Optimizing the Dynamic Behavior of Structures

Using Substructuring and Surrogate Modeling

D. Akçay Perdahcıoğlu

The research described in this thesis has been carried out in the framework of the *Artificial Intelligence for Industrial Applications* (AI4IA) project (Contract no. 514510) of the European Marie-Curie FP6 research training program coordinated by SKF-RDC.

Samenstelling van de promotiecommissie:

voorzitter en secretaris:	
Prof. dr. F. Eising	Universiteit Twente
promotor:	
Prof. dr. ir. A. de Boer	Universiteit Twente
assistent promotor:	
Dr. ir. M.H.M. Ellenbroek	Universiteit Twente
leden:	
Prof. dr. ir. R. Akkerman	Universiteit Twente
Prof. dr. ir. H.F.J.M. Koopman	Universiteit Twente
Prof. dr. ir. D.J. Rixen	Technische Universiteit Delft
Prof. dr. T.M. Heskes	Radboud Universiteit Nijmegen
Dr. ir. L.F.P. Etman	Technische Universiteit Eindhoven

ISBN 978-90-365-3057-6

DOI 10.3990/1.9789036530576

1st printing July 2010

Keywords: Dynamic analysis, optimization, substructuring, surrogate modeling

Cover design by Emin Semih Perdahcıoğlu and Didem Akçay Perdahcıoğlu, depicting a Boeing 737-800 airplane. The image is created using only open-source software: Blender, Inkscape and GIMP.

This thesis was prepared with LaTeX by the author and printed by Ipskamp Drukkers, Enschede, from an electronic document.

OPTIMIZING THE DYNAMIC BEHAVIOR OF STRUCTURES USING SUBSTRUCTURING AND SURROGATE MODELING

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 9 juli 2010 om 16.45 uur

door

Didem Akçay Perdahcıoğlu

geboren op 2 juli 1980
te Ankara, Turkije

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. A. de Boer

en de assistent promotor:

Dr. ir. M.H.M. Ellenbroek

Summary

During a design process, analysis and, when necessary, modification of the vibration characteristics of a structure are important. Thanks to the developments in computer technology and in numerical analysis methods, particularly the Finite Element Method, a structure can be analyzed extensively in the computer environment long before its first prototype is built. To improve its design and to find a globally optimal configuration, in theory it is possible to directly couple a Finite Element (FE) model with a global optimization method. However, in practice this may not be feasible for complex structures due to the required number of the FE calculations and the corresponding computational costs. Analysis time grows rapidly with the amount of details in the FE model. If the vibration characteristics of a structure need to be improved by modifying the design of the detailed sections, long running analyses are a bottleneck in optimization.

The objective of this thesis is to develop an efficient strategy for optimizing the dynamic behavior of complex structures. The strategy is required to be robust, accurate and able to provide a solution which is as close to the global optimum as possible.

In this research two optimization schemes were identified to find the global optimum: Multi-Level Hybrid Optimization (MLHO) and Multi-Start Local Optimization (MSLO). A stochastic derivative-free method, Genetic Algorithms (GAs), and a gradient-based method, Sequential Quadratic Programming (SQP), were utilized in these schemes. Based on the results of the numerical test problems, it was concluded that finding the global optimum necessitates a tremendous number of function evaluations, i.e. FE analyses. Therefore, FE models were decided to be replaced with their fast-to-evaluate surrogate models in the optimization algorithms. Three well-known methods, namely Polynomial approximation, Kriging and Neural Networks were studied for surrogate modeling. The advantages and the disadvantages of these methods were identified by numerical test problems. Defining accurate surrogate models relies on the data generated from the FE model. However, the number of the required data increases significantly with the number of the design variables. Hence, using accurate as well as computationally efficient structural analysis methods is still essential to develop effective optimization strategies.

A considerable part of the research was focused on the Component Mode Synthesis (CMS) for dynamic analysis. Among the available CMS techniques special attention was paid to the Craig-Bampton (CB) method which is highly regarded for its simplicity and computational stability. The CB method was found to be very

advantageous to be utilized in an optimization strategy. Some of its perceivable benefits can be summarized as:

- A complex structure is divided into several substructures (components). The analysis of these can be assigned to different computers or different design groups. Hence, parallel processing is highly encouraged.
- Each substructure model can be condensed significantly while preserving the necessary information. Once these reduced models are coupled, the size of the model of the entire structure is considerably smaller than that of the original one. Accordingly, performing the dynamic analysis is significantly less time consuming.
- Analysis of each component is independent. Therefore, repeated analyses are required only for the modified substructures.
- In a complex structure with similar components in its geometry, one parametric model can be employed for the similar parts. Hence, the structure does not need to be modeled completely.

During optimization, a large number of repeated analyses of a structure might still be required even when surrogate models are used. This in turn implies a large number of recondensation operations of each modified component model. In order to reduce the analysis time even further, reanalysis methods were considered under the framework of CB. The Enriched CB method and a new Combined Approximations (CA) based method were proposed to update the transformation matrix which is responsible for the reduction of the substructures. With these methods, the computational efforts for condensing the modified substructures were aimed to be decreased.
Finally, all these concepts were gathered and a Surrogate-Based Optimization (SBO) strategy was defined where the modeling and the analysis opportunities of CB were exploited. Academic problems were used to demonstrate the influence of the introduced methods on the efficiency of the SBO method. It was concluded that taking advantage of effective structural analysis and reanalysis techniques is as important as utilizing efficient numerical techniques during optimization of complex structures.

Samenvatting

Tijdens een ontwerpproces zijn analyse en, indien nodig, wijziging van de trillingskenmerken van een constructie belangrijk. Dankzij de ontwikkelingen in computertechnologie en in numerieke analysemethoden, met name de Eindige Elementen Methode, kan een constructie uitgebreid worden geanalyseerd in de computeromgeving, lang voordat het eerste prototype wordt gebouwd. Voor het verbeteren van het ontwerp van een constructies, is het in theorie mogelijk om een Eindig Elementen (EE) model direct aan een numerieke optimalisatiemethode te koppelen voor het vinden van een globaal optimale ontwerpconfiguratie. Echter, voor complexe constructies is dit in de praktijk niet mogelijk vanwege de vele EE-berekeningen en de bijbehorende rekenkosten. De analysetijd groeit snel met het aantal details in het EE-model. Als de dynamische eigenschappen van een constructie moeten worden verbeterd door het ontwerp van de gedetailleerde onderdelen aan te passen, vormen tijdrovende analyses een knelpunt in de optimalisatie.

Het doel van dit proefschrift is het ontwikkelen van een efficiënte strategie voor het optimaliseren van het dynamische gedrag van complexe constructies. De strategie moet robuust en nauwkeurig zijn en in staat zijn om een oplossing te leveren die zo dicht mogelijk bij het globale optimum ligt.

