy for all cases even though the training has been done by providing input excitation in both directions at the left end. It is important to emphasize here that these results enable us to use the unit cell in different loading scenarios and does not restrict us to only one kind of application.

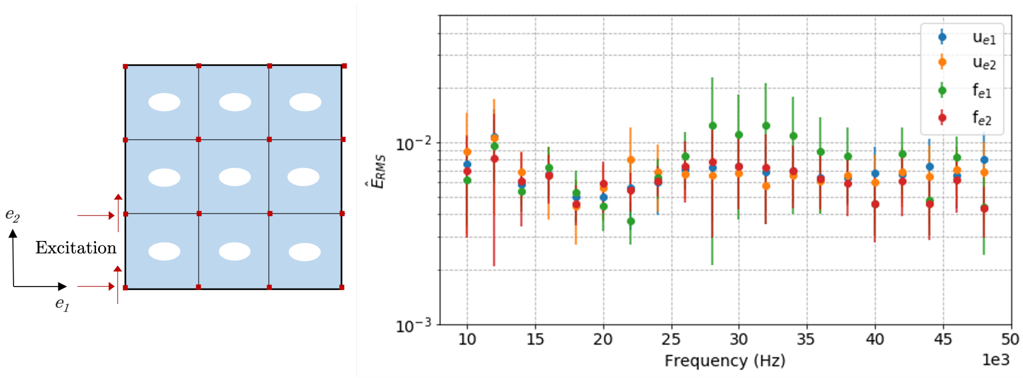


(a)

(b)

Figure 3.11: Validation results for single cell over entire frequency spectrum (a) RMS errors for forces and derived displacements, (b) Cumulative errors for different loading conditions

Further, we conduct a similar error analysis for a $3c$ (3×3) configuration. We show the problem set-up and computed errors in Fig. 3.12a. We apply the excitation in both e_1 and e_2 directions on the left global nodes of $\mathrm{C}_{1,1}$ such that the displacements at all other global nodes are a consequence of this excitation. Similar to the $1c$ (1×1) analysis, we show the $\hat{E}_{\mathrm{RMS}}$ for predicted forces and computed displacements across the considered frequency spectrum. However, in this case due to a large number of global nodes, we also show the standard deviation of the error using the error bar in Fig. 3.12a. This metric represents the variance of the output parameter over all the nodes and helps in identifying cases if the predictions are extremely good or bad for certain nodes with respect to the others. Overall, we observe errors around 1% for all parameters with no specific trends in errors or error bars across the frequency spectrum.



(a)

Figure 3.12: Validation results for 3 × 3 periodic structure over entire frequency spectrum with problem set-up (left) and displacement and force errors over entire frequency spectrum (right)

Finally, we analyze the unit cell for a frequency of 50kHz and compare the predicted results with that of the FEM and static condensation (SC). It is important to note that the minimum wavelength or highest frequency that can be resolved by SC is 14.5 kHz as discussed in Sec. 3.3.1. Beyond this frequency the SC method fails to provide accurate dynamic predictions. We illustrate this using the case schematic shown in Fig. 3.13. We apply the excitation at the left end in e_1 and e_2, and observe the response at the right end, at points A and B. We perform this analysis using three methods: (1) proposed data-driven (DD), (2) finite element (FE), and static condensation (SC) methods, results for which are shown in Fig. 3.14. We can see that the SC method though extremely computationally efficient is not able to capture the high frequency response and in a way acts as a filter to these high frequencies. On the other hand, the data-driven method is able to provide displacements which match well with the high-fidelity FE solution. Further, we also provide quantitative measures to compare these methods as shown in Fig. 3.15 such that Fig. 3.15a shows the ratio of the peak amplitudes captured by FE to SC at point A and B and Fig. 3.15b shows the $\hat{E}_{\text{RMS}}$ for displacements in e_1 and e_2 directions. As observed qualitatively earlier, the errors are two orders higher in static condensation when compared to the proposed data-driven method. Hence, we can safely establish that the scope of the proposed method is much wider in terms of application.

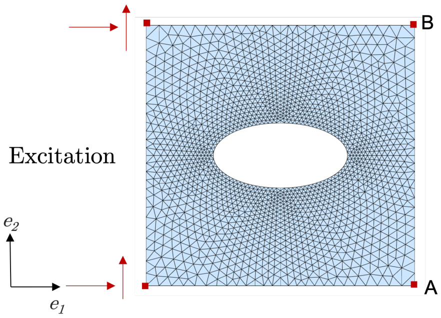
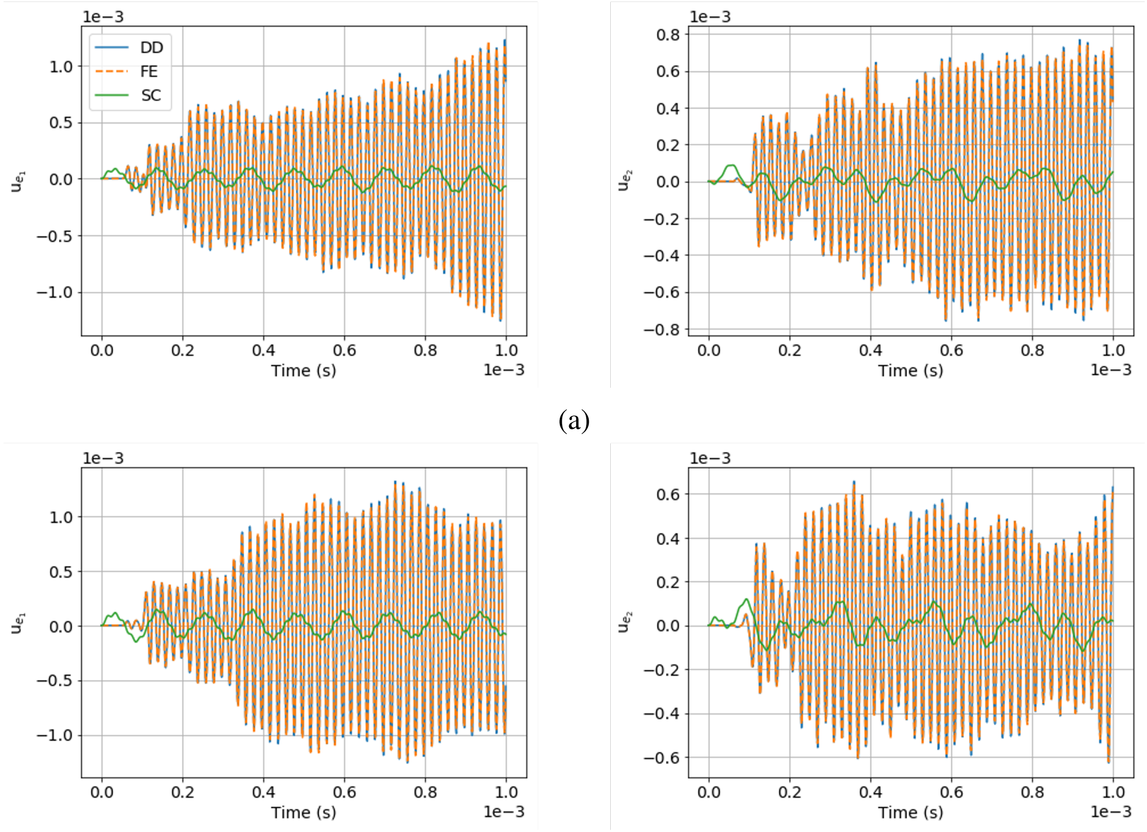


Figure 3.13: Case definition for comparison of static condensation with proposed method

Exhaustive validation

Despite the fact that unseen input displacements are used to analyze the accuracy, the cases discussed in the earlier section are expected to perform well and only serve as preliminary validation. This is because $1c$ and $3c$ configurations are used for training. Thus, it is important to analyze the accuracy of the model for larger structures and configurations. In this section, we use one such configuration which has 8 unit cells in e_1 and 3 unit cells in e_2 direction as shown in Fig. 3.16. Further, instead of applying excitation only on $\mathrm{C}_{1,1}$, we prescribe a displacement of frequency 50kHz in e_1 at the left end of the entire structure, i.e. at the left global nodes of $\mathrm{C}_{1,1}$, $\mathrm{C}_{2,1}$, and $\mathrm{C}_{3,1}$.

We perform analysis for the described case using the proposed approach and with FEM and show the wavefield comparison in Fig 3.17. In this figure, we plot the computed displacements for all global nodes along e_1 direction for a constant value of position along e_2, i.e for $e_2 = 0.9$m over the considered simulation time of 0.005s. The color on the plot indicates the magnitude of displacement obtained at a particular spatial and temporal instant. We observe that the analysis using the surrogate model (DD) predicts the wavefield accurately when compared to the FE solution. We also conduct the same analysis using static condensation [19] method, which in Fig. 3.17c shows its inability to predict an accurate wavefield. This is attributed to the fact that this method cannot capture frequencies higher



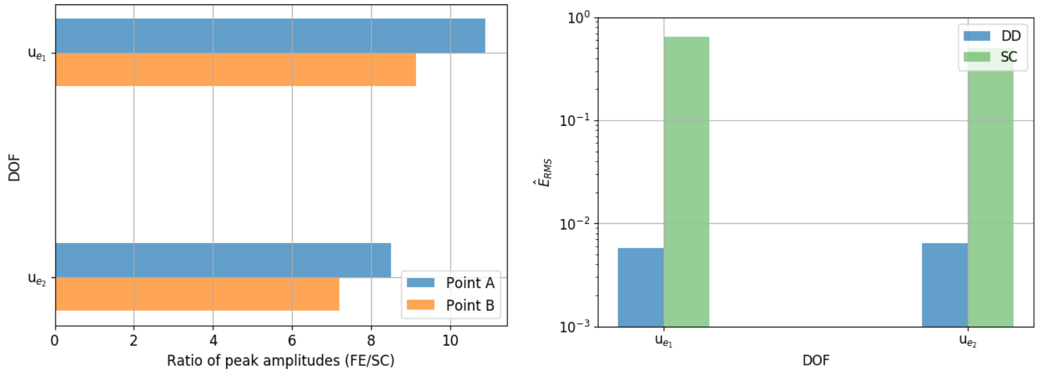
(a)

(b)

Figure 3.14: Comparison of computed displacements using proposed method (DD), finite element method (FE), and static condensation (SC) in time domain for (a) Point A and (b) Point B

than the highest natural frequency of the unit cell obtained as discussed in the previous paragraphs.

Finally, we show error heatmaps for this case of a large finite structure for all nodes and degrees of freedom in Fig. 3.18 over randomly selected discrete time instants through the simulation time. We use nodes to denote the spatial position. There are a total of 36 global nodes for the considered configuration of which 4 nodes are used on the left end to prescribe displacements. The remaining 32 nodes are shown in Fig. 3.18 such that a group of 8 nodes represent all the global nodes at a particular value of e_2, and $e_2 \in [0, 0.9]$. The intensity of heat (color) indicates the magnitude of normalized error ($\mathcal{E}_n$) between the predicted ($\mathbf{y}^p(e_1, t)$) and actual value($\mathbf{y}^a(e_1, t)$) of the observed displacement and is given



(a) (b)

Figure 3.15: Quantitative measures comparing static condensation and proposed method (a) Ratio of peak amplitudes captured in e_1 and e_2 displacement (u), (b) Cumulative RMS error for e_1 and e_2 displacements (u)

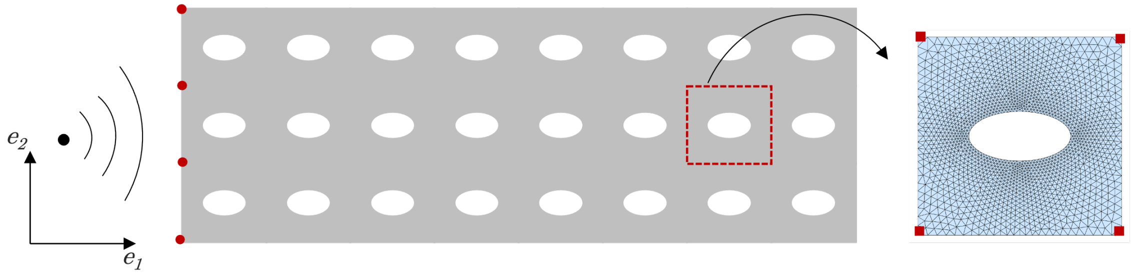
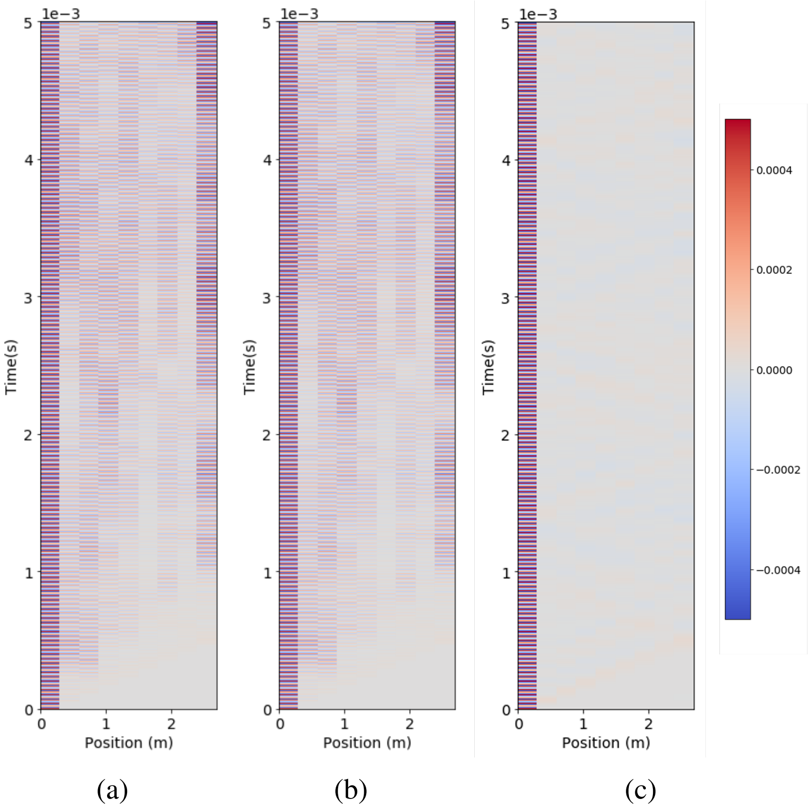


Figure 3.16: Case definition for studying larger periodic structures

by $\mathcal{E}_n = \left|\frac{\mathbf{y}^a(e_1,t)-\mathbf{y}^p(e_1,t)}{\mathbf{y}^a(e_1,t)}\right|$. These maps are useful to identify signs of error propagation over spatial and/or temporal domain. For e.g. for the results shown, we can see that the highest magnitude of errors does not necessarily occur at the last node or end of simulation. We do observe that the errors are negligible at the start of the simulation for many nodes. This is because the wavefront has not yet reached those nodes and hence the parameter values are close to zero. Further, the errors remain bounded within 1.6% for displacements in both directions. Overall, these results indicate that the surrogate model captures the wave accurately and the errors also remain bounded in space and time.

To conclude this section, we report some aspects related to the computational performance of our approach. Remarkably, the proposed dynamic homogenization approach provides a comparable solution with respect to FEM at a fraction of the computational

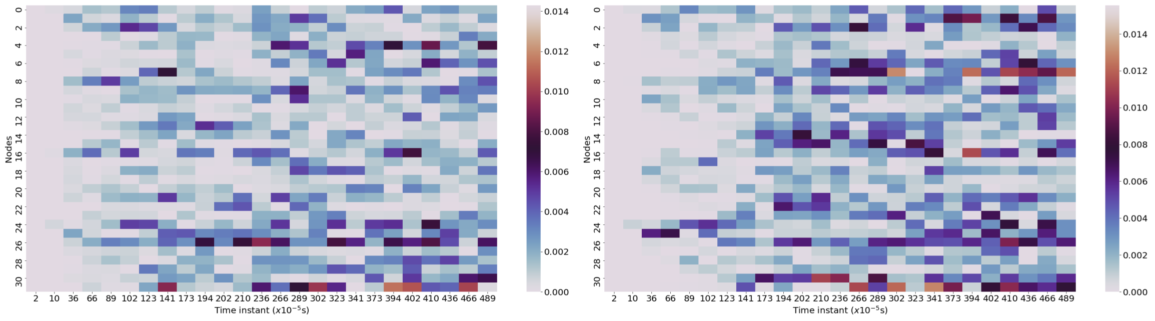


(a) (b) (c)

Figure 3.17: Wavefield comparison at $e_2 = 0.9$m for displacements in e_1 direction using (a) FE, (b) DD and (c) SC methods

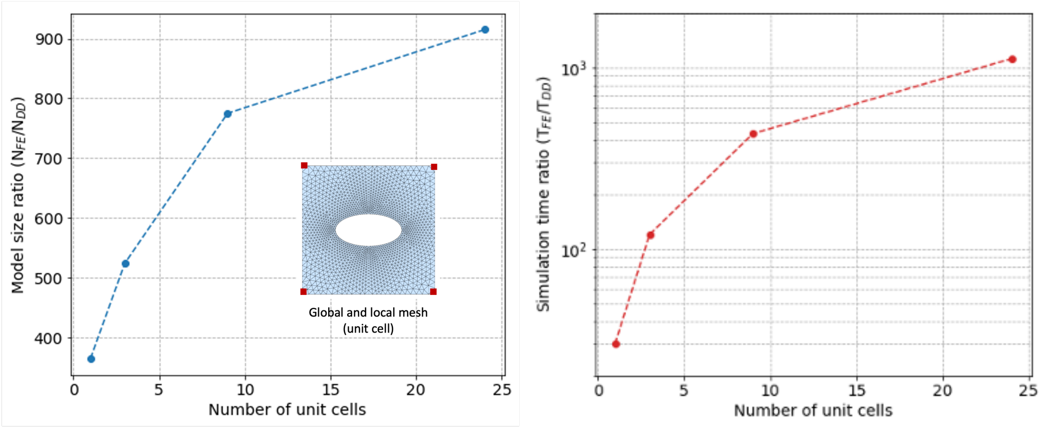
cost. Furthermore, the advantage of the proposed method is even more significant when a larger model is considered. This is shown in Fig. 3.19a which quantifies the increase of the relative model size, defined as the ratio of the number of DOFs in the FEM (N_{FE}) and the DD model (N_{DD}) as a function of the number of unit cells that comprise the finite structure. Further, in Fig. 3.19b we quantify the computational benefit using the ratio of simulation time in FEM to the DD model (T_{FE}/T_{DD}), which significantly increases as the number of comprising cells increase.