In dit onderzoek zijn twee optimalisatiestrategieën gebruikt om het globale optimum te vinden: Multi-Level Hybrid Optimization (MLHO) en Multi-Start Local Optimization (MSLO). Een stochastisch afgeleide-vrije methode, Genetic Algorithms (GAs), en een gradiëntgebaseerde methode, Sequential Quadratic Programming (SQP), zijn in deze strategieën gebruikt. Gebaseerd op de resultaten van numerieke testproblemen, is geconcludeerd dat voor het vinden van het globale optimum een enorm aantal functie-evaluaties, dat wil zeggen EE-analyses noodzakelijk is. Derhalve is er besloten de EE-modellen in de optimalisatiealgoritmen te vervangen door hun snel te evalueren surrogaatmodellen.

Drie bekende methoden, namelijk Polynomial approximation, Kriging en Neural Networks zijn bestudeerd voor surrogaatmodellering. De voor- en de nadelen van deze methoden zijn geïdentificeerd aan de hand van numerieke testproblemen. De definitie van nauwkeurige surrogaatmodellen is gebaseerd op de gegevens die zijn gegenereerd met het EE-model. Het aantal vereiste gegevens neemt echter aanzienlijk toe met het aantal ontwerpvariabelen. Het gebruik van nauwkeurige en rekenefficiënte structurele analysemethoden is nog steeds van essentieel belang voor het ontwikkelen van effectieve optimalisatiestrategieën.

Een aanzienlijk deel van het onderzoek is gericht op de Component Mode Synthesis

(CMS) voor de dynamische analyse. Van de beschikbare CMS-technieken is speciaal aandacht besteed aan de Craig-Bampton (CB) methode die wordt gewaardeerd vanwege haar eenvoud en rekenstabiliteit. De CB-methode blijkt zeer effectief te kunnen worden gebruikt in een optimalisatiestrategie. Enkele van de belangrijkste voordelen kunnen als volgt worden samengevat:

- Een complexe constructie is onderverdeeld in verschillende substructuren (componenten). De analyse van deze substructuren kan worden toegewezen aan verschillende computers of verschillende ontwerpgroepen. Parallelle verwerking wordt daardoor sterk bevorderd.
- Elk subconstructiemodel kan door condensatie aanzienlijk worden gereduceerd met behoud van de nodige informatie. Wanneer deze gereduceerde modellen worden gekoppeld, is de grootte van het model van de hele constructie aanzienlijk kleiner dan die van het originele EE-model. Het uitvoeren van de dynamische analyse wordt dan significant minder tijdrovend.
- De analyse van elke component is onafhankelijk. Analyses hoeven daarom alleen herhaald te worden voor de gewijzigde substructuren.
- Voor een complexe constructie met vergelijkbare componenten in de geometrie, kan één parametrisch model worden gebruikt voor de vergelijkbare onderdelen. De constructie hoeft daarom niet volledig te worden gemodelleerd.

Tijdens de optimalisatie kan nog steeds een groot aantal herhaalde analyses van een constructie noodzakelijk zijn, zelfs als surrogaatmodellen worden gebruikt. Dit impliceert een groot aantal her-condensaties van elk gewijzigd componentmodel. Om de analysetijd nog verder te reduceren, zijn her-analysemethoden onderzocht in het kader van de CB-methode. De Enriched CB-methode en een nieuwe op de Combined Approximations (CA) gebaseerde methode zijn voorgesteld om de transformatiematrix te updaten die verantwoordelijk is voor de reductie van de subconstructies. Met deze methoden is beoogd de rekentijd voor het condenseren van de gewijzigde substructuren te reduceren.
Tenslotte zijn al deze concepten samengevoegd en is er een Surrogate-Based Optimization (SBO) strategie gedefinieerd waarbinnen de modellerings- en de analysemogelijkheden van de CB-methode zijn benut. Academische problemen zijn gebruikt voor het demonstreren van de invloed van de geïntroduceerde methoden op de efficiëntie van de SBO-methode. Er is geconcludeerd dat het benutten van effectieve structurele analyse- en heranalysetechnieken even belangrijk is als het gebruik van efficiënte numerieke technieken tijdens de optimalisatie van complexe structuren.

Contents

1

Introduction

Vibration is the oscillatory motion of a physical system. The motion may be harmonic, periodic or a general motion in which the amplitude varies in time [120].

This universal phenomenon of nature and many of its aspects are a part of everyday life from fundamental human activities such as speaking and hearing to skyscrapers and bridges oscillating under the wind. The fact that "*sounding bodies are in a state of vibration*" [112] is a simple clue indicating the importance of this subject.

Vibrating systems are generally utilized in close association with sound and communications *e.g.* speaker systems and musical instruments. However, vibrations can also be generated unintentionally due to external effects such as wind or earthquakes, or internal effects such as unbalance in rotating machinery.

Excessive vibrations result in the generation of varying stress amplitudes which are likely to induce fatigue in structures. In machines, they may lead to rapid wear and tear of parts like bearings and gears. In many engineering systems human interaction is unavoidable. In that case, transmission of vibrations on a human body may have physical as well as psychological consequences. Blurred vision, loss of balance, hearing loss, back pain etc. are some of the symptoms seen in people exposed to vibrations [57, 88]. Noise, unwanted sound, interferes with the comfort and peace of neighboring inhabitants.

Suppression or elimination of unwanted vibrations as well as amplification of desired ones are crucial to improve product quality, to increase process efficiency, to prolong machinery life, to increase reliability and safety of systems, to improve the quality of work and living environments, etc. For this purpose a thorough understanding of vibration characteristics of physical systems is essential.

Until about a few decades ago vibration studies were performed using much simplified models. These models are not sufficient to describe vibration characteristics of continuous systems with complex geometries. Parallel developments in numerical methods and computer technology enabled engineers to analyze complex mechanical and structural systems with high accuracy. This progress led to performing *what if* design scenarios on the shape, size and material properties of structures for modifying

their vibration characteristics in the computer environment. Consequently, it became possible to perform extensive analyses before building the first prototypes. The next natural step was to perform these modifications in a systematic way instead of using *trial-and-error* methodologies.

Optimization is concerned with the analysis and the solution of problems so as to achieve the best way of satisfying the original need within the available means [107].

Optimization is an ever-present concept of daily life. In essence, any decision-making process such as traveling from one place to another, time management, investments, etc. is based on solving optimization problems where the solver is the human brain. More consciously, in fields including business, economics, natural sciences, engineering, etc. optimization problems are solved with the aid of computers where the problem formulations are based on mathematical abstractions of reality.