It is worth noting that all the examples considered in this Chapter and the comparison above were performed on the same hardware to enable a consistent comparison. The finite element simulations as well as data-driven predictions were executed on a GPU-based machine with an NVIDIA accelerator which allows for parallelization. Overall, we can see that the DD implementation results in a drastic reduction in the required computational



(a) (b)

Figure 3.18: Error heat maps on all global nodes other than excited nodes for displacements in (a) e_1 direction, (b) e_2 direction



(a) (b)

Figure 3.19: Quantified computational performance (a) Relative model size and (b) Relative simulation time as a function of unit cells

time. This could be key in enabling efficient computations in time-critical applications, for example studying large structures for vibration management (as demonstrated in subsequent sections).

3.4.2 Study of large arbitrary structures via modular training

Let us first summarize the results from previous sections. For a given unit cell, we first simulated 20000 input scenarios in the form of displacements via FEM. We then utilized those results to train the unit cell and develop its surrogate model. Subsequently, we observed that the surrogate unit cell is able to perform well across the considered frequency spectrum (even for very high frequencies where established methods such as static condensation fail to provide accurate results) and also when used to simulate large finite structures. While valuable, this on its own provides limited opportunity to explore structures for vibration management.

Using this section, we demonstrate the modularity of the proposed approach by first developing a 2-D homogeneous unit cell (without hole) and then implementing it with the previously developed heterogeneous unit cell to simulate large arbitrary finite structures. We adopt the same procedure to develop and validate the surrogate model for the homogeneous unit cell as discussed in Sec. 3.3. We provide the details and validation results for this unit cell in the Appendix. Further, we consider two examples of large finite structures that employ multiple types of unit cells to demonstrate the accuracy and potential applications of the proposed approach.

Wave propagation in arbitrary structures

We analyze the accuracy of the proposed method for a problem that includes a randomized arrangement of the two developed 2-D unit cells, i.e. one with a hole (UC$_1$) and one without (UC$_2$). Let us consider a large finite structure as shown in Fig. 3.20 which has a square notch. This large structure can be discretized into 14 UC$_1$s and 12 UC$_2$s as shown. For the purpose of this case, we impose a sinusoidal excitation of 20kHz for a simulation time of 2ms on the left global nodes of C$_{1,1}$ in e_1 direction. We perform this study using the proposed method and implementation algorithm discussed in Sec. 3.4.1. The only difference in the implementation is that, we now need to load and initialize two types of surrogate unit cells instead of one. Subsequently, we compute the solution for this problem using FEM and use it to compare the behavior of the proposed approach. Note that while solving using FEM, the entire spatial domain is disctretized with a local mesh such as the one shown for different unit cells in Fig. 3.20, whereas the data-driven approach does not need such a local mesh as that information has already been learnt by the surrogate during training.

It is important to emphasize here that this analysis represents a very difficult test for the surrogate unit cells for various reasons. First, the imposed displacement is given by a sinusoidal, which is very different than the samples with which it has been trained and does not belong to the same family of curves. Second, the unit cells have been trained separately and do not have the experience of being assembled with a different type of unit cell. Consequently, the dynamic behavior that needs to be predicted is significantly different and it is expected that the unit cell has learnt the physics-based information that can enable it to predict forces accurately.

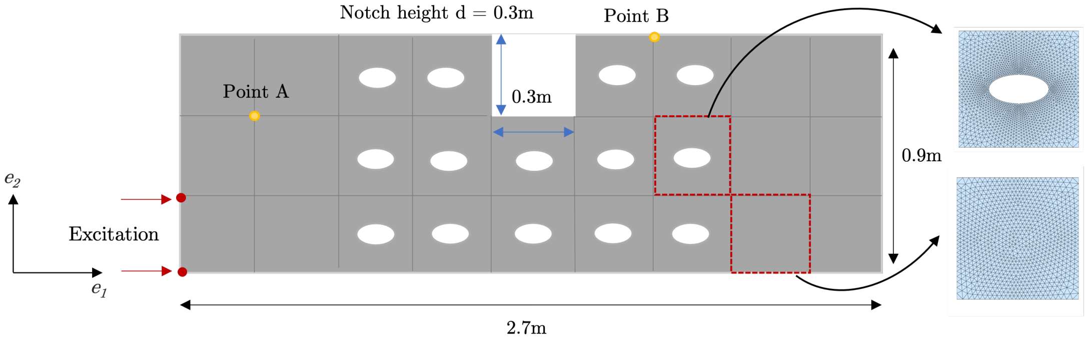
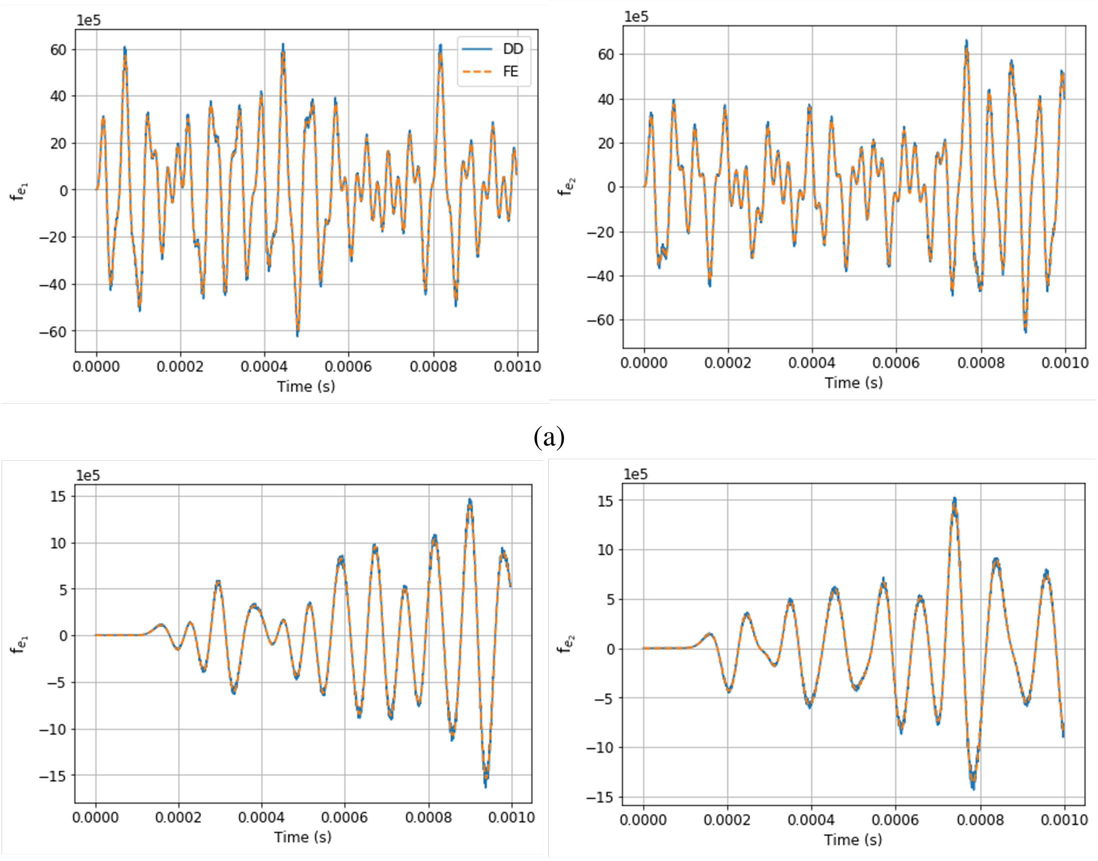


Figure 3.20: Case definition for finite structure with modular unit cells developed separately

In Fig. 3.21 we compare the resulting forces (in e_1 and e_2 direction) in time domain for points A and B (denoted in Fig. 3.20) for both the DD and FE models. We can see that there is an excellent agreement between both models for the entire simulation duration. That said, what is more interesting here, is that we were able to capture the global behavior without losing any local information at a fraction of the computational cost. This is because the FE simulation took 25 hours to complete, while the DD model prediction ended in under 130s.

To present a comprehensive assessment of the goodness of the DD model, we also study the coefficient of determination denoted as R^2 score at each time increment given by Eq. 2.18. We compute these scores over the entire simulation from which snapshots for four time increments are shown in Fig. 3.22. It is remarkable that not only the forces have



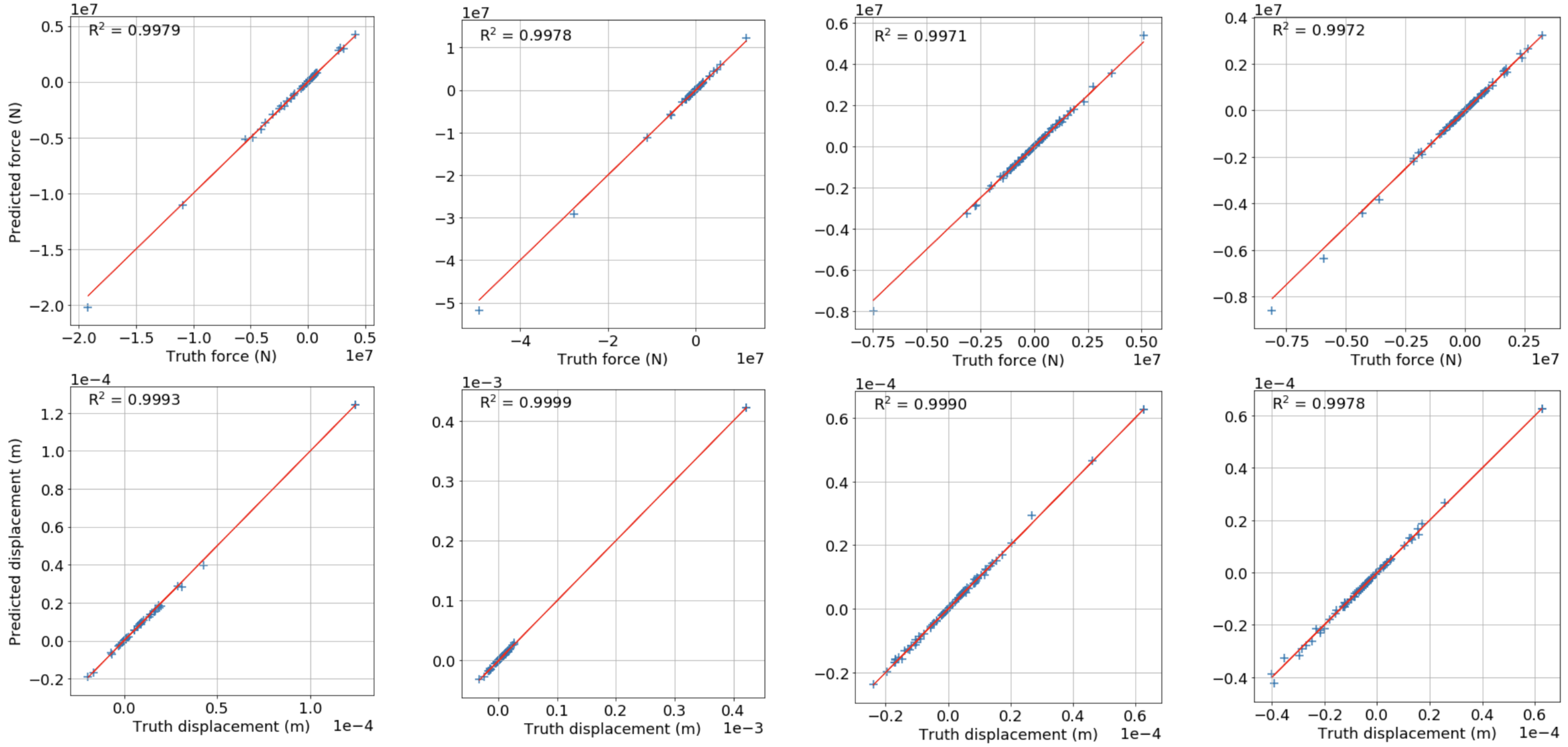
(a)

(b)

Figure 3.21: Validation of predicted output parameters (forces in e_1 and e_2 directions) for (a) Point A and (b) Point B

a great correlation, but the displacements also match well. For the forces, there is a direct relationship with how well the surrogate unit cell is performing in the simulation since it has been trained for that. On the other hand for the displacements, there is no explicit relationship as they are derived quantities from the predicted forces (displacements in the next time step are dictated by the corresponding force prediction within the structure).

Furthermore to summarize the goodness of the prediction through the entire simulation and not just for specific time instants, we compute the R^2 score for each output parameter, for every time instant, and for all the global nodes. We show these results in Fig. 3.23, wherein we can see that the R^2 score is always greater than 99%. These plots provide an overall estimate of how good the DD model can predict the solution at every time step.



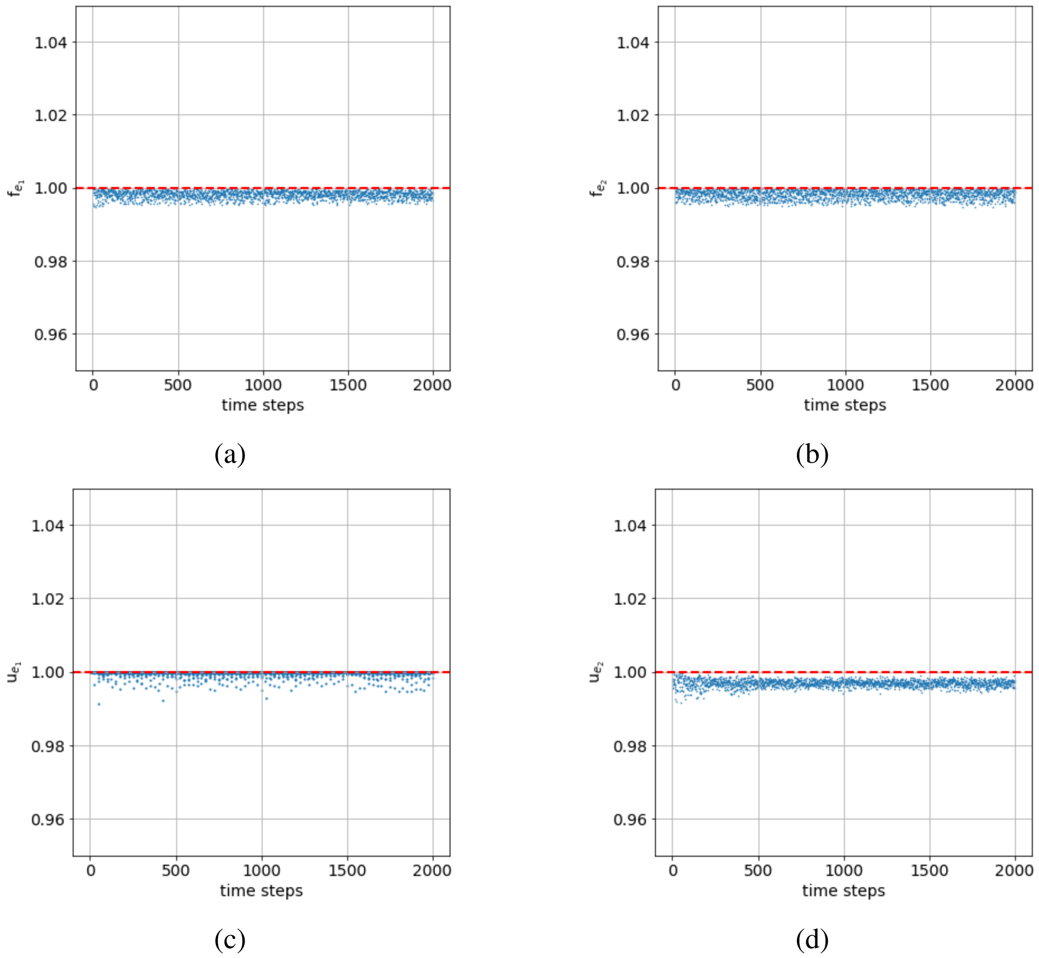
(a)

(b)

(c)

(d)

Figure 3.22: Snapshots for R^2 scores for all global nodes at specific times, (a) 0.201ms, (b) 0.467ms, (c) 1.400ms, and (d) 1.851ms



(a) (b) (c) (d)

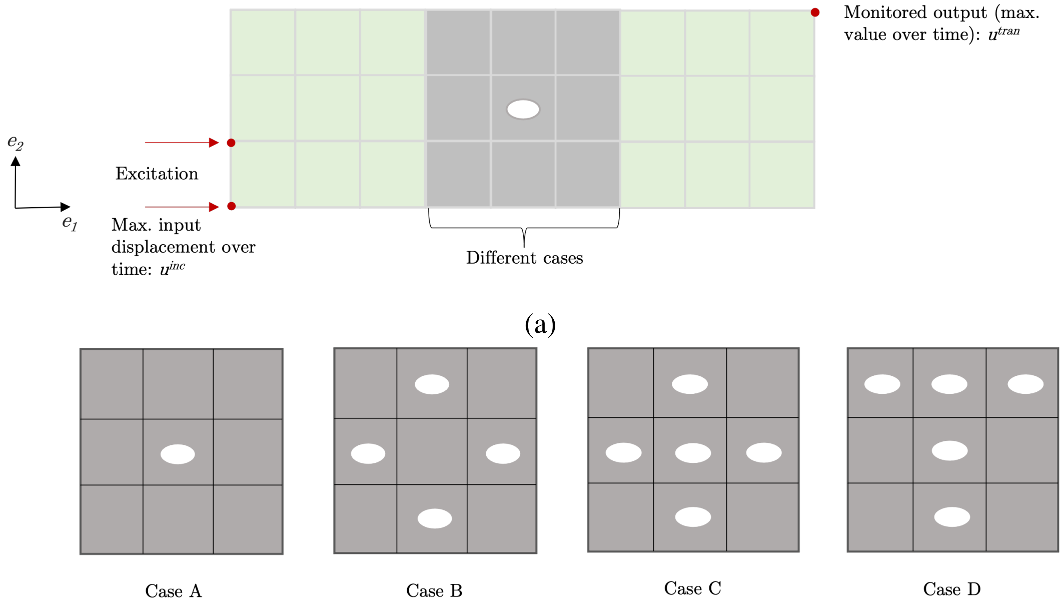
Figure 3.23: Cumulative R^2 scores over simulation time for all global nodes for (a) Predicted force in e_1 direction, (b) Predicted force in e_2 direction, (c) Computed displacement in e_1 direction and (d) Computed displacement in e_2 direction

Moreover, it enables us to visualize if there is any error accumulation during the simulation (the score will decrease with increase in time steps). Overall, the DD model achieves a high correlation with the FE model over the entire spatial and temporal domain.

Geometric optimization using surrogate unit cells

In this example, we apply the proposed method for simulating a large structure with different arrangements of unit cells to analyze the vibration attenuation ability of each configuration. Such studies are particularly useful in applications which need to have certain geometric/weight characteristics but also need to reduce the transmission of vibrations across the structure [118]. Thus, we illustrate the benefits of the proposed method in one of its potential application areas namely, geometric optimization.

We consider a large structure such as one shown in Fig. 3.24. To consider the effect of the presence and configuration of holes on vibration attenuation, we consider different cases by altering the center three columns (grey portion of Fig. 3.24a) of the structure. As shown in Fig. 3.24b, we consider four cases for this study which have a different number and arrangement of holes. Further, we impose a sinusoidal excitation of 20kHz on the left end of C$_{1,1}$ in the e_1 direction and also note the amplitude used, as u_{e1}^{inc}. After the prediction, we monitor the displacement amplitude in e_1 direction over time at the right end of C$_{3,9}$, and denote it as u_{e1}^{tran}.


Monitored output (max. value over time): u^{tran}
e_2
e_1
Excitation
Max. input displacement over time: u^{inc}
Different cases
Case A
Case B
Case C
Case D

(a)

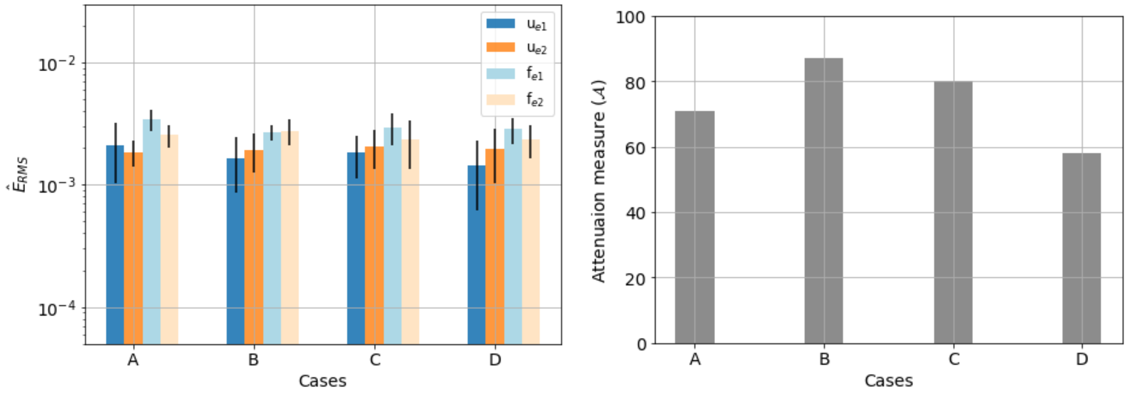
(b)

Figure 3.24: Case definition illustrating application of proposed method in geometric optimization, (a) Set-up , (b) Four different cases considered for this study

We repeat this study for all four arrangements to analyze the structure's vibration attenuation capability. A suitable measure of the attenuation is provided by the attenuation measure $\mathcal{A}$ defined as the ratio between the maximum amplitude of the transmitted wave to the maximum amplitude of the imposed excitation:

$$\mathcal{A} = \frac{\max\, u_{e1}^{\mathrm{tran}}}{\max\, u_{e1}^{\mathrm{inc}}} \times 100 \tag{3.10}$$

We show the results for this study in Fig. 3.25. First, for the purpose of completeness we conduct the same study using FEM and compare the DD model's performance with it in Fig. 3.25a. In this plot, we show the RMS error (Eq. 3.9) averaged over all the global nodes for predicted forces and derived displacements. We observe that the errors are low ($< 1\%$), indicating the reliability of the proposed DD model. Further, the variance of these errors (shown as black error bars in Fig. 3.25a), indicate a similar order of error across the entire finite structure. More interestingly, we can observe the attenuation measure for the four cases considered in Fig. 3.25b. The results indicate that Case D has the lowest value of $\mathcal{A}$ when compared to other three cases, which means that Case D will transmit only 60$\%$ of the incident wave. Though there might be better and more cases that may give a lower $\mathcal{A}$ value, based on the considered arrangements, the proposed method provides an efficient way (each case takes 142s using DD model vs. an average of 22hrs using FE model) to study different cases for vibration management in finite structures.



(a)

(b)

Figure 3.25: Results for geometric optimization case, (a) Accuracy with respect to high-fidelity FE solution, (b) Attenuation measure for different cases

3.4.3 Application to a nonlinear problem

When problems become nonlinear, there are not many alternatives to FEM for studying the dynamic behavior of finite structures. This is primarily because of the fact that the stiffness of the unit cell and consequently the overall structure evolves with time and is history-dependent. To this effect, our approach presents an efficient way to analyze such structures as it is not limited to a particular type of structure or nonlinearity. In this example, we demonstrate this capability of our approach using a 2-D unit cell that follows a hyperelastic material model that entails material nonlinearity. We adopt the procedure described in Sec. 3.3 to develop a surrogate model for the hyperelastic unit cell, then utilize it to form large finite structures and finally analyze their dynamic behavior and accuracy.

Problem Definition

We consider a homogeneous square 2-D unit cell with an edge length of 0.3m as done in Sec. 3.4.2, but with a hyperelastic material model to describe their dynamic behavior. Specifically, we use the Neo-Hookean material model with a strain energy function [119] given by:

$$U = \frac{\mu}{2}(\bar{I}_1 - 3) + \frac{\kappa}{2}(J-1)^2 \tag{3.11}$$

where, μ and κ represent material properties namely, shear and bulk modulus respectively, and $\bar{I}_1 = I_1 J^{-\frac{2}{3}}$ with I_1 being the first invariant of the right Cauchy-Green deformation tensor. Also, J is the Jacobian of the deformation tensor F and is expressed as $det(F)$. For the presented study, we use $\mu = 0.4$MPa [119] and $\kappa/\mu = 10$ to allow for certain compressibility. To identify the range of the unit cell's nonlinear behavior and subsequently generate training samples, we generate material curves for the unit cell as shown in Fig. 3.26. We obtain these material curves by imposing constant amplitude harmonic signals across the frequency range and computing the amplitude of displacement at the other end of the unit cell. The obtained amplitude is then normalized with the input amplitude. We repeat this process for different input amplitudes which are depicted as different curves in Fig. 3.26. Note that, the input amplitude is depicted as the percentage of unit cell dimension (0.3m). We observe that, the material response is linear till an input amplitude of 0.5%. Beyond that the material model exhibits significant nonlinearity. Based on these findings, we generate training input displacements with amplitudes that span in low amplitude (exhibits linearity in this regime) and high amplitude (exhibits nonlinearity in this regime-till 8%) regime for a frequency range of 200-320Hz. We obtain the input displacements for training using the wavelet approach described in Sec. 3.3.1, for the identified amplitude and frequency range. Just like in the previous sections, we generate 2 sequences to represent each DOF for a global node of the 2-D unit cell.

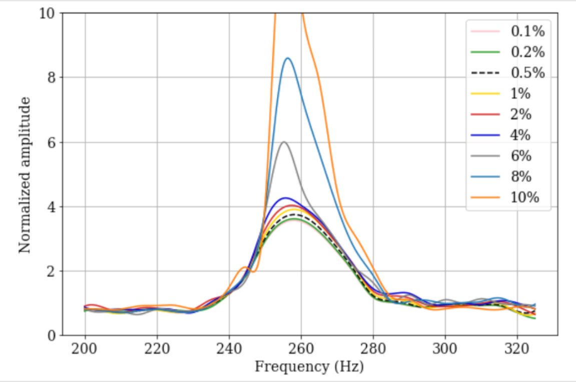


Figure 3.26: Material curves from implemented hyperelastic model

It is important to ensure a fairly uniform distribution of samples across the selected frequency and amplitude range. This would lead to a surrogate unit cell that is not biased towards any range of parameters. We study the sample distribution using a joint plot as described in Sec. 3.3.1. We show the detailed distribution for 0.1% amplitude samples in Fig. 3.27 such that each dot in the scatter plot represents one sequence. The corresponding amplitude and frequency distribution are shown on the top and right of the scatter plot which represent the maximum absolute amplitude and the dominant harmonic of the generated sequences. We generate 200 samples for each amplitude range (for which distributions

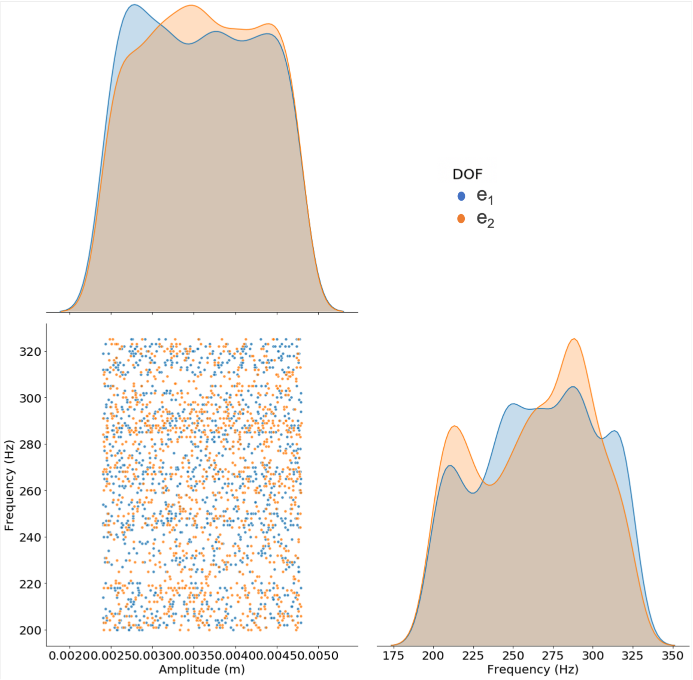
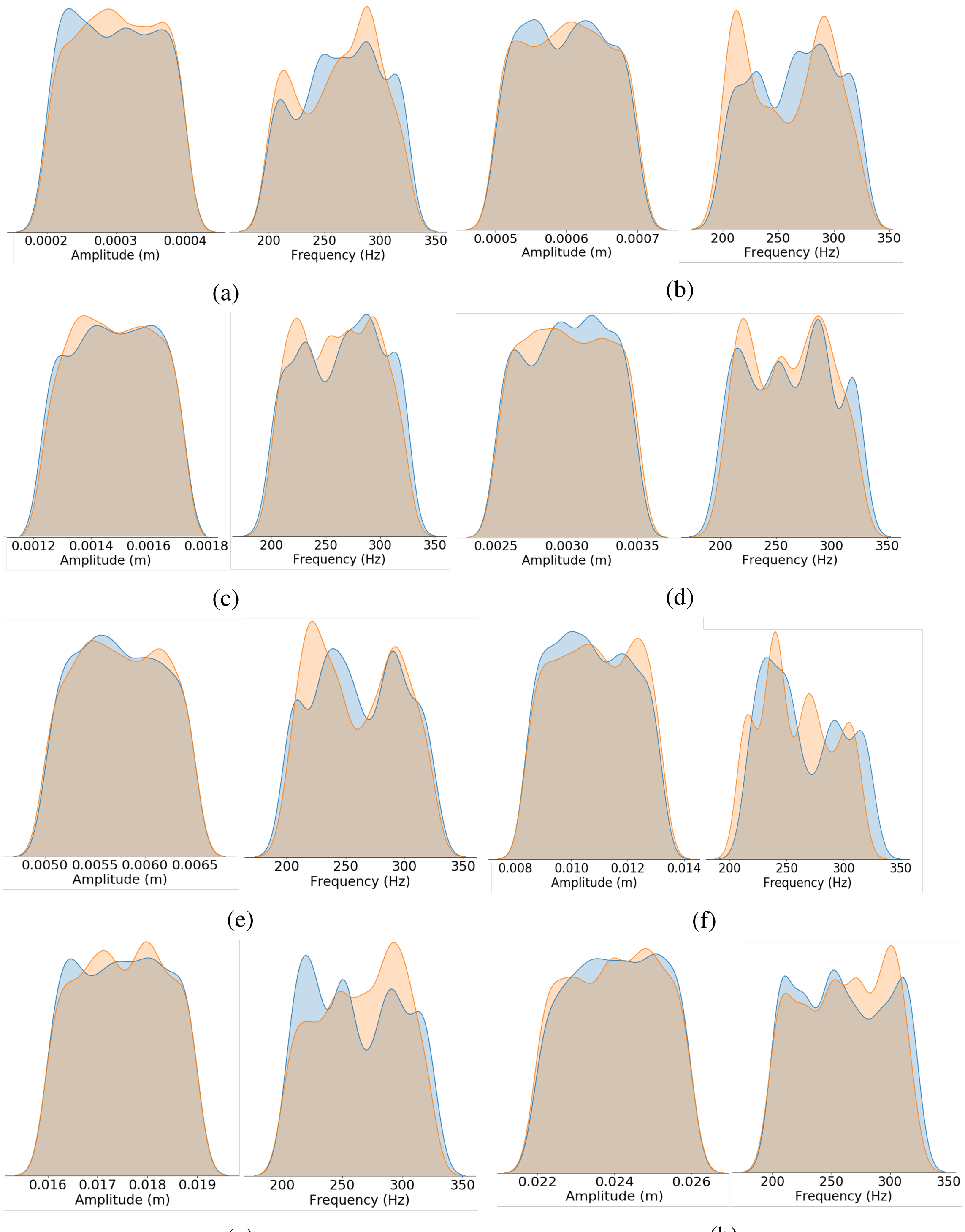


Figure 3.27: Detailed sample distribution for 0.1% prescribed amplitude

are shown in Fig. 3.28), resulting in a total of 16000 training samples.

Neural network architecture and results

We define the network to use 8 input features which represent displacements for 4 corner nodes, and the corresponding 8 output features for nodal forces. We train the model using LSTM layers for history retention and dense layers for local resolution. The dual-step training framework, explained in Sec. 3.3.2 is implemented with 1 LSTM layer and 1 dense layer, each having 500 neurons. Note that, this study requires a wider network (more number of neurons), resulting in more IPs, so that the statistical model can sufficiently represent the nonlinear relation between the displacements and forces. We also develop an analogous model for a unit cell with hole by using the same inputs and material model to study large modular structures with material nonlinearity. As a final note, it is worthwhile to emphasize that, this study is significantly complex with respect to the development of the surrogate model due to numerous reasons. This is because, in addition to the reasons



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3.28: Sample distributions over selected frequency range for different ranges of amplitude for x and y displacement (DOF); (a)-(c) Low amplitude range: 0.1%, 0.2%, 0.5% and (d) - (h) High amplitude range: 1%, 2%, 4% ,6%, 8%

that contribute to complexity in the linear elastic unit cell, for this case, due to the highly nonlinear material behavior, the data-driven unit cell has to represent a large variation in its dynamic behavior for a large range of amplitudes and that too for periodic/arbitrary structures with a higher number of comprising cells.