A mathematical optimization problem consists of [107]:

- A set of *variables* describing the design alternatives,
- An *objective function* expressed in terms of the design variables which defines the original need to be maximized or minimized,
- *Constraint functions* expressed in terms of the design variables which are the conditions that must be satisfied by any acceptable design.

Once the problem is defined, a solution is obtained using suitable numerical optimization methods. These methods require the evaluation of the objective and the constraint functions several times in order to find an optimal solution.

In engineering applications, designing and producing both economical and efficient products are necessary in order to be able to withstand global competition in the market. This is one of the main motivation of using optimization methods for many manufacturing companies. *Structural design optimization* problems require reliable analysis which is generally carried out by the Finite Element (FE) method. One of the main difficulties is that optimization of complex structures may necessitate numerous computationally demanding FE analyses.

1.1 About this Research

The research described in this thesis has been carried out in the framework of the *Artificial Intelligence for Industrial Applications* (AI4IA) project (Contract no. 514510) of the European Marie-Curie FP6 research training program coordinated by SKF-RDC. The goal of the project is to investigate, identify, demonstrate and promote Artificial Intelligence (AI) techniques for industrial applications. The objective of this thesis under the AI4IA project is to develop a strategy for the optimization of the dynamic behavior of structures. The strategy is aimed to be suitable for complex structures for which the global dynamic behavior is required to be modified via local design parameters. Throughout the research, various numerical methods are

utilized. Among these Genetic Algorithms and Artificial Neural Networks are the two well-known AI techniques.

The research topics concerning this study can be divided into two main groups: *Structural Dynamics* and *Structural Optimization*. The scope of this thesis is limited to certain fields in these broad categories.

In the structural dynamics field, only the modal and the harmonic response analyses are in focus. The analyses are based on the linear theory of vibration in which the geometry and the material properties of the analyzed structures are assumed to be not affected from the amplitude of vibration.

In structural optimization, only the sizing optimization problems are concerned in which the design variables can be, for instance, the cross-section properties of the beam elements and the thickness of the shell elements as well as the material properties of an FE model. Hence, the shape and the topology of the structures are preserved during the optimization procedure. Moreover, selected design variables are assumed to be defined over a continuous domain where they can take any real value.

As emphasized earlier, analysis time is a major drawback for optimization of complex structures. In order to be able to define an efficient optimization strategy, employing accurate and computationally less demanding analysis methods plays a crucial role. Moreover, the optimum solution, preferably the global optimum, should be obtained with as small a number of calls to the FE model as possible.

Understanding the algorithms behind the numerical optimization methods is important to define an effective strategy as well as to assess its limits. Thus, one of the research topics of the thesis is to analyze the differences between the *global optimization* and the *local optimization* methods. Algorithms that are developed for finding *the global* optimum usually require much more function evaluations compared to the local optimization algorithms whose concern is to calculate *an* optimum. In structural optimization this means numerous executions of an FE model.

In order to develop a computationally efficient optimization strategy, another important research subject of the thesis is *surrogate modeling*. The logic behind using surrogate models is to generate a simple mathematical model (surrogate) based on the data collected from an FE model for various values of the predefined design variables. This model then replaces the FE model in the numerical optimization procedure. Once generated, evaluation of surrogate models is many orders of magnitude faster than the FE analyses. This makes them very effective to be used in a global optimization algorithm.

Although using surrogate models minimizes the interaction between the global optimization method and the FE model, structural analyses are still required to generate data for surrogates. Thus, computational efficiency of the FE analyses is as important as their accuracy during optimization. A considerable part of the research is focused on studying the *reduction* and the *sub-structuring* methods for dynamic analysis. With these methods detailed FE models can be reduced (condensed) significantly while preserving a certain level of accuracy. Additionally, they provide an opportunity to divide a complete structure into parts and perform independent condensation.

During optimization, the repeated analyses (reanalysis) of even the reduced FE models can be computationally infeasible. Therefore in the thesis, advantages of using *reanalysis methods* in combination with reduction are studied as well. Thereby the computational burden is tried to be decreased further.

1.2 Outline of the Thesis

This thesis consists of seven chapters and four appendices. The current chapter, Chapter 1, describes the motivation and the objective of the research. The theoretical basis about the structural analysis methods and the numerical methods utilized in this study are given in Chapters 2, 3 and 4. The proposed optimization strategy is presented in Chapter 5 which is the central part of the thesis. Chapter 6 involves two academic test problems which are used to test the strategy in terms of efficiency, accuracy and robustness. Finally, in Chapter 7, the main conclusions of the research are summarized.

The chapters do not have to be read successively. One can start with Chapter 5 and whenever required return to the previous chapters for detailed information about the employed methods. Chapters 2, 3 and 4 include self-contained subjects which can be read separately.

A brief summary of the contents of each chapter is as follows:

Chapter 2 discusses the reduction and the sub-structuring methods. Special attention is paid to the Craig-Bampton (CB) method. Additionally, a new reanalysis approach developed to be used within the CB method is presented.

Chapter 3 covers a brief introduction about the concepts of numerical optimization. A stochastic derivative-free method, Genetic Algorithms, and a gradient-based method, Sequential Quadratic Programming, are presented. Furthermore, Multi-Start Local Optimization and Multi-Level Hybrid Optimization schemes are introduced which are employed for the search of the global optimum. The effectiveness of these schemes is discussed by numerical test problems.

Chapter 4 is about surrogate modeling. After defining the fundamental steps of the approach, three well-known techniques, namely Polynomial models, Kriging and Artificial Neural Networks are introduced. The advantages and the disadvantages of these techniques are discussed using numerical test problems.

Chapter 5 is the heart of the thesis where the methods discussed in the previous chapters are gathered to define the proposed optimization strategy. A brief introduction about the concepts of structural optimization field and the corresponding literature is also presented in this chapter.

Chapter 6 is the part where the proposed optimization strategy is demonstrated by two academic test problems. In the problems, the harmonic response and the modal properties of the structures are intended to be modified. The selected structures have repeating patterns in their geometries in order to exhibit the analysis and the modeling features of the CB method during optimization. The influence of the Kriging and the

Neural Network surrogates on the final results of the proposed strategy is investigated. Moreover, the contribution of the reanalysis methods on the computational efficiency and the accuracy of the strategy is tested.

Chapter 7, the final chapter of this thesis, includes the summary, conclusions and the future research areas concerning this study.

2

Reduction & Reanalysis Methods

As introduced in Chapter 1, the objective of this research is to develop an optimization strategy which can be used in the field of structural dynamics. To be able to do this, first, a model of a structure is required which allows dynamic response analysis to be performed. The Finite Element Method (FEM) is used to build this model in the current study. In FEM, a geometrically complex structure is divided into a finite number of simple domains (elements) and the solution is approximated using interpolation functions defined over these elements. Generally, the accuracy of the solution improves with the number of the elements in the model, but this also increases the computation time. For the dynamic analysis problems, detailed models can be condensed using appropriate reduction methods for decreasing the computational costs while it is still possible to preserve the accuracy for certain frequency ranges.