In order to analyze the prediction accuracy of the model, we first conduct an analysis on the homogeneous hyperelastic unit cell to generate the material curves shown in Fig. 3.29, but now using the trained surrogate (data-driven) model. We show these results in Fig. 3.29, such that the DD model is able to recover the material curves over the entire frequency and amplitude range. This serves as the first part of the model validation.

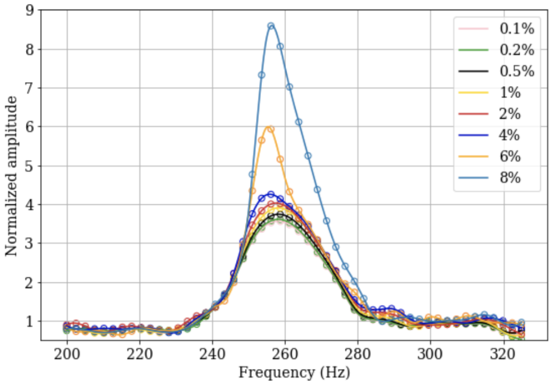
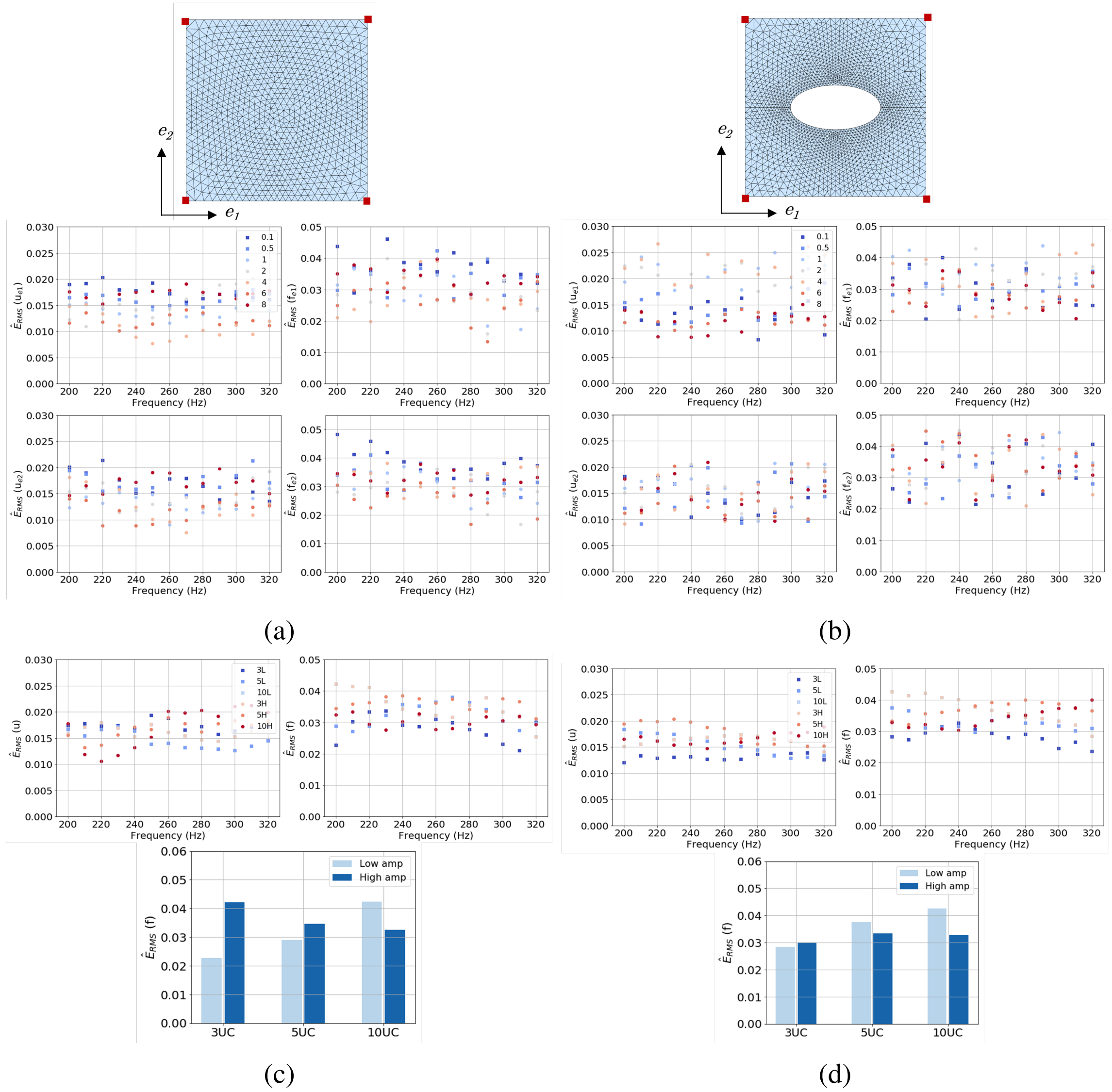


Figure 3.29: Validation of developed surrogate model by investigating recovery of material curve over entire frequency spectrum and amplitude range. Note: The lines indicate the values obtained from FE simulations and circles indicate computed values from ML predictions

Next, we compute the $\hat{E}_{\text{RMS}}$ using Eq. 3.9 to quantify the goodness of the DD model with respect to FE model. We perform these studies for a single as well as larger periodic configurations as shown in Fig. 3.30. We use the validation tests similar to those explained in Sec. 3.4.1, such that sinusoidal signals are applied in e_1 and e_2 at the left global nodes of $\text{C}_{1,1}$ for each configuration. We can observe in Figs. 3.30a and 3.30b that, for both the unit cells, the $\hat{E}_{\text{RMS}}$ is $< 5\%$ for forces and $< 3\%$ for the displacements. These errors are relatively higher than the linear elastic unit cell models discussed in Sec. 3.4.1. This is attributed to the more complex information that the model needs to learn for this case, as the behavior of the unit cell varies for the same frequency at different amplitudes.



(a) (b)

(c) (d)

Figure 3.30: Accuracy of developed hyperelastic surrogate models for square unit cells without (left) and with hole (right): (a,b) Single cell over entire frequency and amplitude range and (c,d) Periodic structures with larger number of repeating unit cells (UC): 3×3, 5×5 and 10×10

Nonetheless, since the errors are overall less, we establish that the developed models have good prediction accuracy. As mentioned earlier we also analyze the model behavior when used to study larger periodic configurations as shown in Figs. 3.30c and 3.30d. We perform the error analysis for 3-,5- and, 10-cell configurations. In this Figure, we show the errors computed for each of these configurations using 0.1% (low amplitude: L) and 4% (high amplitude: H) cases. We observe that the $\hat{E}_{\mathrm{RMS}}$ remains below 5%, which proves that the error does not propagate when we increase the number of unit cells in the structure. We also present cumulative results (averaged force prediction errors for all amplitudes and frequencies in the considered input spectrum) for these cases in the bar plots in Figs. 3.30c and 3.30d.

Finally, something that is probably relevant in terms of future applications of this approach, is its capability to estimate the behavior/ wave propagation characteristics for arbitrary large structures. So, we consider a large structure of 3×9 configuration (shown in Fig. 3.31) comprising of two types of unit cells, both of which exhibit hyperelastic material behavior. In this study, we apply a sinusoidal excitation on $\mathrm{C}_{1,1}$ at the left end in e_1 and predict the dynamic behavior of the entire structure over a simulation time of 50ms.

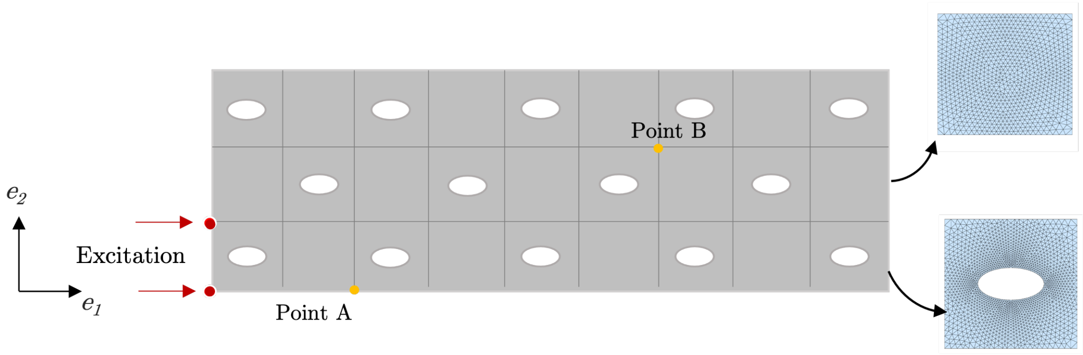
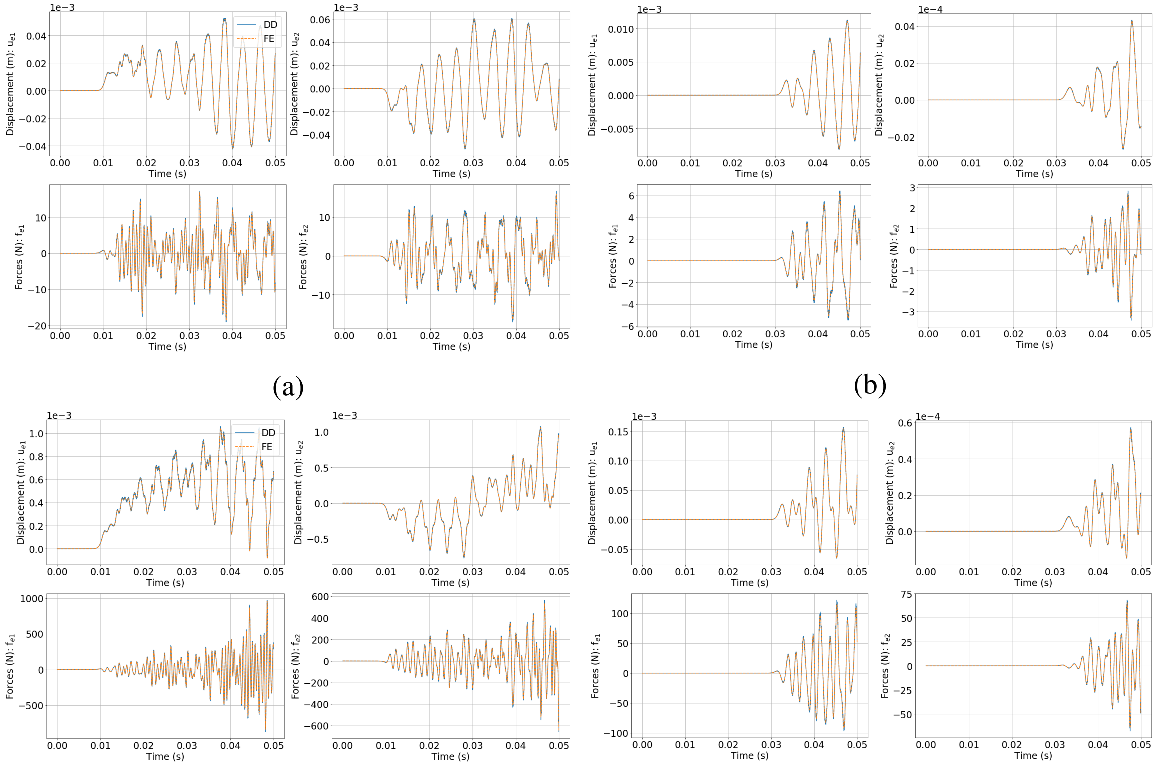


Figure 3.31: Case setup for study of large structures with hyperelastic unit cells

Figure 3.32, shows the time domain predictions of forces and computed displacements at points A and B. We conduct the same analysis twice, once with a low amplitude input excitation and then with high amplitude. We can see that for both cases, we obtain an excellent agreement for the output parameters when compared with the FE solution.

Furthermore, to summarize the goodness of the fit over the entire spatial and temporal domain, we compute the R^2 scores using Eq. 2.18 for all global DOFs over the entire simulation time. We report these scores using Fig. 3.33, which shows that we obtain R^2 scores of 96% or higher for this structure for both low and high excitation amplitudes. That is, the DD model has a high correlation with the FE model and is able to accurately capture the nonlinear behavior of the underlying unit cells.



(a)

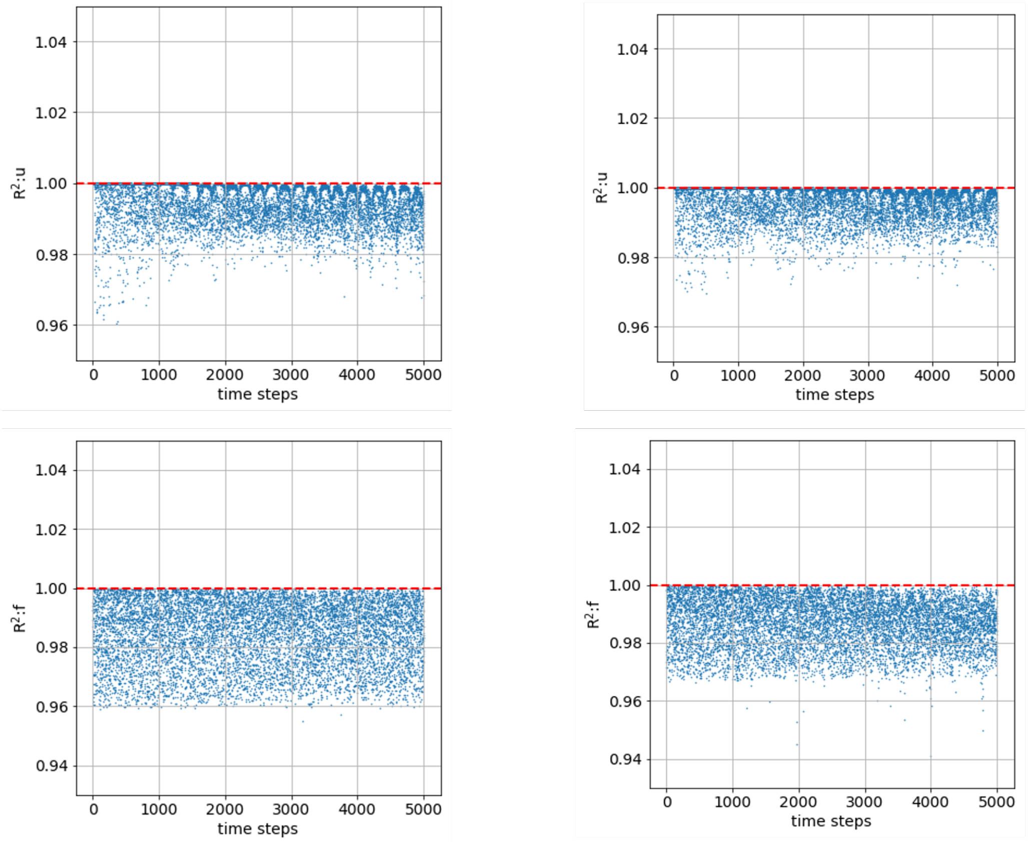
(b)

(c)

(d)

Figure 3.32: Validation of computed displacements and predicted forces for (a) Point A with low excitation amplitude, (b) Point B with low excitation amplitude, (c) Point A with high excitation amplitude and (d) Point B with high excitation amplitude

To conclude this section, it is important to emphasize on the benefits of the proposed method. Our method not only addresses an existing gap that exists to simulate the dynamic behavior of nonlinear unit cells in a computationally efficient manner, it also provides for a model which is faster by almost three orders of magnitude (750x speedup) when analyzing large finite nonlinear structures. Thus the proposed approach provides for a simplistic and drastically efficient approach to analyze problems where alternative homogenization techniques fail due to the continuously evolving stiffness matrix (such as for the case discussed in this section).



(a) (b)

Figure 3.33: Cumulative R^2 scores over simulation time for all global nodes for computed displacements (top) and predicted forces (bottom) using (a) low excitation amplitude, (b) high excitation amplitude

3.4.4 Bandgap studies for 2-D phononic crystal

Extensive research efforts have been devoted to analysis and design of periodic/aperiodic structures and metamaterials for acoustic wave management. Such structures have interesting characteristics such as but not limited to bandgaps [111, 112, 120] and response directionality [121]. The dynamic homogenization approach discussed in this chapter provides for an efficient way to study such phenomena by developing data-driven unit cells. We exemplify this capability of the proposed approach through this example.

Problem Definition: 2-D phononic crystal

The unit cell configuration considered in this example is characterized by bandgaps generated by the periodic modulation of their acoustic impedance by introducing an array of inclusions within a matrix. These periodic modulations essentially cause the scattering of waves at wavelengths that are of the order of the characteristic size of the unit cell, which thereby enables bandgap formations through Bragg scattering [122].

We consider a 2-D square unit cell as shown in Fig. 3.34 with an edge length (L) of 1m and a circular inclusion at its centre of radius (R) = 0.25. The matrix material is isotropic and resembles Aluminum with a modulus of elasticity (E) of 69GPa, Poissons ratio (ν) of 0.33, and density ρ of 2700kg/m^3. The material properties used for the inclusions are theoretical [123] and have an E and ρ which are ten times higher than the Aluminum matrix.