In structural optimization, repeated analysis (reanalysis) of a structure is needed for each modification in the design. This is a computationally demanding task if the structure under study is complex, even when the reduced models are employed. In order to decrease the computational effort more, reanalysis methods can be used. They utilize the knowledge of the initial design to calculate the response of the structure for the modified design values.

Consequently, reduction and reanalysis methods play a crucial role for efficient optimization of structures. These methods are discussed briefly in this chapter. Special attention is paid to the reduction method, Craig-Bampton (CB). Two reanalysis methods that can be utilized in CB are presented. The introduced concepts are demonstrated by an academic test problem.

Part of this chapter is based on the article "D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, H.J.M. Geijselaers and A. de Boer. (2010) *Updating the Craig-Bampton Reduction Basis for Efficient Structural Reanalysis.* International Journal for Numerical Methods in Engineering. Accepted." [4].

2.1 General Concepts in Dynamic Analysis

This section briefly highlights the derivation of the discrete form of the *equations of motion* used in the FE context. Moreover, the general terminology of *dynamic analysis* concerning this thesis is presented. The interested reader is referred to [33, 99, 113] for further details.

2.1.1 Governing Equations, Weak Form and FE Formulation

The governing equations to describe the undamped motion of a flexible body given in indicial notation are

$$\begin{aligned} \frac{\partial \sigma_{ij}}{\partial x_i} + \bar{b}_j - \rho\, \ddot{u}_j = 0 \qquad & \text{in}\,\Omega \\ \sigma_{ij} = \sigma_{ji} \qquad & \text{in}\,\Omega \\ n_i \sigma_{ij} = \bar{t}_j \qquad & \text{on}\,\Gamma_t \qquad (2.1) \\ u_j = \bar{u}_j \qquad & \text{on}\,\Gamma_u \qquad (2.2) \end{aligned}$$

where σ_{ij} is the stress tensor, $x_i = [x_1, x_2, x_3]$ are the reference Cartesian coordinates, $u_i(x_j, t) = [u_1, u_2, u_3]$ are the displacement field observed at any point resulting from the dynamic deformation of the body. In the formulation, $\bar{b}_j$ are the applied body forces, $\bar{t}_j$ are the surface tractions imposed on the surface Γ_t, $\bar{u}_j$ are the prescribed displacements on the surface Γ_u, Ω defines the domain occupied by the body and its associated mass density ρ and, finally, n_i are the outward normals to the surface Γ_t of the body.

Using the weighted residual method, integration by parts and the divergence theorem, the weak form of Equation (2.1) becomes

$$\int_\Omega \frac{\partial w_j}{\partial x_i}\, \sigma_{ij}\, \mathrm{d}\Omega - \int_\Omega w_j\, \bar{b}_j\, \mathrm{d}\Omega - \int_{\Gamma_t} w_j\, \bar{t}_j\, \mathrm{d}\Gamma + \int_\Omega w_j\, \rho\, \ddot{u}_j\, \mathrm{d}\Omega = 0 \qquad (2.3)$$

where w_j are the weighting functions.

When the linear elastic stress-strain relationship

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl}$$

and the linear strain-displacement relationship

$$\epsilon_{kl} = \frac{1}{2}\left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\right)$$

are inserted into Equation (2.3), the weak formulation becomes:

$$\begin{aligned} \frac{1}{2} \int_\Omega \frac{\partial w_j}{\partial x_i}\, C_{ijkl} \left(\frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k}\right) \mathrm{d}\Omega - \int_\Omega w_j\, \bar{b}_j\, \mathrm{d}\Omega \\ - \int_{\Gamma_t} w_j\, \bar{t}_j\, \mathrm{d}\Gamma + \int_\Omega w_j\, \rho\, \ddot{u}_j\, \mathrm{d}\Omega = 0 \end{aligned} \qquad (2.4)$$

where C_{ijkl} is the tensor with the elasticity constants and ϵ_{kl} is the strain tensor.

In FE formulations, generally instead of the indicial one, the matrix-vector notation is utilized. Therefore, Equation (2.4) can be converted to the matrix-vector notation as:

$$\int_{\Omega} \nabla \mathbf{w}^{\mathrm{T}} \mathbf{C}\, \boldsymbol{\epsilon}\, \mathrm{d}\Omega - \int_{\Omega} \mathbf{w}^{\mathrm{T}}\, \bar{\mathbf{b}}\, \mathrm{d}\Omega - \int_{\Gamma_t} \mathbf{w}^{\mathrm{T}}\, \bar{\mathbf{t}}\, \mathrm{d}\Gamma + \int_{\Omega} \mathbf{w}^{\mathrm{T}}\, \rho\, \ddot{\mathbf{u}}\, \mathrm{d}\Omega = 0. \tag{2.5}$$

For defining the FE formulation of Equation (2.5), the current domain Ω is divided into sub-domains (elements) Ω_e and the solution is approximated in each element using interpolation functions. Hence, the weak form, Equation (2.5), forms the basis of the FE discretization. "*With an assumed displacement field within an element that only depends on the nodal displacements (degrees of freedom (d.o.f.)) from that element, the integrals that appear in the weak form can be evaluated element by element*" [133]. The d.o.f. of an element are represented by the vector $\mathbf{d}_e$ which is a sub-vector of $\mathbf{d}$. The vector $\mathbf{d}$ involves the nodal d.o.f. of the complete structure.

The interpolation over an element is defined as

$$\mathbf{u} = \mathbf{N}\mathbf{d}_e \tag{2.6}$$

where $\mathbf{N}$ is the matrix of interpolation (shape) functions. By using Equation (2.6), the linear strain-displacement relationship becomes

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{d}_e \tag{2.7}$$

where $\mathbf{B}$ is a matrix containing the gradients of $\mathbf{N}$.

With these definitions, the weak formulation at the element level is

$$\mathbf{K}_e\, \mathbf{d}_e + \mathbf{M}_e\, \ddot{\mathbf{d}}_e = \bar{\mathbf{f}}_e \tag{2.8}$$

where

$$\mathbf{K}_e = \int_{\Omega_e} \mathbf{B}^{\mathrm{T}}\mathbf{C}\mathbf{B}\, \mathrm{d}\Omega\,, \quad \mathbf{M}_e = \int_{\Omega_e} \rho \mathbf{N}^{\mathrm{T}}\mathbf{N}\, \mathrm{d}\Omega\,, \quad \bar{\mathbf{f}}_e = \int_{\Omega_e} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{b}}\, \mathrm{d}\Omega + \int_{\Gamma_e} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\, \mathrm{d}\Gamma$$

are the element stiffness and mass matrices and the external force vector, respectively. Within each element, the set of weighting functions is chosen equal to the set of interpolation functions.