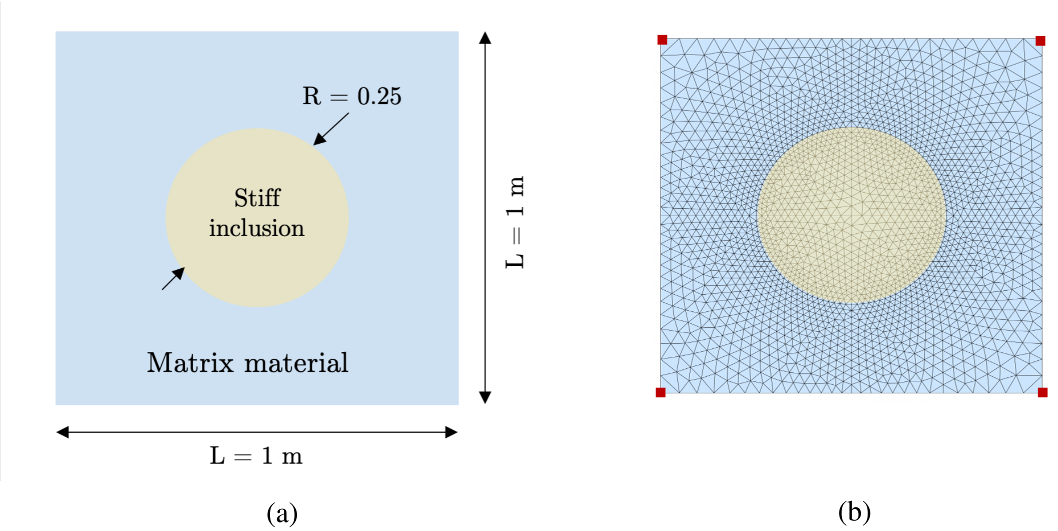


Figure 3.34: Case definition for 2-D phononic crystal: (a) unit cell schematic and (b) Local mesh of unit cell with global nodes (red)

To identify the range of the unit cell's frequency of operation and correspondingly generate training samples, we evaluate the dispersion properties of a periodic media composed of the considered unit cell. For this, we obtain the band diagram through the application of Bloch's theorem [124, 40] spanning a range of wave vectors of the irreducible Brillouin zone [125] (two vectors representing the adjacent edges of the unit cell and one vector representing the diagonal direction). The band diagram for the considered phononic crystal is shown in Fig. 3.35, wherein the plot shows the relation between the frequency and wavenumber space. It can be clearly observed that a bandgap exists in the frequency range of 1600 to 2500 Hz. Thus, we consider a frequency range of 100 to 4000 Hz for sample generation, to ensure that the unit cell can represent the diverse wave propagation behavior.

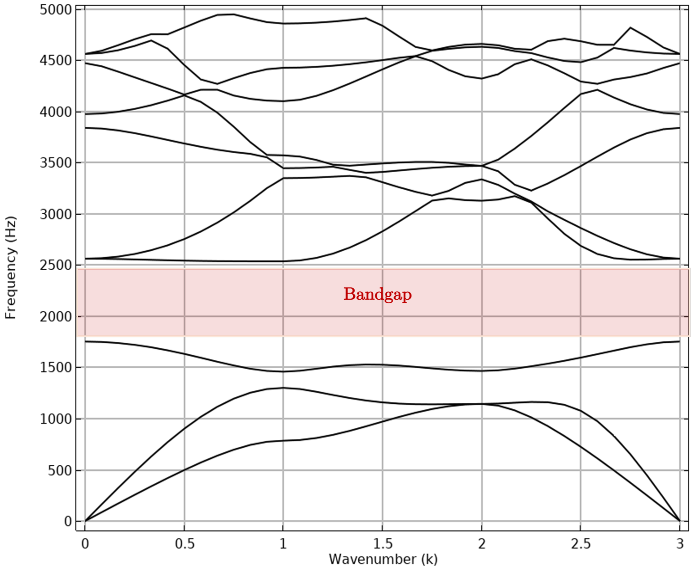
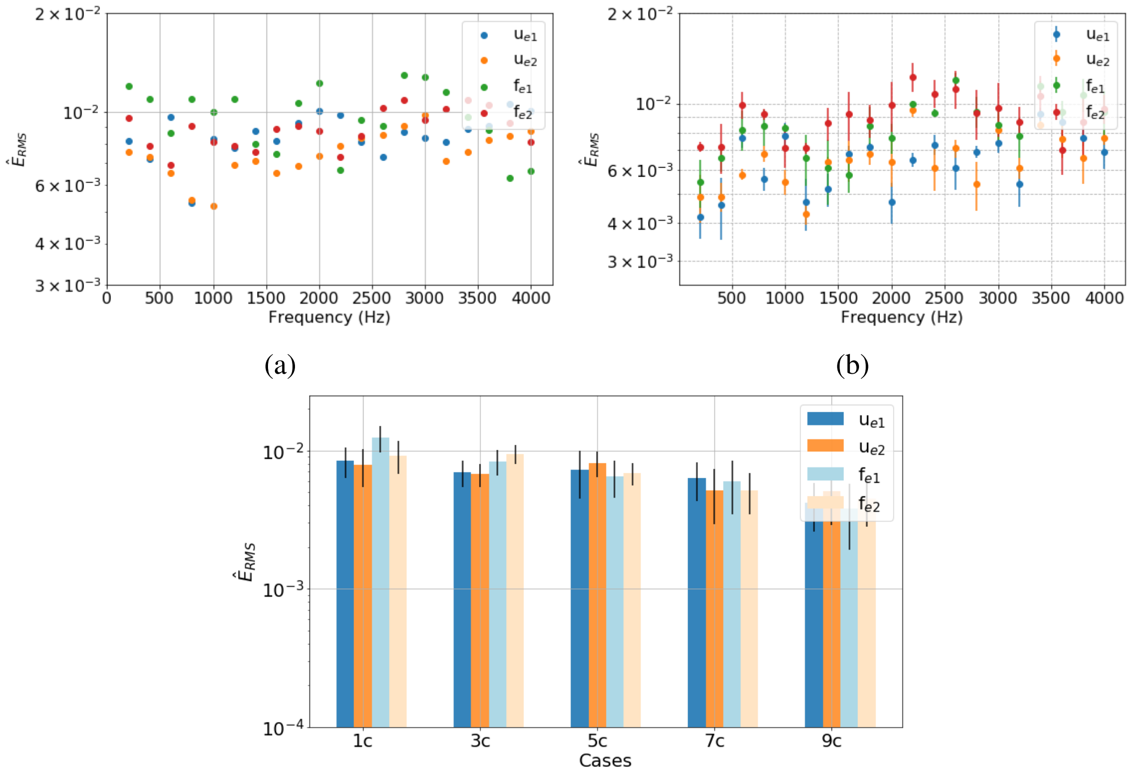


Figure 3.35: Case definition for finite structure with modular unit cells developed separately

Training and results

We define the network to use 8 input features which represent displacements for 4 corner nodes, and the corresponding 8 output features for nodal forces as done for all the other cases in this chapter. We train the model using LSTM layers for history retention and dense layers for local resolution. The dual-step training framework, explained in Sec. 3.3.2 is implemented with 1 LSTM layer and 1 dense layer, each having 600 neurons. To analyze the prediction accuracy of the model, we conduct an extensive study for finite structures simulated with periodicity in both directions. Further, we validate the developed data-driven unit cell for the entire frequency range that is considered, including the region where the bandgap exists. These validation tests are conducted using a similar set-up that is used for the unit cells described earlier in this chapter, details for which are provided in Sec. 3.4.1. Figure 3.36a and 3.36b shows the prediction errors in the form of $\hat{E}_{\text{RMS}}$ (Eq. 3.9) for different nodal degrees of freedom for $1c$ (1×1) and $3c$ (3×3) cases. These arrangements are the ones used for training, and despite being tested against unknown inputs (pure sinusoidal excitation used for validation) are expected to perform good. A rigorous test is performed by considering configurations which are new to the model. We do this by considering arrangements upto 9-cell which resembles a 9×9 periodic structure. Cumulative results are shown in Fig. 3.36c for this test. Overall, we observe that the prediction errors ($\hat{E}_{\text{RMS}}$) remain bounded withing 2% for the predicted forces and 1% for the subsequently computed displacements. Another interesting observation that can be made is that the standard deviation of the error (shown as the error bar in Fig. 3.36c) increases as the number of cells increase. In general, this metric represents the variance of the output parameter over all the nodes and helps in identifying cases if the predictions are extremely good or bad for certain nodes with respect to the others. From this perspective, we can say that the prediction errors get worse as we increase the number of unit cells. However, this is not true for this case. On close observation, we infer that the increase in variance is due to the fact that the wave takes significant time to reach the other end of an extremely large structure. Hence, for much part of the simulation, many nodes have a prediction close to zero and consequently an analogous negligible error, which when considered with the following non-negligible errors, leads to higher variances.

More interestingly, we also analyze an arbitrary finite structure to observe the wave propagation characteristics. For this, we consider a 4×9 arrangement of unit cell shown in Fig. 3.37. We apply a sinusoidal excitation of frequencies ranging from 100 to 4000Hz in the e_1 direction at the left end global nodes of cells $\text{C}_{1,1}$, $\text{C}_{2,1}$, $\text{C}_{3,1}$, and $\text{C}_{4,1}$. To identify how the wave propagates, after every case we note the maximum amplitude of displacement: y_i of vibration obtained at each global node (represented by i) and over the entire duration of simulation. Analogously, we conduct the same set of studies using FE analysis



(a)

(b)

(c)

Figure 3.36: 2-D phononic crystal validation results over entire frequency spectrum for: (a) single cell and (b) 3×3 cell arrangement; (c) Cumulative results for large periodic arrangements

in order to test the accuracy of the prediction provided by the data-driven unit cell.

Results for this study are shown in Fig. 3.38 and Fig. 3.39. First, in Fig. 3.38 we show the normalized error across the frequency spectrum for each global nodes i given by:

$$\mathcal{E}_{amp} = \frac{y_i^a - y_i^p}{y_i^a} \text{ such that } y_i^* = \max|y_i| \tag{3.12}$$

where, y_i^a and y_i^p are the displacement obtained using FE analysis and DD model respectively. We obtain good agreement between the DD model and FE analysis as the errors remain bounded under 4% with an average error of 2.5%. Further, we show the plots from the frequency response study for this case in Fig. 3.39. In these plots, the x-axis represents

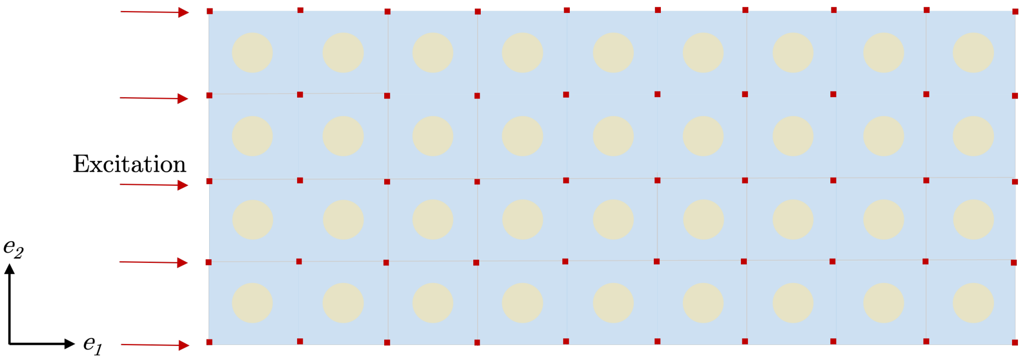


Figure 3.37: Case definition for finite structure to study bandgaps

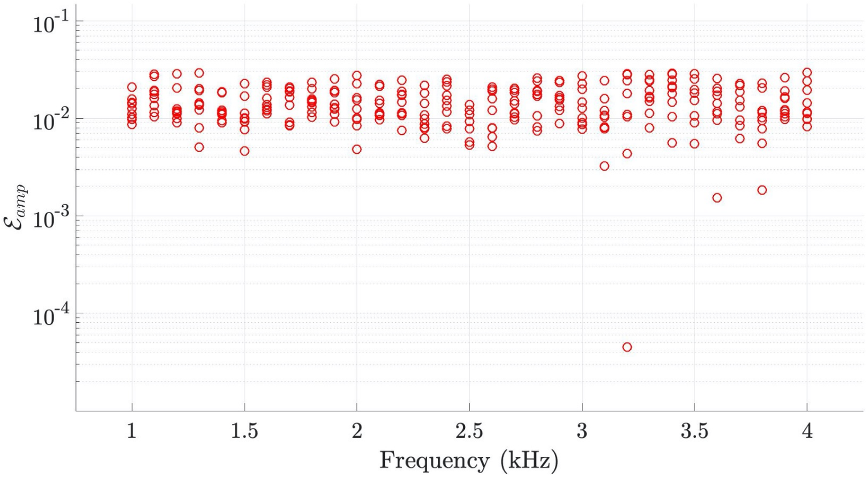
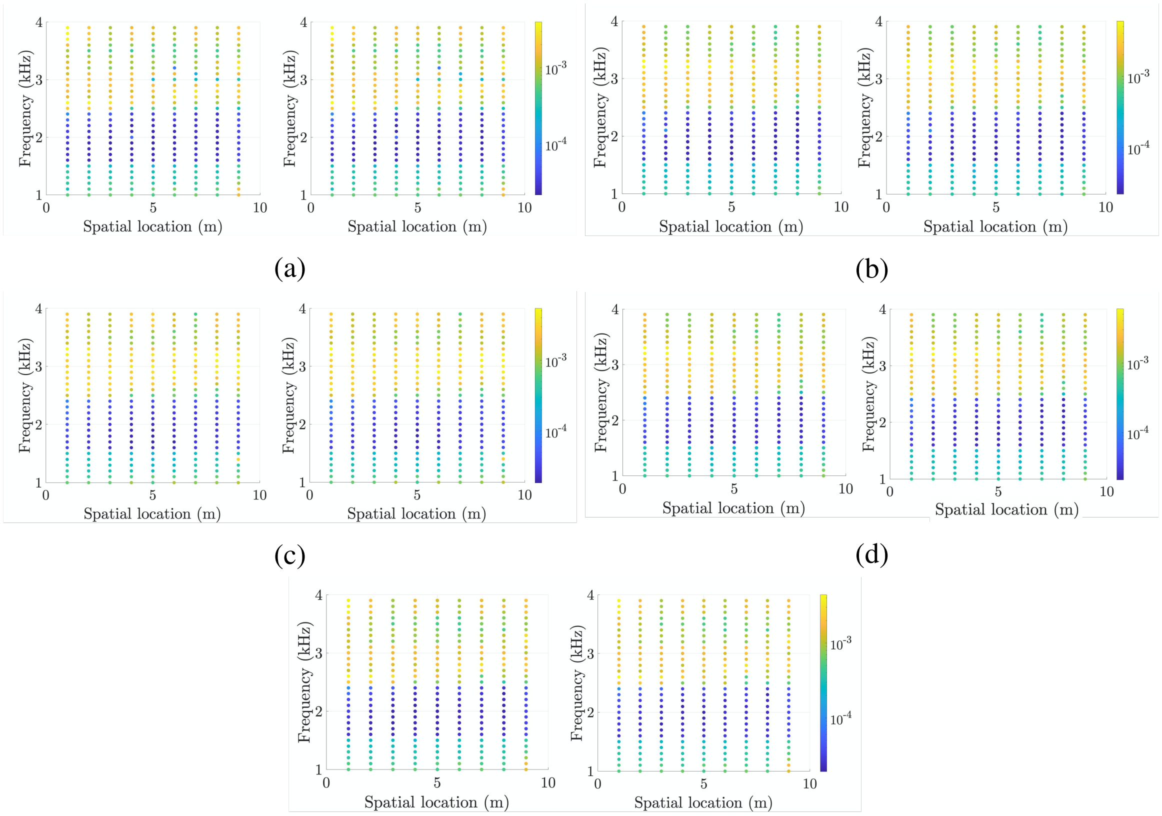


Figure 3.38: Normalized error in displacement amplitude over entire frequency spectrum

the spatial location of the global node with respect the left end nodes where the excitation is applied and the y-axis represents the discrete values of the frequencies used for the excitation. The color of the scatter plot essentially represents the amplitude magnitude of the displacement computed at a particular spatial location and for a particular frequency case. Moreover, in Fig. 3.39 we also show a comparison between FE and DD approaches for $e_2 \in [0, 4m]$. We can clearly observe that the amplitude of displacement or in other words the wave, is significantly (by more than an order) lower for the frequencies in the ranging from 1600 to 2400Hz. This thereby, exhibits that the data-driven unit cell when used in a finite structure is able to correctly identify regions of bandgaps. Moreover, the proposed approach takes less than 200s for each case as compared to 14 hrs taken by FE analysis. Hence, the proposed approach serves to be useful for efficient studies in the areas of vibration guiding/isolation.



(a) (b)

(c) (d)

(e)

Figure 3.39: Validation for frequency response obtained using FE analysis (left) and DD unit cell (right) for finite structure with left end excitation at $e_2 =$ (a) 0m, (b) 1m, (c) 2m, (d) 3m, and (e) 4m

3.5 Parameterized unit cell: More generic unit cell

The examples in the previous section only show the applications of a particular unit cell. For instance, in Sec. 3.4.2, we show the modularity of the approach such that the proposed methodology enables us to use separately developed surrogate unit cells together when simulating a large structure. Though extremely powerful for many applications, such a unit cell cannot be effectively used when for example, the size of the hole is changing. This section addresses this aspect of the problem by parameterizing the unit cell.

For the paramterized unit cell, we assume linear elastic behavior such that the base geometry is the same as that used in Sec. 3.3.1. But, as introduced earlier, we now want to add more generality to the same unit cell. To do so, we consider two parameters, namely rotation of the elliptical hole (θ) and the aspect ratio that is the ratio of the minor to major axis of the elliptical hole (b/a) as show in Fig. 3.40. Thus, such a unit cell enables prediction of the dynamic behavior for different hole configurations.

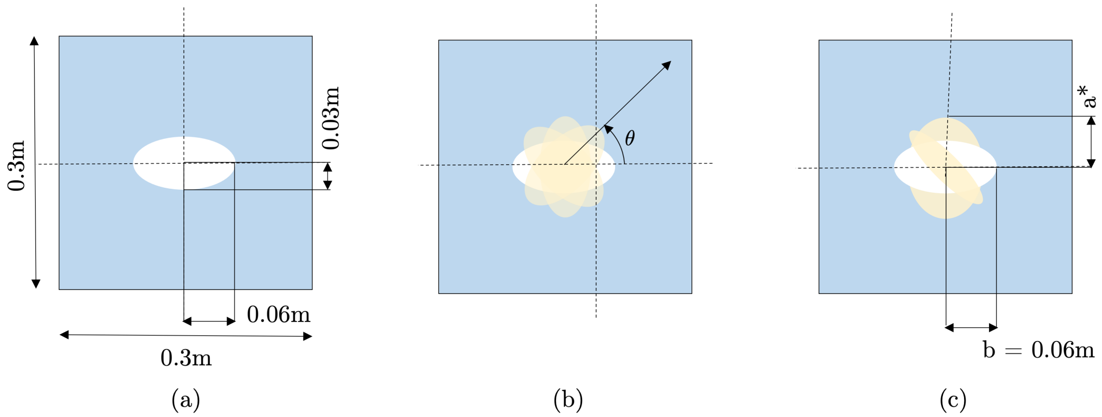
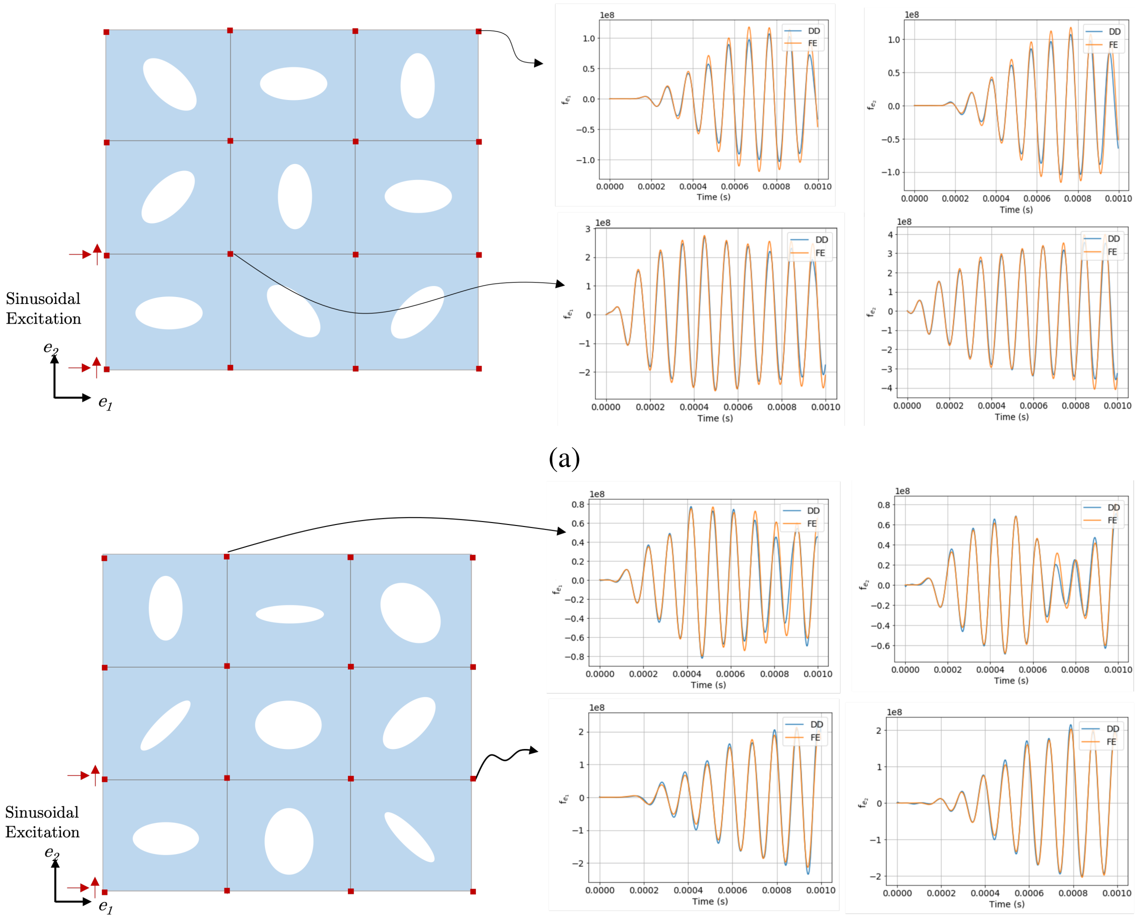


Figure 3.40: Parameterized unit cell geometry (a) Base model (P_0) (b) Parameterized for rotation angle (P_1), and (c) Parameterized for rotation angle and aspect ratio (P_2)

In order to develop a parameterized model, we adopt the wavelet approach and meta learning approach for obtaining input displacements and model training respectively, as described in Sec. 3.3. Further, to analyze the effect of adding parameters to the dynamic homogenization approach we train three models: 1) Base (P_0) model: with $0°$ and $b/a = 0.5$ (same geometry as Sec. 3.3.1), 2) P_1 model: with five rotation angles $(0°, 45°, 90°, 135°, 180°)$ and a fixed b/a ratio of 0.5 used for samples, and 3) P_2 model: with five rotation angles along with three b/a ratios (0.25, 0.5, 0.75). Note that, when we parameterize the model for the rotation of the hole, we also use the $180°$ case. This configuration, though is exactly the same as that of $0°$ in terms of physical behavior, it is needed for training as it helps the model to perform well beyond $\theta = 135°$. Hence, essentially, now with this model, we can use and make predictions for the unit cell with any randomly selected hole rotation angle in the range $[0°, 180°]$. For sample generation we use these different geometry configurations in FE method and obtain the input-output features that is, displacement-force sequences as explained in Sec. 3.3.1. For model training however, we make one change. Instead of having 8 features of inputs (displacements) and 8 features of outputs (forces) as done for the base model in Sec. 3.3.2, we now have 9 input features for P_1 and 10 input features for P_2. This is because, we now have an additional feature resembling the rotation angle in P_1 and two additional features of rotation angle and b/a ratio in P_2. The output features (8 in number) do not require any modifications and remain the same to represent the forces at global nodes of the unit cell.

In order to analyze the prediction accuracy of the parameterized models, we use the validation tests similar to those explained in Sec. 3.4.1 such that sinusoidal excitation is applied in e_1 and e_2 direction at the left global nodes of $C_{1,1}$ for each configuration. We obtain an $\hat{E}_{RMS}$ which is $< 7.3\%$ for forces and is $< 5.5\%$ for the displacements for P_1 and correspondingly $< 10.1\%$ and 9.2% for P_2. These errors are relatively higher than the linear elastic base unit cell discussed in Sec. 3.4.1. But, it is important to emphasize here that surrogate model has to learn much more complex information in the parameterized case. In order to visualize, what these errors actually resemble, we conduct an analysis for a 3×3 configuration with a random arrangement of holes with different rotations in Fig. 3.41a and with different rotations and b/a ratios in Fig. 3.41b. The predictions for these cases are done by surrogate unit cell models P_1 and P_2 respectively. Figure 3.41 also shows the comparison of force predictions in e_1 and e_2 directions for randomly selected global nodes. Overall we obtain a good agreement in the prediction trends as the models are able to capture the dynamic behavior over the entire time of the simulation with an E_{RMS} of 7.01$\%$ for forces and 6.42$\%$ for displacements for model P_1. The corresponding values for P_2 model are 10.45$\%$ and 8.77$\%$.

We also study the model architecture sensitivity for this study to analyze the effect of

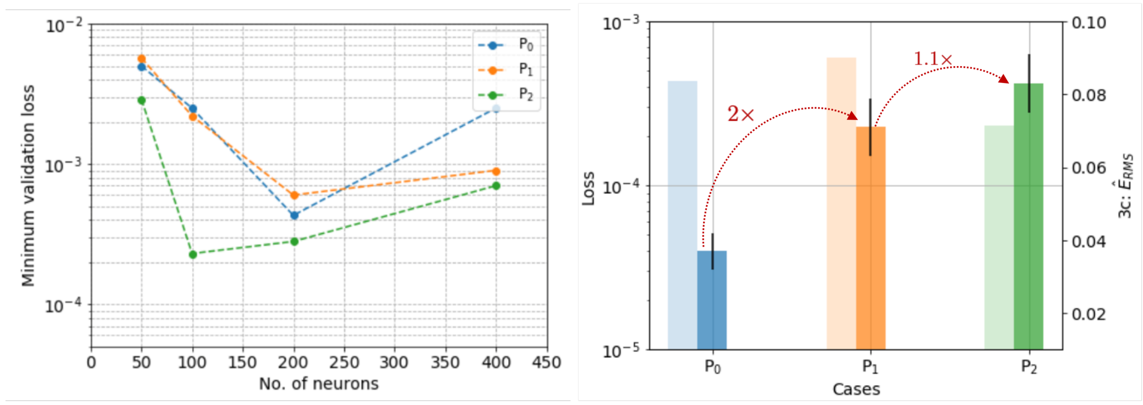


(a)

(b)

Figure 3.41: Test samples with random configuration showing prediction quality for (a) P_1 model and P_2 model

adding parameters to the model. We perform this study in the same manner as done for the base case in Sec. 3.3.2 along with an early stopping patience parameter of 20 to avoid overfitting. Also, note that, since we obtained the best validation loss for the L_1D_1 case (1 LSTM layer and 1 Dense layer) in Sec. 3.3.2, we use the same for training the parameterized models. However, we observe sensitivity of the model by varying the neurons in each layer. The results of the sensitivity study are shown in Fig 3.42 which enables us to make helpful observations. Firstly, we observe that in general, an increase in the number of neurons in each layer improves the validation loss. This is typical, because less number of neurons implies less number of IPs that are available to represent the input-output

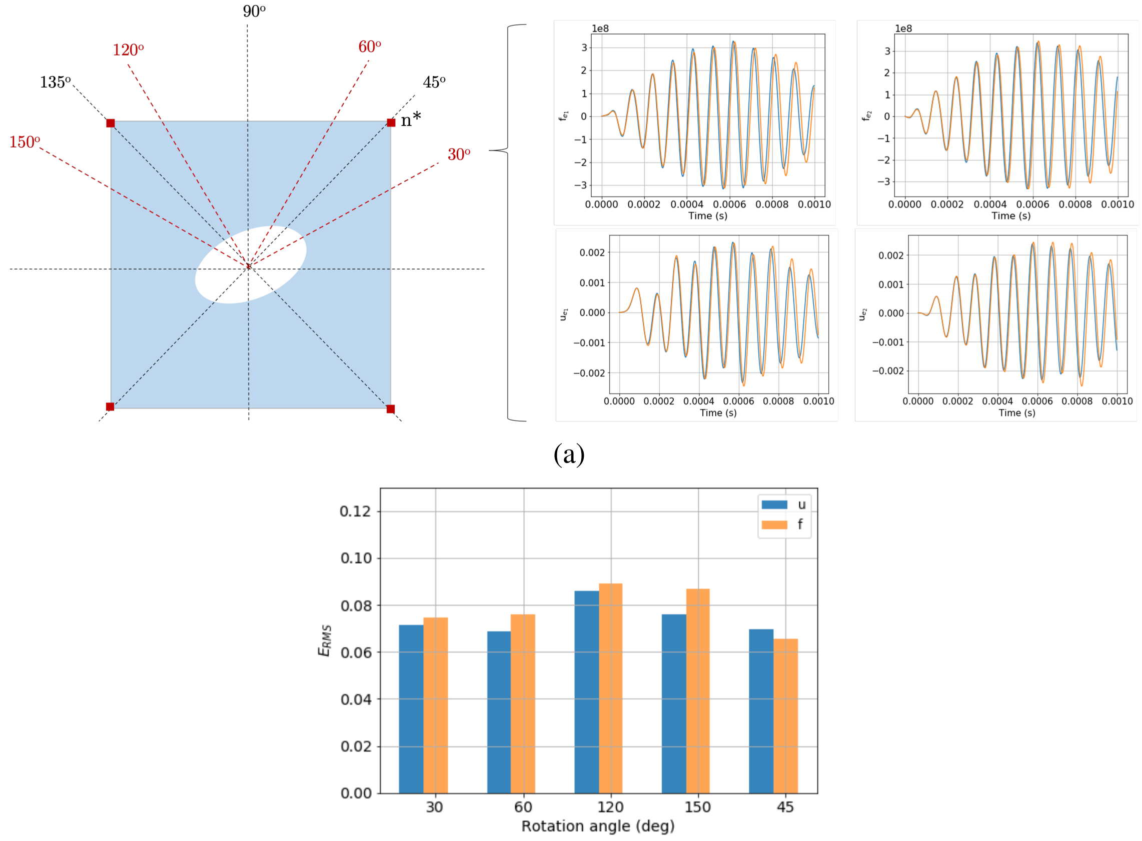


(a)

(b)

Figure 3.42: Architecture studies for parameterized unit cells (a) Validation loss versus number of neurons and (b) Averaged prediction errors for P_0, P_1, and P_2.

mapping. And as we keep increasing the neurons, we start to observe an increase in validation loss. This happens due to the fact that a more complex architecture usually needs more iterations for improvement or that the framework detected overfitting due to which training was interrupted. Thus, essentially we observe a bucket formation for the loss due to this decrease and subsequent increase in validation loss. Perhaps more interestingly, we observe (Fig. 3.42a) that this loss bucket tends to move left from P_0 model to P_1 and then to P_2. This basically implies that the neurons requirement is decreasing as we add parameters. This is counter-intuitive. However, we attribute this to the fact that since the number of samples significantly increase from P_0 model to P_2 model (12 times), the network is able to perform better inspite of the presence of the parameter effect. Hence, we can reasonably establish that a trade-off exists between the parameter effect on the results versus the samples needed to capture the effect. A deeper study into this aspect can potentially aid in parameterizing the unit cell in multiple ways in an effective manner. Finally, we observe that the best validation loss obtained for all models are in a similar order of $\mathcal{O}(10^{-4})$. But it is important to analyse whether this observation translated to prediction accuracy. We do this in Fig. 3.42b which shows the best validation loss using the left y-axis and $\hat{E}_{RMS}$ for an unseen test sample with a 3×3 configuration of the base unit cell. We can see that the prediction errors increase with the addition of parameters as even though the sequence trend may be well captured by the model, the point-to-point accuracy may differ. That said, we observe that the prediction errors become twice (2×) in P_1 when compared to the base unit cell P_0. But, subsequently the errors show a trend to saturate as adding the next parameter only leads to 1.1× increase in prediction errors. Thus overall, these statistical studies enable many possible opportunities to parameterize the unit cell without sacrificing much accuracy.



(a)

(b)

Figure 3.43: Prediction accuracy for parameterized model for unseen rotations: (a) Forces and displacement comparison with FE results for node n* in 30° hole configuration and (b) Cumulative results for multiple unseen angles compared with a seen angle $45°$.

As a final example, we also analyze whether the trained unit cell is able to capture behavior of parameter values of a particular parameter for which it is not specifically trained. For this, we use the trained P_1 model, and check its accuracy for a $1{\times}1$ configuration for $\theta = 30°, 60°, 120°,$ and $150°$. Note that, during training we provided samples for

$\theta = 0°, 45°, 90°, 135°, 180°$ and hence, the considered angles for this example are unseen by the model before. We show time domain results for $30°$ for the forces and displacements at the global node n* of the unit cell in Fig. 3.43a. The results show good prediction accuracy for the forces and give an E_{RMS} of 7.12$\%$ and 7.41$\%$ for forces and displacements respectively. Further, we also show E_{RMS} values obtained for all the other considered unseen angles in Fig. 3.43b and also show the errors for the $45°$ (seen rotation angle by the model). We can see that the errors are in the same range as the $45°$ case and do no show any specific trends for the unseen rotation angles. This thereby establishes, that this model can predict well for all angles of rotation for the hole. Therefore, the model serves as a very important application of the presented methodology, as now we can train a single model with different geometric parameters, which not only perform well at certain specific values of that parameter, but also for all the unseen parameter values across the seen parameter value range.

CHAPTER IV

CONCLUSION

4.1 Summary and conclusions

Using ML techniques, a framework to develop dimensionally-reduced surrogate models, or *smart parts* (SPs), with the objective to establish a direct relationship between the input-output parameters of a component exhibiting nonlinearities and history-dependent behavior is established through this thesis. This can be done since, for most cases, the behavior of a component in an assembly can be represented by simple nonlinear relations at special locations such as interface nodes and appropriate kinematic constraints. Moreover, we use high-fidelity finite element data to train an ANN, thus ensuring a well-represented domain and the reliability of data used.