When all the element equations are assembled, the equations of motion in discrete form are obtained as:

$$\mathbf{K}\, \mathbf{d} + \mathbf{M}\, \ddot{\mathbf{d}} = \bar{\mathbf{f}}. \tag{2.9}$$

2.1.2 The Generalized Eigenvalue Problem

The generalized eigenvalue problem of an undamped discrete system is given by

$$(\mathbf{K} - \omega^2 \mathbf{M})\mathbf{d} = (\mathbf{K} - \lambda \mathbf{M})\mathbf{d} = \mathbf{0}. \tag{2.10}$$

In Equation (2.10), $\lambda = \omega^2$ are the eigenvalues, indicating the square of the free vibration frequencies while the corresponding displacement vectors $\mathbf{d}$ express the eigenvectors or the natural mode shapes of the vibrating system.

The eigenvalues and the eigenvectors are often written in the matrix form of

$$\mathbf{\Lambda} = \begin{bmatrix} \omega_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \omega_n^2 \end{bmatrix}, \quad \mathbf{\Phi} = [\Phi_1, \Phi_2, \dots, \Phi_n]$$

where $\mathbf{\Lambda}$, $\mathbf{\Phi}$ are called the *spectral matrix* and the *modal matrix*, respectively and n is the total number of d.o.f. of the system.

The solution of the eigenvalue problem, for large structures having many d.o.f., is often the most costly phase of the dynamic response analysis. The calculation of the eigenvalues and the eigenvectors requires high computational effort.

2.2 Reduction Methods

For the analysis of structures having a large number of d.o.f., a large system of equations must be solved. In most dynamic analysis problems, the lower natural frequencies and the corresponding modes are more interesting than the higher ones. This is because these modes tend to dominate the dynamic behavior of structures and resonance effects are more severe at the lower natural frequencies [75]. In these problems, the required number of d.o.f. to solve the system accurately is much less than the number of the actual d.o.f. Therefore, condensing the FE models of these structures before the dynamic response analysis is a very frequently used strategy. Reduction is achieved by employing a few preselected basis vectors which span the solution space of the approximation. This is known as the *reduced basis approach* [81, 83].

In the context of large projects (e.g. an aircraft design), the tendency is to divide a complex structure into several parts, generate the corresponding FE models, reduce each model and then couple them to obtain the reduced model of the entire structure. Dividing a large problem into subparts in order to simplify its analysis is called *substructuring*. This technique is commonly preferred in industry because it allows modeling of each substructure (component) by different design groups. On top of that, if a modification is required in any specific component (e.g. the solid rocket boosters of a space shuttle or the interstage of a launcher), only the system matrices of that particular substructure are changed and coupled with the rest of the already analyzed substructures. This saves valuable computation time. Some structures involve repeating patterns in their geometry e.g. the fuselage panels of a plane, one cyclic sector of an industrial blisk, a bladed disk, etc. Modeling one repeating component and utilizing its system matrices for the remaining identical parts is another advantage of substructuring.

The so-called Component Mode Synthesis (CMS) technique is both a reduction and a sub-structuring method which has been utilized since the 1960s for the dynamic

analysis of complex structures. CMS consists of breaking up a large structure into several substructures, obtaining reduced order system matrices of each component and then assembling these matrices to obtain the reduced order system matrices of the entire structure. Technical details of CMS are summarized in Section 2.2.1.

Depending on the type of the boundary conditions applied on the component interface nodes, CMS can be grouped into *fixed interface* methods [36, 73], *free interface* methods [54, 92, 94, 116, 117] and *loaded interface* methods [13]. In this study, *Craig-Bampton* (CB) [36], a fixed interface CMS method is considered which is highly regarded for its simplicity and computational stability. A brief introduction about CB is given in Section 2.2.2.

2.2.1 Component Mode Synthesis

Assume that an FE model of a structure is constructed on a domain Ω and is divided into S non-overlapping substructures such that each component is defined on the sub-domain Ω^c. Thus, excepting the nodes on the interface boundaries, each node belongs to one and only one component. The linear dynamic behavior of an undamped component, labeled c, is governed by the equations,

$$\mathbf{M}^c\ddot{\mathbf{d}}^c + \mathbf{K}^c\mathbf{d}^c = \mathbf{f}^c + \mathbf{g}^c \quad c = 1, 2, \ldots, S \tag{2.11}$$

where $\mathbf{M}^c$, $\mathbf{K}^c$ and $\mathbf{d}^c$ are the mass matrix, the stiffness matrix and the vector of the local d.o.f. of the component, respectively. The vector $\mathbf{f}^c$ represents the external loads, and the vector $\mathbf{g}^c$ represents the interface loads between the component c and the neighboring components that ensure compatibility at the interfaces.

For reducing the size of the component matrices, $\mathbf{K}^c$ and $\mathbf{M}^c$, a subspace spanned by the columns of $\mathbf{T}^c$ is built in such a way that the solution of Equation (2.11) can be written in the form:

$$\mathbf{d}^c \approx \mathbf{T}^c\mathbf{q}^c \tag{2.12}$$

where $\mathbf{q}^c$ is a vector of generalized coordinates and $dim(\mathbf{q}^c) \ll dim(\mathbf{d}^c)$. $\mathbf{T}^c$ is referred to as a *reduction basis*, a *transformation matrix* or a *Ritz basis*. The reduction basis should be able to describe the deformations on a component as the component undergoes a free vibration motion as a part of an entire structure [126].

After defining the reduction basis $\mathbf{T}^c$, first, the right-hand side of Equation (2.12) is substituted into Equation (2.11) and then, Equation (2.11) is pre-multiplied by ${\mathbf{T}^c}^{\mathrm{T}}$. Hence, the reduced matrices of each component are defined by: $\bar{\mathbf{K}}^c = {\mathbf{T}^c}^{\mathrm{T}}\mathbf{K}^c\mathbf{T}^c$, $\bar{\mathbf{M}}^c = {\mathbf{T}^c}^{\mathrm{T}}\mathbf{M}^c\mathbf{T}^c$. The external loads and the interface loads are $\bar{\mathbf{f}}^c = {\mathbf{T}^c}^{\mathrm{T}}\mathbf{f}^c$ and $\bar{\mathbf{g}}^c = {\mathbf{T}^c}^{\mathrm{T}}\mathbf{g}^c$, respectively.

The next step is the coupling (assembly) of all these matrices to obtain the reduced model of the entire structure. Coupling is achieved by satisfying the compatibility and the equilibrium conditions at the interfaces of the components.