First, we defined and discussed the smart parts approach. Specifically, we demonstrated its implementation and performance via two exemplary engineering applications. In the first one, we simulated the dynamic response of a helicopter pitch link which develops localized damage in the form of localized plasticity. In the second one, we developed a SP to simulate the quasistatic operation of a lap-joint including contact nonlinearity. We also implemented an assembly of two lap-joints with appropriate constraints to illustrate the procedure of systematically integrating a SP into larger assemblies. For both cases, we also provide means to monitor localized parameters such as internal damage, which is crucial in predicting or identifying structural failure and is usually lost in traditional model order reduction techniques. We assessed the performance of a SP by comparison with results obtained from high-fidelity simulations. Our findings indicate that the SPs produce predictions without sacrificing much accuracy. Finally, we quantified the computational benefit associated with this approach and observed that SPs are able to produce predictions in a few seconds as compared to a few hours required by traditional methods.

We also proposed ways to improve the efficiency and performance of the developed SP. First, we trained the SP using a corotational reference frame by removing rigid body displacements. This guarantees the SP to be frame indifferent and hence can be used in different orientations without retraining it. Then we enforced conservation of linear and angular momentum on the SP to avoid violation of physical laws by using equilibrium conditions on the component and reducing the computational error. Depending on the application different types of physical constraints such as conservation of energy may be applied to improve the accuracy of a model. Second, for cases when the model is operational for several hours, we enhanced the model by using a bi-network approach. This employed a data-driven switch model that decides when to trigger the following damage evolution model, thereby reducing the prediction error and computational time.

We then extended the smart parts approach to study dynamic (wave propagation) problems. For this, we proposed a method for predicting the dynamic response of large arbitrary heterogeneous structures that leverages state-of-the-art machine learning techniques to develop a surrogate model for a unit cell and effectively homogenize it. Such a surrogate model represents the global behavior of the unit cell while preserving the local information and/or the effect of heterogeneities on the global behavior. To generate our model, we proposed a novel training scheme inspired from meta-learning that expands the operational frequency spectrum and improves the prediction resolution of the unit cell by using information pertaining to tangent stiffness of the unit cell during training. We made no assumptions about the type of material or heterogeneities in the unit cell as demonstrated through multiple examples. We also illustrated how to implement a surrogate unit cell in large arbitrary structures to study wave propagation phenomena. We demonstrated our method's ability by comparing its results with those obtained via high-fidelity finite element simulations. Overall, we established that using surrogate unit cells can dramatically boost the computational efficiency and simplicity to analyze dynamic behavior of large structures.

We also showed how the developed smart parts can be used to enable further studies and applications. For this, we firstly used the smart part of the helicopter pitch link and conducted localized damage sensitivity investigations. Using these studies, we proposed a novel metric: the integrity ratio that can be used to improve existing life extending control schemes and guide the development of future ones. Second, we showed that bandgap prediction and geometric optimization can be performed in an efficient manner by developing surrogate models that represent a unit cell's dynamic behavior. And finally, we showed that the dynamic homogenization method can be extended to parameterize the unit cell too represent a range of geometric parameters such as the rotation angle of the elliptical hole and the aspect ratio of the hole.

In summary, our results show great promise in using machine learning techniques for model order reduction as applied to mechanical parts and assemblies. Smart parts show a good compromise of computational efficiency and accuracy, which can be further enhanced when knowledge about physics can be directly enforced instead of requiring the neural networks to learn them.

4.2 Contributions

The research presented in this thesis provides the following contributions to the state-of-the-art:

1. A novel approach to represent the structural behavior of a component that exhibits nonlinear or history-dependent behavior using advanced machine learning techniques. Such an approach enables real-time estimations of interested parameters. This has not been addressed by any modeling approach up to date.

2. An approach to represent dynamic behavior of large heterogeneous structures by developing surrogate model for a single unit cell. This approach is versatile and can be used for different types of unit cells as shown for: linear, nonlinear (hyperelastic), homogeneous, and heterogeneous unit cells.

3. A framework to simulate mechanical assemblies in an efficient manner. This can be done for nonlinear and history-dependent problems with the corresponding smart parts. Moreover, dynamic or wave propagation problems can also be simulated for large arbitrary finite structures with modular unit cells. That is, the unit cells are trained separately and then used in conjunction to study the dynamic behavior of arbitrary structures.

4. A novel training methodology that is inspired from meta learning and uses a custom loss function and transfer learning for dynamic problems. This method helps the unit cell learn effectively in situations when a broad frequency spectrum of inputs need to be included by using tangent stiffness in the loss function definition and a dual-step training process.

5. Numerous enhancements are presented to be used in conjunction with the developed smart part. Such enhancements help in improving the proposed approach in two ways:

    (a) Improve the accuracy and precision of the model:

        i. Introduced conservation principles on predicted forces in the implementation algorithm

        ii. Developed a bi-network approach

    (a) Improve the capability and application domain of the model by:

        i. Developed a wavelet-based method for obtaining low noise-signal samples and established the minimum number of cells that are needed in training for general representation of a unit cell

   ii. Extended corotational formulation to induce frame indifference for 3D components

6. Relevant engineering implementations of smart parts to enable and expand the scope of further studies:

   (a) Established a novel metric, the integrity ratio to enable less conservative strategies for extending the operational life of helicopter components.

   (b) Predicted bandgaps and conducted vibration attenuation studies for geometric optimization and/or vibration management in large structures.

   (c) Parameterize the surrogate unit cell to make the unit cell more generic in terms of geometric parameters.

4.3 Future Directions

There are several interesting research avenues, at multiple fronts, that can be pursued in the future. Some of the key areas related to this thesis are:

1. Despite the enforcement of conservation of linear and angular momentum during predictions, there might still be certain physical laws that may get violated depending on the application. Thus, future research in this direction to study how network architecture could be exploited to satisfy further constraints can be valuable.

2. A deeper investigation into smart parts/ unit cells which transcend a domain of geometries or materials, thereby making the smart part fully/ partially independent of these parameters by leveraging the dependencies of different parameters on training statistics and prediction accuracy.

3. Investigating and characterizing the stochastic nature of network architecture for different types of structural analysis based on the mathematical representation of the

training features.

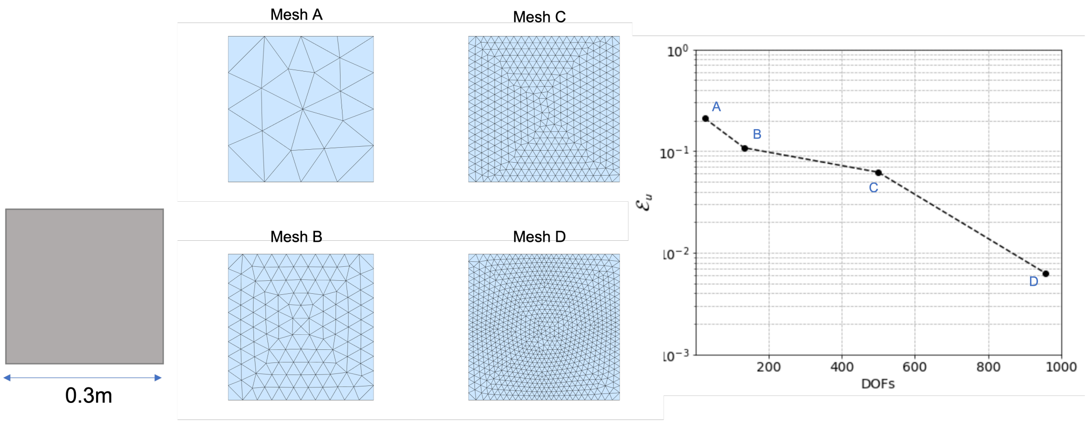
4. Investigating the use of the smart parts approach for multiphysics applications to obtain approximate but deterministic relations for coupled models.

Appendices

APPENDIX A

SURROGATE MODEL FOR HOMOGENEOUS 2D UNIT CELL

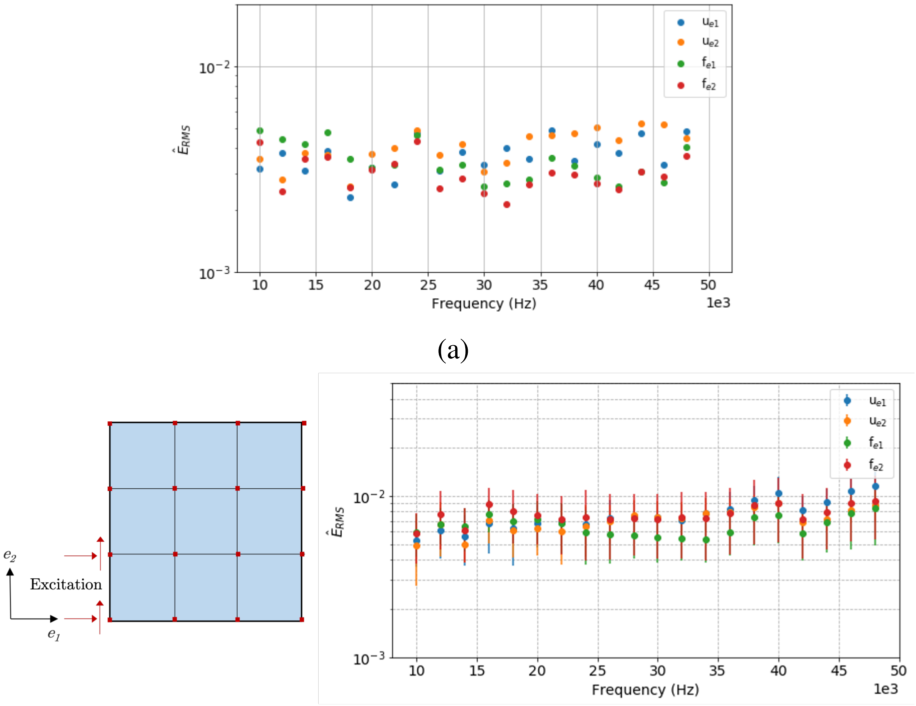
In this section, we provide the details and validation studies pertaining to the 2D unit cell without hole. We use the same geometric description and material properties as that used for the heterogeneous unit cell in Sec. 3.3.1, but remove the hole from the unit cell. However, since this unit cell does not include a hole, we obtain converged results with relatively lesser number of elements. We obtain the finalized mesh by conducting numerical analysis for different local dicretizations and consequently computing the error norm given by Eqn. 3.2 as shown in Fig. 1.1. For this unit cell, we finalize *Mesh D* which has 1012 nodes and an $\mathcal{E}_{\mathcal{U}} = 0.1\%$.



(a) (b)

Figure 1.1: Model development for homogeneous 2D square unit cell (a) Unit cell schematic (b) Mesh convergence for unit cell

Further, we use the same input displacements as those used in Sec. 3.3.1 to generate the training samples. However, it is important to note that this is not a requirement for the proposed approach and a different set of input displacements could also be used. That said, since we want to use this unit cell in conjunction with the previously developed heterogeneous unit cell, we must ensure that there is atleast some overlap between their respective frequency spectrums. This is because the operational frequency range for a unit cell is determined based on the training samples used to develop it. Finally, we adopt the same procedure for training this unit cell as discussed in Sec. 3.3.2 and obtain a model using L_1D_1 architecture and 100 neurons with the best performance (minimum validation loss).



(a)

(b)

Figure 1.2: Validation results for homogeneous unit cell (a) single cell over entire frequency and (b) 3 × 3 periodic structure over entire frequency spectrum with problem set-up (left) and displacement and force errors over entire frequency spectrum (right)

Once trained, we conduct a validation study to determine the accuracy of the developed unit cell. We use a $1c$ (1×1) and $3c$ (3×3) configuration for validation. To remain consistent, we use the same problem set-up for validation as used for preliminary validation in Sec. 3.4.1 such that the left end global nodes of $C_{1,1}$ are excited in directions e_1 and e_2. We show the cumulative results for $1c$ case in Fig. 1.2a and $3c$ case in Fig. 1.2b We observe that the unit cell is able to perform well across the entire frequency spectrum and for different loading conditions. We obtain errors which are below 1% for all cases, indicating good accuracy. In fact, the errors are in the same range as that obtained for the heterogeneous unit cell (Sec. 3.4.1). Overall, such an accurate model enables its use with other unit cells and hence for numerous applications.

BIBLIOGRAPHY

[1] A. Quarteroni, G. Rozza, *et al.*, *Reduced order methods for modeling and computational reduction*. Springer, 2014, vol. 9.

[2] R. J. Guyan, “Reduction of stiffness and mass matrices,” *AIAA journal*, vol. 3, no. 2, pp. 380–380, 1965.

[3] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations: an introduction*. Springer, 2015, vol. 92.

[4] M. Ohlberger and S. Rave, “Reduced basis methods: Success, limitations and future challenges,” *arXiv preprint arXiv:1511.02021*, 2015.

[5] J. S. Hesthaven, G. Rozza, B. Stamm, *et al.*, *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016, vol. 590.

[6] W. Yu, D. H. Hodges, and J. C. Ho, “Variational asymptotic beam sectional analysis–an updated version,” *International Journal of Engineering Science*, vol. 59, pp. 40–64, 2012.

[7] A. K. Noor, “Global-local methodologies and their application to nonlinear analysis,” *Finite Elements in Analysis and Design*, vol. 2, no. 4, pp. 333–346, 1986.

[8] I. Babuška and J. M. Melenk, “The partition of unity method,” *International journal for numerical methods in engineering*, vol. 40, no. 4, pp. 727–758, 1997.

[9] A. K. Noor and J. M. Peters, “Reduced basis technique for nonlinear analysis of structures,” *AIAA journal*, vol. 18, no. 4, pp. 455–462, 1980.

[10] A. Imazatène, J. M. Cadou, H. Zahrouni, and M. Potier-Ferry, “A new reduced basis method for non-linear problems,” *Revue Européenne des Eléments Finis*, vol. 10, no. 1, pp. 55–76, 2001.

[11] B. Haasdonk, M. Dihlmann, and M. Ohlberger, “A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space,” *Mathematical and Computer Modelling of Dynamical Systems*, vol. 17, no. 4, pp. 423–442, 2011.

[12] M. Dihlmann, M. Drohmann, and B. Haasdonk, “Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning,” *Proc. of ADMOS*, vol. 2011, p. 64, 2011.

[13] K. Carlberg, “Adaptive h-refinement for reduced-order models,” *International Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1192–1210, 2015.

[14] K. Lu *et al.*, “A review of model order reduction methods for large-scale structure systems,” *Shock and Vibration*, vol. 2021, 2021.

[15] I. Babuška, “Homogenization and its application. mathematical and computational problems,” in *Numerical solution of partial differential equations–III*, Elsevier, 1976, pp. 89–116.

[16] G Papanicolau, A Bensoussan, and J.-L. Lions, *Asymptotic analysis for periodic structures*. Elsevier, 1978.

[17] J. Guedes and N. Kikuchi, “Preprocessing and postprocessing for materials based on the homogenization method with adaptive finite element methods,” *Computer methods in applied mechanics and engineering*, vol. 83, no. 2, pp. 143–198, 1990.

[18] F. J. Vernerey and M. Kabiri, “An adaptive concurrent multiscale method for microstructured elastic solids,” *Computer Methods in Applied Mechanics and Engineering*, vol. 241, pp. 52–64, 2012.

[19] Z.-Q. Qu, *Model Order Reduction Techniques with Applications in Finite Element Analysis: With Applications in Finite Element Analysis*. Springer Science & Business Media, 2004.

[20] M. Paz, "Dynamic condensation," *AIAA journal*, vol. 22, no. 5, pp. 724–727, 1984.

[21] N Bouhaddi and R Fillod, "Model reduction by a simplified variant of dynamic condensation," *Journal of Sound and Vibration*, vol. 191, no. 2, pp. 233–250, 1996.

[22] S.-H. Boo and P.-S. Lee, "A dynamic condensation method using algebraic substructuring," *International Journal for Numerical Methods in Engineering*, vol. 109, no. 12, pp. 1701–1720, 2017.

[23] J. O'CALLAHAN, "A procedure for an improved reduced system (irs) model," *Proceedings of 7th IMAC, Las Vegas, NV., 1989*, 1989.

[24] M. Friswell, S. Garvey, and J. Penny, "Model reduction using dynamic and iterated irs techniques," *Journal of sound and vibration*, vol. 186, no. 2, pp. 311–323, 1995.

[25] K. Pham, V. G. Kouznetsova, and M. G. Geers, "Transient computational homogenization for heterogeneous materials under dynamic excitation," *Journal of the Mechanics and Physics of Solids*, vol. 61, no. 11, pp. 2125–2146, 2013.

[26] L. Liu, A Sridhar, M. Geers, and V. Kouznetsova, "Computational homogenization of locally resonant acoustic metamaterial panels towards enriched continuum beam/shell structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 387, p. 114 161, 2021.

[27] T. F. van Nuland, P. B. Silva, A. Sridhar, M. G. Geers, and V. G. Kouznetsova, "Transient analysis of nonlinear locally resonant metamaterials via computational homogenization," *Mathematics and Mechanics of Solids*, vol. 24, no. 10, pp. 3136–3155, 2019.

[28] F Casadei, J. Rimoli, and M Ruzzene, “A geometric multiscale finite element method for the dynamic analysis of heterogeneous solids,” *Computer Methods in Applied Mechanics and Engineering*, vol. 263, pp. 56–70, 2013.

[29] F Casadei, J. Rimoli, and M Ruzzene, “Multiscale finite element analysis of wave propagation in periodic solids,” *Finite Elements in Analysis and Design*, vol. 108, pp. 81–95, 2016.

[30] G. Capuano, M. Ruzzene, and J. J. Rimoli, “Modal-based finite elements for efficient wave propagation analysis,” *Finite Elements in Analysis and Design*, vol. 145, pp. 10–19, 2018.

[31] W. C. Hurty, “Dynamic analysis of structural systems using component modes,” *AIAA journal*, vol. 3, no. 4, pp. 678–685, 1965.

[32] S Gopalakrishnan, A Chakraborty, and D. R. Mahapatra, “Spectral finite element method for active wave control,” *Spectral Finite Element Method: Wave Propagation, Diagnostics and Control in Anisotropic and Inhomogeneous Structures*, pp. 365–422, 2008.

[33] R. E. Nickell, “Nonlinear dynamics by mode superposition,” *Computer Methods in Applied Mechanics and Engineering*, vol. 7, no. 1, pp. 107–129, 1976.

[34] N. F. Morris, “The use of modal superposition in nonlinear dynamics,” *Computers & Structures*, vol. 7, no. 1, pp. 65–72, 1977.

[35] S. Remseth, “Nonlinear static and dynamic analysis of framed structures,” *Computers & Structures*, vol. 10, no. 6, pp. 879–897, 1979.

[36] S. R. Idelsohn and A. Cardona, “A reduction method for nonlinear structural dynamic analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 49, no. 3, pp. 253–279, 1985.

[37] V. Shah, G. Bohm, and A. Nahavandi, “Modal superposition method for computationally economical nonlinear structural analysis,” 1979.

[38] K.-J. Bathe and S. Gracewski, “On nonlinear dynamic analysis using substructuring and mode superposition,” *Computers & Structures*, vol. 13, no. 5-6, pp. 699–707, 1981.

[39] Y. Xia and R. Lin, “A new iterative order reduction (ior) method for eigensolutions of large structures,” *International Journal for Numerical Methods in Engineering*, vol. 59, no. 1, pp. 153–172, 2004.

[40] R. M. Orris and M Petyt, “A finite element study of harmonic wave propagation in periodic structures,” *Journal of Sound and Vibration*, vol. 33, no. 2, pp. 223–236, 1974.

[41] H. Adeli, “Neural networks in civil engineering: 1989–2000,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 16, no. 2, pp. 126–142, 2001.

[42] G. Yagawa and H Okuda, “Neural networks in computational mechanics,” *Archives of Computational Methods in Engineering*, vol. 3, no. 4, p. 435, 1996.

[43] F. E. Bock, R. C. Aydin, C. J. Cyron, N. Huber, S. R. Kalidindi, and B. Klusemann, “A review of the application of machine learning and data mining approaches in continuum materials mechanics,” *Frontiers in Materials*, vol. 6, p. 110, 2019.

[44] R. Ceravolo, A. De Stefano, and D. Sabia, “Hierarchical use of neural techniques in structural damage recognition,” *Smart Materials and Structures*, vol. 4, no. 4, p. 270, 1995.

[45] B. Kurian and R. Liyanapathirana, “Machine learning techniques for structural health monitoring,” in *Proceedings of the 13th International Conference on Damage Assessment of Structures*, Springer, 2020, pp. 3–24.

[46] I. Flood and N. Kartam, “Neural networks in civil engineering. i: Principles and understanding,” *Journal of computing in civil engineering*, vol. 8, no. 2, pp. 131–148, 1994.

[47] M Takadoya, J. Achenbach, Q. Guo, and M Kitahara, “Crack parameter characterization by a neural network,” in *Review of Progress in Quantitative Nondestructive Evaluation*, Springer, 1996, pp. 797–804.

[48] G. Stavroulakis and H Antes, “Neural crack identification in steady state elastodynamics,” *Computer methods in applied mechanics and engineering*, vol. 165, no. 1-4, pp. 129–146, 1998.

[49] Y. Xu, G. Liu, Z. Wu, and X. Huang, “Adaptive multilayer perceptron networks for detection of cracks in anisotropic laminated plates,” *International journal of solids and structures*, vol. 38, no. 32-33, pp. 5625–5645, 2001.

[50] S.-W. Liu, J. H. Huang, J.-C. Sung, and C. Lee, “Detection of cracks using neural networks and computational mechanics,” *Computer methods in applied mechanics and engineering*, vol. 191, no. 25-26, pp. 2831–2845, 2002.

[51] N. Mera, L Elliott, and D. Ingham, “The use of neural network approximation models to speed up the optimisation process in electrical impedance tomography,” *Computer methods in applied mechanics and engineering*, vol. 197, no. 1-4, pp. 103–114, 2007.

[52] M. Papadrakakis and N. D. Lagaros, “Reliability-based structural optimization using neural networks and monte carlo simulation,” *Computer methods in applied mechanics and engineering*, vol. 191, no. 32, pp. 3491–3507, 2002.

[53] D Fernández-Fdz and R Zaera, “A new tool based on artificial neural networks for the design of lightweight ceramic–metal armour against high-velocity impact of solids,” *International Journal of Solids and Structures*, vol. 45, no. 25-26, pp. 6369–6383, 2008.

[54] T. D. Dinh, A. Rezaei, T Linthout, M Mollaert, D Van Hemelrijck, and W. Van Paepegem, “A computational compensation method for fabric panels of tensioned membrane structures using a shape optimization method based on gradientless al-

gorithms," *International Journal of Solids and Structures*, vol. 112, pp. 16–24, 2017.

[55] G. X. Gu, C.-T. Chen, and M. J. Buehler, "De novo composite design based on machine learning algorithm," *Extreme Mechanics Letters*, vol. 18, pp. 19–28, 2018.

[56] M. Bessa and S Pellegrino, "Design of ultra-thin shell structures in the stochastic post-buckling range using bayesian machine learning and optimization," *International Journal of Solids and Structures*, vol. 139, pp. 174–188, 2018.

[57] X. Lei, C. Liu, Z. Du, W. Zhang, and X. Guo, "Machine learning-driven real-time topology optimization under moving morphable component-based framework," *Journal of Applied Mechanics*, vol. 86, no. 1, 2019.

[58] P. Deshpande, B. Gautham, A Cecen, S Kalidindi, A. Agrawal, and A Choudhary, "Application of statistical and machine learning techniques for correlating properties to composition and manufacturing processes of steels," in *Proceedings of the 2nd world congress on integrated computational materials engineering (ICME)*, Springer, 2013, pp. 155–160.

[59] B. L. DeCost and E. A. Holm, "A computer vision approach for automated analysis and classification of microstructural image data," *Computational materials science*, vol. 110, pp. 126–133, 2015.

[60] J. A. Gomberg, A. J. Medford, and S. R. Kalidindi, "Extracting knowledge from molecular mechanics simulations of grain boundaries using machine learning," *Acta Materialia*, vol. 133, pp. 100–108, 2017.

[61] V. Sundararaghavan and N. Zabaras, "Classification and reconstruction of three-dimensional microstructures using support vector machines," *Computational Materials Science*, vol. 32, no. 2, pp. 223–239, 2005.

[62] R. Hambli, H. Katerchi, and C.-L. Benhamou, “Multiscale methodology for bone remodelling simulation using coupled finite element and neural network computation,” *Biomechanics and modeling in mechanobiology*, vol. 10, no. 1, pp. 133–145, 2011.

[63] M Lefik, D. Boso, and B. Schrefler, “Artificial neural networks in numerical modelling of composites,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 21-26, pp. 1785–1804, 2009.

[64] N Huber, I Tsagrakis, and C. Tsakmakis, “Determination of constitutive properties of thin metallic films on substrates by spherical indentation using neural networks,” *International Journal of solids and structures*, vol. 37, no. 44, pp. 6499–6516, 2000.

[65] H Man and T Furukawa, “Neural network constitutive modelling for non-linear characterization of anisotropic materials,” *International journal for numerical methods in engineering*, vol. 85, no. 8, pp. 939–957, 2011.

[66] S. Jung and J. Ghaboussi, “Characterizing rate-dependent material behaviors in self-learning simulation,” *Computer methods in applied mechanics and engineering*, vol. 196, no. 1-3, pp. 608–619, 2006.

[67] R. E. Jones, J. A. Templeton, C. M. Sanders, and J. T. Ostien, “Machine learning models of plastic flow based on representation theory,” *arXiv preprint arXiv:1809.00267*, 2018.

[68] D. Versino, A. Tonda, and C. A. Bronkhorst, “Data driven modeling of plastic deformation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 318, pp. 981–1004, 2017.

[69] T. Kirchdoerfer and M. Ortiz, “Data-driven computational mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 304, pp. 81–101, 2016.

[70] M. Kapteyn, D. Knezevic, D. Huynh, M Tran, and K. Willcox, “Data-driven physics-based digital twins via a library of component-based reduced-order models,” *International Journal for Numerical Methods in Engineering*, 2020.

[71] M. Guo and J. S. Hesthaven, “Reduced order modeling for nonlinear structural analysis using gaussian process regression,” *Computer methods in applied mechanics and engineering*, vol. 341, pp. 807–826, 2018.

[72] T. Chatterjee, D. Karličić, S. Adhikari, and M. I. Friswell, “Wave propagation in randomly parameterized 2d lattices via machine learning,” *Composite Structures*, vol. 275, p. 114 386, 2021.

[73] G. Capuano and J. J. Rimoli, “Smart finite elements: A novel machine learning application,” *Computer Methods in Applied Mechanics and Engineering*, vol. 345, pp. 363–381, 2019.

[74] X. Xu, M. D’Elia, and J. T. Foster, “A machine-learning framework for peridynamic material models with physical constraints,” *arXiv preprint arXiv:2101.01095*, 2021.

[75] E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.

[76] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[77] T. M. Mitchell *et al.*, “Machine learning,”

[78] P. J. Braspenning, F. Thuijsman, and A. J. M. M. Weijters, *Artificial neural networks: an introduction to ANN theory and practice*. Springer Science & Business Media, 1995, vol. 931.

[79] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[80] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[81] M. F. Horstemeyer and D. J. Bammann, “Historical review of internal state variable theory for inelasticity,” *International Journal of Plasticity*, vol. 26, no. 9, pp. 1310–1334, 2010.

[82] A. Graves, “Long short-term memory,” in *Supervised sequence labelling with recurrent neural networks*, Springer, 2012, pp. 37–45.

[83] D. C. Stouffer and L. T. Dame, *Inelastic deformation of metals: models, mechanical properties, and metallurgy*. John Wiley & Sons, 1996.

[84] W. G. Bousman, “A qualitative examination of dynamic stall from flight test data,” *Journal of the American Helicopter Society*, vol. 43, no. 4, pp. 279–295, 1998.

[85] K. Fraser, “General requirements and techniques for component fatigue life substantiation in australian service helicopters,” AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA), Tech. Rep., 1991.

[86] S. J. Davis, “Predesign study for a modern 4-bladed rotor for rsra,” 1981.

[87] C. C. McColl, “A matched-harmonic confluence approach to rotor loads prediction with comprehensive application to flight test,” Ph.D. dissertation, Georgia Institute of Technology, 2012.

[88] C. Mballo and J. Prasad, “A real time scheme for rotating system component load estimation using fixed system measurements,” 2018.

[89] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[90] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[91] N. Ketkar, “Introduction to keras,” in *Deep learning with Python*, Springer, 2017, pp. 97–111.

[92] R. Blandford, D. Morton, S. Snow, and T. Rahl, “Tensile stress-strain results for 304l and 316l stainless steel plate at temperature,” in *ASME 2007 Pressure Vessels and Piping Conference*, American Society of Mechanical Engineers Digital Collection, 2007, pp. 617–628.

[93] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*, 3. MIT press Cambridge, MA, 2006, vol. 2.

[94] H. J. Logarzo, G. Capuano, and J. J. Rimoli, “Smart constitutive laws: Inelastic homogenization through machine learning,” *Computer Methods in Applied Mechanics and Engineering*, vol. 373, p. 113 482, 2021.

[95] J.-M. Battini, “A non-linear corotational 4-node plane element,” *Mechanics research communications*, vol. 35, no. 6, pp. 408–413, 2008.

[96] I. Navon and D. M. Legler, “Conjugate-gradient methods for large-scale minimization in meteorology,” *Monthly Weather Review*, vol. 115, no. 8, pp. 1479–1502, 1987.

[97] E Polak and G Ribiere, “Note on the convergence of methods of conjugate directions,” *Revue Francaise d’Informatique et de Recherche Operationnelle*, vol. 3, no. 16, pp. 35–43, 1969.

[98] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” 1999.

[99] U. Saetti and J. F. Horn, “Load alleviation flight control design using high-order dynamic models,” *Journal of the American Helicopter Society*, vol. 65, no. 3, pp. 1–15, 2020.

[100] U. Saetti, J. F. Horn, T. Berger, and M. B. Tischler, “Handling-qualities perspective on rotorcraft load alleviation control,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 10, pp. 1792–1804, 2020.

[101] C. Mballo and J. V. R. Prasad, “Trade-off between maneuver performance and component load limiting,” 2020.

[102] D. N. Alleyne and P. Cawley, “The interaction of lamb waves with defects,” *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 39, no. 3, pp. 381–397, 1992.

[103] J. Achenbach, *Wave propagation in elastic solids*. Elsevier, 2012.

[104] M. J. Lowe, “Matrix techniques for modeling ultrasonic waves in multilayered media,” *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 42, no. 4, pp. 525–542, 1995.

[105] W. Staszewski, C. Boller, and G. R. Tomlinson, *Health monitoring of aerospace structures: smart sensor technologies and signal processing*. John Wiley & Sons, 2004.

[106] W. D. Smith, “The application of finite element analysis to body wave propagation problems,” *Geophysical Journal International*, vol. 42, no. 2, pp. 747–768, 1975.

[107] I. Dobrovolsky, S. Zubkov, and V. Miachkin, “Estimation of the size of earthquake preparation zones,” *Pure and applied geophysics*, vol. 117, no. 5, pp. 1025–1044, 1979.

[108] T. C. Hanks, “B values and ω- γ seismic source models: Implications for tectonic stress variations along active crustal fault zones and the estimation of high-frequency strong ground motion,” *Journal of Geophysical Research: Solid Earth*, vol. 84, no. B5, pp. 2235–2242, 1979.

[109] B Taylor, H. Maris, and C Elbaum, “Phonon focusing in solids,” *Physical Review Letters*, vol. 23, no. 8, p. 416, 1969.

[110] A. Sukhovich, L. Jing, and J. H. Page, “Negative refraction and focusing of ultrasound in two-dimensional phononic crystals,” *Physical Review B*, vol. 77, no. 1, p. 014 301, 2008.

[111] I. E. Psarobas, “Phononic crystals–sonic band-gap materials,” *Zeitschrift für Kristallographie-Crystalline Materials*, vol. 220, no. 9-10, pp. IV–IV, 2005.

[112] R. S. Langley, “The response of two-dimensional periodic structures to point harmonic forcing,” *Journal of sound and vibration*, vol. 197, no. 4, pp. 447–469, 1996.

[113] F. Moser, L. J. Jacobs, and J. Qu, “Modeling elastic wave propagation in waveguides with the finite element method,” *Ndt & E International*, vol. 32, no. 4, pp. 225–234, 1999.

[114] A. Shah and J. J. Rimoli, “Smart parts,” *Finite Elements in Analysis and Design*, vol. under review, pp. 363–381, 2021.

[115] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[116] F. Chollet *et al.*, *Keras*, `https://keras.io`, 2015.

[117] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, 2. MIT press Cambridge, 2016, vol. 1.

[118] R Rafiee-Dehkharghani, A. Aref, and G. Dargush, “Planar stress wave attenuation in plates with circular voids and inclusions,” *Composites Part B: Engineering*, vol. 75, pp. 307–318, 2015.

[119] A. F. Bower, *Applied mechanics of solids*. CRC press, 2009.

[120] A. Movchan, N. Movchan, and S Haq, “Localised vibration modes and stop bands for continuous and discrete periodic structures,” *Materials Science and Engineering: A*, vol. 431, no. 1-2, pp. 175–183, 2006.

[121] M Ruzzene and F Scarpa, “Directional and band-gap behavior of periodic auxetic lattices,” *physica status solidi (b)*, vol. 242, no. 3, pp. 665–680, 2005.

[122] W. L. Bragg, “The diffraction of short electromagnetic waves by a crystal,” *Scientia*, vol. 23, no. 45, 1929.

[123] F Casadei, J. Rimoli, and M Ruzzene, “Multiscale finite element analysis of wave propagation in periodic solids,” *Finite Elements in Analysis and Design*, vol. 108, pp. 81–95, 2016.

[124] F. Bloch, “Über die quantenmechanik der elektronen in kristallgittern,” *Zeitschrift für physik*, vol. 52, no. 7, pp. 555–600, 1929.

[125] L. Brillouin, “Wave propagation in periodic structures: Electric filters and crystal lattices,” 1953.