2.2.2 Craig-Bampton (CB) Method

In the Craig-Bampton method [36], the reduction basis is obtained by utilizing the *fixed interface normal modes* and the *constraint modes*. The first set of modes describes the internal dynamic behavior of a substructure. The motion on the substructure interfaces, the propagation of the forces between substructures and the necessary information about the rigid body motions are defined by the *constraint modes*.

For defining the reduction basis of the CB method, the partitioned form of Equation(2.11)

$$\begin{bmatrix} \mathbf{M}^c_{\mathrm{ii}} & \mathbf{M}^c_{\mathrm{ib}} \\ \mathbf{M}^c_{\mathrm{bi}} & \mathbf{M}^c_{\mathrm{bb}} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{d}}^c_{\mathrm{i}} \\ \ddot{\mathbf{d}}^c_{\mathrm{b}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}^c_{\mathrm{ii}} & \mathbf{K}^c_{\mathrm{ib}} \\ \mathbf{K}^c_{\mathrm{bi}} & \mathbf{K}^c_{\mathrm{bb}} \end{bmatrix} \begin{Bmatrix} \mathbf{d}^c_{\mathrm{i}} \\ \mathbf{d}^c_{\mathrm{b}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^c_{\mathrm{i}} \\ \mathbf{f}^c_{\mathrm{b}} \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{g}^c_{\mathrm{b}} \end{Bmatrix} \tag{2.13}$$

is used where "i" and "b" refer to *interior* and *boundary*, respectively.

The fixed interface normal modes are calculated by restraining all d.o.f. at the interface and solving an undamped free vibration problem:

$$(\mathbf{K}^c_{\mathrm{ii}} - \omega_j^2 \mathbf{M}^c_{\mathrm{ii}}) \{\Phi^c_{\mathrm{i}}\}_j = 0 \qquad j = 1, 2, \ldots, N_{\mathrm{T}} \tag{2.14}$$

where ω_j, $\{\Phi^c_{\mathrm{i}}\}_j$ are the jth natural frequency and the corresponding mode shape respectively, and, N_{T} is the truncated number of the normal modes which is usually a lot less than the actual number. Hence, the fixed interface normal modes of a component c are given by

$$\begin{bmatrix} \{\Phi^c_{\mathrm{i}}\}_1 & \{\Phi^c_{\mathrm{i}}\}_2 & \ldots & \{\Phi^c_{\mathrm{i}}\}_{N_{\mathrm{T}}} \\ \mathbf{0}_{\mathrm{b}} & \mathbf{0}_{\mathrm{b}} & \ldots & \mathbf{0}_{\mathrm{b}} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi}^c_{\mathrm{i}} \\ \mathbf{0} \end{bmatrix}. \tag{2.15}$$

The constraint modes are calculated by statically imposing a unit displacement to the interface d.o.f. one by one while keeping the displacement of the other interface d.o.f. zero and assuming that there are no internal reaction forces, i.e.,

$$\begin{bmatrix} \mathbf{K}^c_{\mathrm{ii}} & \mathbf{K}^c_{\mathrm{ib}} \\ \mathbf{K}^c_{\mathrm{bi}} & \mathbf{K}^c_{\mathrm{bb}} \end{bmatrix} \begin{bmatrix} \mathbf{\Psi}^c_{\mathrm{ib}} \\ \mathbf{I}^c_{\mathrm{bb}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^c_{\mathrm{ib}} \\ \mathbf{R}^c_{\mathrm{bb}} \end{bmatrix}. \tag{2.16}$$

In Equation (2.16), the columns of the matrix $\mathbf{R}^c_{\mathrm{bb}}$ are the unknown reaction forces acting on the interface. The constraint mode matrix of a component c is

$$\begin{bmatrix} \mathbf{\Psi}_{\mathrm{ib}}{}^c \\ \mathbf{I}_{\mathrm{bb}} \end{bmatrix} = \begin{bmatrix} -\mathbf{K}^{c\,-1}_{\mathrm{ii}} \mathbf{K}^c_{\mathrm{ib}} \\ \mathbf{I}_{\mathrm{bb}} \end{bmatrix}. \tag{2.17}$$

Therefore, the *Craig-Bampton transformation matrix* $\mathbf{T}^c_{\mathrm{CB}}$ for component c is defined as,

$$\mathbf{T}^c_{\mathrm{CB}} = \begin{bmatrix} \mathbf{\Phi}^c_{\mathrm{i}} & \mathbf{\Psi}^c_{\mathrm{ib}} \\ \mathbf{0} & \mathbf{I}_{\mathrm{bb}} \end{bmatrix}. \tag{2.18}$$

In the CB method, the assembly is done using the compatibility of the interface d.o.f [116]. This implies matching FE meshes at the interfaces.

2.3 Reanalysis Methods for the CB Method

When a substructure is reanalyzed with modified design parameters (e.g in an optimization algorithm), the static and the dynamic properties of the component are not the same as the initial ones anymore. Consequently, the reduction basis is also different. One can either reuse the reduction basis of the initial component or compute a new basis for the condensation of the matrices of the modified component. The first option usually leads to inaccurate results. The second one requires solving free vibration problems and performing static analysis which are computationally demanding for the complex structures. Investigating alternative and fast methods that can be used for the calculation of the CB reduction basis during the reanalysis of the substructures is the main motivation of this section. In particular, the focus is on identifying methods that can be employed within sizing optimization problems. In these types, modifications are introduced via changing the thicknesses of the shell elements, the cross-section properties of the beam elements and the corresponding material properties. Hence, the FE mesh of the initial structure is preserved.

The fixed interface normal modes are utilized to define the internal dynamic behavior of a component. They are calculated by solving free vibration problems which can be demanding with an increasing number of d.o.f. in an FE model. In the literature, there are some methods proposed for updating the normal (vibration) modes of a CMS basis. In [23], these modes are updated using the Combined Approximations (CA) approach [82] whereas the constraint modes are calculated all over again. Masson et al. [96] propose to extend the fixed interface normal mode set of an initial substructure with some additional basis vectors (Ritz vectors) and to use it as a new normal mode set in the reanalysis of this substructure. These additional basis vectors are gathered calculating the residual forces acting on the modified substructure. The proposed method in [96] assumes that the constraint modes of a modified component are correctly represented by that of the initial one. This method will be named as the Enriched Craig-Bampton (ECB) method in the rest of the chapter.

The constraint modes are used for describing the motion of the interfaces of the components. The rigid body motions of a structure are also defined using these modes. Moreover, they contribute to define the interior flexibility of the reduced models. In lower frequency ranges, these modes are very important to prescribe the global dynamic behavior accurately. Therefore, they should be taken into account in the reanalysis procedures. In this research, in order to reduce the calculation time of the exact constraint modes, a reanalysis method based on the Combined Approximations (CA) approach is proposed whose details are presented in Section 2.3.2. Furthermore, it is combined with the ECB method for efficient, fast and accurate estimation of the CB reduction bases of the modified component models. The ECB method is introduced in Section 2.3.1. The proposed updating scheme is validated and compared with the results of several other methods using an academic test problem in Section 2.4.

2.3.1 Updating the Fixed Interface Normal Modes

This section summarizes the ECB method, proposed by Masson et al. [96], used for updating the initial fixed interface normal modes when a substructure is modified.

In the rest of the text only one component will be considered, although the introduced method can be extended for multiple components as well. Hence, the superscript "c" will be omitted in the formulations.

The fixed interface normal modes of a modified substructure are found by solving

$$[(\mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}) - \omega_j^2(\mathbf{M}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{M}_{\mathrm{ii}})]\{\Phi_{\mathrm{i}}\}_j = 0 \quad j = 1, 2, \ldots, N_{\mathrm{T}} \tag{2.19}$$

where $\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}$, $\boldsymbol{\Delta}\mathbf{M}_{\mathrm{ii}}$ stand for the introduced modifications on $\mathbf{K}_{\mathrm{ii}}$ and $\mathbf{M}_{\mathrm{ii}}$ and, ω_j, $\{\Phi_{\mathrm{i}}\}_j$ are the jth natural frequency and the corresponding mode shape of a modified substructure, respectively.

Rearranging Equation (2.19),

$$[\mathbf{K}_{\mathrm{ii}} - \omega_j^2\mathbf{M}_{\mathrm{ii}}]\{\Phi_{\mathrm{i}}\}_j = \mathbf{f}_{\boldsymbol{\Delta}}(\omega_j) \tag{2.20}$$

is obtained where

$$\mathbf{f}_{\boldsymbol{\Delta}}(\omega_j) = -[\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}} - \omega_j^2\boldsymbol{\Delta}\mathbf{M}_{\mathrm{ii}}]\{\Phi_{\mathrm{i}}\}_j \quad j = 1, 2, \ldots, N_{\mathrm{T}} \tag{2.21}$$

are the residual forces acting on the initial substructure due to the design modifications. These forces can be used to define a correction to the displacement field which can be employed to update the initial normal mode set. An approach to do this might be assuming that the inertia forces are negligible compared to the resisting elastic forces on the left-hand side of Equation (2.20). Thus, the residual displacement matrix (corrections)

$$\mathbf{R}_{\mathrm{D}} = [\mathbf{r}_{\mathrm{D}}(\omega_1), \mathbf{r}_{\mathrm{D}}(\omega_2), \ldots, \mathbf{r}_{\mathrm{D}}(\omega_{N_{\mathrm{T}}})]$$

is defined by calculating the static response of the substructure to the residual forces such that

$$\mathbf{R}_{\mathrm{D}} = \mathbf{K}_{\mathrm{ii}}^{-1}\mathbf{R}_{\mathrm{L}} \tag{2.22}$$

where

$$\mathbf{R}_{\mathrm{L}} = [\mathbf{f}_{\boldsymbol{\Delta}}(\omega_1), \mathbf{f}_{\boldsymbol{\Delta}}(\omega_2), \ldots, \mathbf{f}_{\boldsymbol{\Delta}}(\omega_{N_{\mathrm{T}}})]$$

is the residual force matrix whose columns are the residual force vectors $\mathbf{f}_{\boldsymbol{\Delta}}(\omega_j)$.

The residual displacement matrix $\mathbf{R}_{\mathrm{D}}$ can be utilized to update the fixed interface normal mode set of the initial model as explained below:

There is no information about the eigenvectors $\boldsymbol{\Phi}_{\mathrm{i}} = \{\{\Phi_{\mathrm{i}}\}_j,\ j = 1, 2, \ldots, N_{\mathrm{T}}\}$ of the modified component. Consequently, the residual forces $\mathbf{f}_{\boldsymbol{\Delta}}(\omega_j)$ in Equation (2.22) cannot be calculated exactly. Instead, they can be approximated using the fixed interface normal modes $\boldsymbol{\Phi}_{\mathrm{i}}^0 = \left\{\{\Phi_{\mathrm{i}}^0\}_j, \quad j = 1, 2, \ldots, N_{\mathrm{T}}\right\}$ and the corresponding eigenvalues $\boldsymbol{\Lambda}_{jj}^0 = \left\{\{\omega_j^0\}^2, \quad j = 1, 2, \ldots, N_{\mathrm{T}}\right\}$ of the initial model. Hence,

$$\mathbf{f}_{\boldsymbol{\Delta}}(\omega_j) \approx \hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_1)\, y_1 + \hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_2)\, y_2 + \ldots + \hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_{N_T})\, y_{N_T} \tag{2.23}$$

where

$$\hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_j) = -[\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}} - \{\omega_j^0\}^2 \boldsymbol{\Delta}\mathbf{M}_{\mathrm{ii}}]\{\Phi_{\mathrm{i}}^0\}_j \tag{2.24}$$

and $\mathbf{y}^{\mathrm{T}} = \{y_1, y_2, \ldots, y_{N_T}\}$ is a vector of unknown coefficients. The basis $\hat{\mathbf{R}}_{\mathrm{L}}$

$$\hat{\mathbf{R}}_{\mathrm{L}} = [\hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_1), \hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_2), \ldots, \hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_{N_{\mathrm{T}}})] \tag{2.25}$$

spans the subspace that contains the approximations of the residual force vectors. Equation (2.25) also defines the residual forces acting on the modified substructure due to the substitution of the fixed interface normal modes and the corresponding natural frequencies of the initial design into the new system of equations given in Equation (2.19). The essential idea behind Equation (2.23) is: if the subspace spanned by $\hat{\mathbf{R}}_{\mathrm{L}}$ does not contain the exact residual force vectors with respect to a specific design modification, it may at least contain a reasonable representation of these vectors. Since all the residual forces in $\mathbf{R}_{\mathrm{L}}$ are generated by the vectors of $\hat{\mathbf{R}}_{\mathrm{L}}$, $\hat{\mathbf{R}}_{\mathrm{L}}$ can be utilized to replace $\mathbf{R}_{\mathrm{L}}$ in Equation (2.22). The basis $\hat{\mathbf{R}}_{\mathrm{L}}$ is rarely of full rank and it needs to be reconditioned before being used [96] for which Singular Value Decomposition (SVD) can be employed [20].

The approximate residual displacement matrix is rewritten as

$$\tilde{\mathbf{R}}_{\mathrm{D}} = \mathbf{K}_{\mathrm{ii}}^{-1} \tilde{\mathbf{R}}_{\mathrm{L}} \tag{2.26}$$

where $\tilde{\mathbf{R}}_{\mathrm{L}}$ consists of the reconditioned basis vectors of $\hat{\mathbf{R}}_{\mathrm{L}}$. The matrix $\tilde{\mathbf{R}}_{\mathrm{D}}$ is employed to enrich the initial fixed interface normal mode set.

Hence, the extended mode set gets the form of

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_{\mathrm{i}}^0 & \tilde{\mathbf{R}}_{\mathrm{D}} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2.27}$$

which can then be used in the CB transformation matrix for the condensation of the modified component.

The ECB method is based on two main assumptions. The first one is neglecting the inertia forces in Equation (2.20) for the computation of $\mathbf{R}_{\mathrm{D}}$. Although this can be valid in low frequency range, it may cause problems for higher frequencies. The second assumption is to define each $\mathbf{f}_{\boldsymbol{\Delta}}(\omega_j)$ as a linear combination of $\hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\omega_j)$ where $j = 1, 2, \ldots, N_{\mathrm{T}}$ (see Equation (2.23)). If this condition does not hold, the basis vectors $\tilde{\mathbf{R}}_{\mathrm{D}}$ may not be able to describe the dynamic behavior of a modified component when it is added to the normal mode set of the initial substructure.

2.3.2 Updating the Constraint Modes

The constraint modes of a modified substructure are found by solving

$$\begin{bmatrix} \mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}} & \mathbf{K}_{\mathrm{ib}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}} \\ \mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}} & \mathbf{K}_{\mathrm{bb}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bb}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_{\mathrm{ib}}^0 + \boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} \\ \mathbf{I}_{\mathrm{bb}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{\mathrm{ib}} \\ \mathbf{R}_{\mathrm{bb}}^0 + \boldsymbol{\Delta}\mathbf{R}_{\mathrm{bb}} \end{bmatrix} \tag{2.28}$$

where $\boldsymbol{\Delta\Psi}_{\mathrm{ib}}$, $\boldsymbol{\Delta R}_{\mathrm{bb}}$ stand for the modifications in the initial constraint modes and the reaction force matrix due to the introduced perturbations on the design variables, respectively. Using the first line of Equation (2.28) and taking into account that $\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}^0_{\mathrm{ib}} + \mathbf{K}_{\mathrm{ib}} = \mathbf{0}$, $\boldsymbol{\Delta\Psi}_{\mathrm{ib}}$ can be calculated solving

$$(\mathbf{I} + \mathbf{K}_{\mathrm{ii}}^{-1}\boldsymbol{\Delta K}_{\mathrm{ii}})\boldsymbol{\Delta\Psi}_{\mathrm{ib}} + \mathbf{K}_{\mathrm{ii}}^{-1}(\boldsymbol{\Delta K}_{\mathrm{ii}}\boldsymbol{\Psi}^0_{\mathrm{ib}} + \boldsymbol{\Delta K}_{\mathrm{ib}}) = \mathbf{0}. \tag{2.29}$$

The matrix $\boldsymbol{\Delta\Psi}_{\mathrm{ib}}$ has a size $N_{\mathrm{i}} \times N_{\mathrm{s}}$ where N_{i} is the number of the internal d.o.f. and N_{s} is the number of the constraint modes in a component. For structures which have to be reanalyzed several times, the calculations of the exact residual constraint modes $\boldsymbol{\Delta\Psi}_{\mathrm{ib}}$ might be computationally demanding. In order to reduce the overall computational costs, the objective is to employ reanalysis methods to prevent solving the complete set of equations. Local approximations can be used for the solution of such systems. The binomial series expansion is one of the most common.

Binomial Series Expansion for Constraint Modes

Using the binomial series expansion of $(\mathbf{I} + \mathbf{K}_{\mathrm{ii}}^{-1}\boldsymbol{\Delta K}_{\mathrm{ii}})^{-1}$ in Equation (2.29), $\boldsymbol{\Delta\Psi}_{\mathrm{ib}}$ can be approximated as,

$$\boldsymbol{\Delta\Psi}_{\mathrm{ib}} \approx (\mathbf{I} - \mathbf{B} + \mathbf{B}^2 - \ldots)\boldsymbol{\Delta r}_1 = \boldsymbol{\Delta r}_1 + \boldsymbol{\Delta r}_2 + \boldsymbol{\Delta r}_3 + \ldots \tag{2.30}$$

where

$$\mathbf{B} = \mathbf{K}_{\mathrm{ii}}^{-1}\boldsymbol{\Delta K}_{\mathrm{ii}},\ \boldsymbol{\Delta r}_1 = \mathbf{K}_{\mathrm{ii}}^{-1}\mathbf{R},$$
$$\mathbf{R} = -\boldsymbol{\Delta K}_{\mathrm{ii}}\boldsymbol{\Psi}^0_{\mathrm{ib}} - \boldsymbol{\Delta K}_{\mathrm{ib}},\ \boldsymbol{\Delta r}_k = -\mathbf{B}\boldsymbol{\Delta r}_{k-1}\ \ k = 2, 3, \ldots$$

The matrix $\boldsymbol{\Delta r}_k$ has a size of $N_{\mathrm{i}} \times N_{\mathrm{s}}$. The tth column, $\{\boldsymbol{\Delta r}_k\}_t$, of the matrix corresponds to the kth term of the binomial series for the tth constraint mode. Therefore, the tth residual constraint mode $\{\boldsymbol{\Delta\Psi}_{\mathrm{ib}}\}_t$ is defined as

$$\{\boldsymbol{\Delta\Psi}_{\mathrm{ib}}\}_t \approx \{\boldsymbol{\Delta r}_1\}_t + \{\boldsymbol{\Delta r}_2\}_t + \{\boldsymbol{\Delta r}_3\}_t + \ldots. \tag{2.31}$$

In the expansion, the high-order binomial series terms can easily be computed with forward and backward substitution. Since $\mathbf{K}_{\mathrm{ii}}$ belongs to the initial substructure, its inverse needs not to be calculated again.

As reported in [83], a sufficient criterion for the convergence of $(\mathbf{I} - \mathbf{B} + \mathbf{B}^2 - \ldots)$ is $\|\mathbf{B}\| \leq 1$ where $\|.\|$ stands for the norm. On the other hand, for large changes in $\mathbf{K}_{\mathrm{ii}}$, the elements of $\mathbf{B}$ get very large values and the series diverge from the actual solution which causes inaccurate predictions of $\boldsymbol{\Delta\Psi}_{\mathrm{ib}}$.

To summarize, the binomial series expansion only uses the information of the initial model and the accuracy of its results is highly based on the introduced perturbations on the initial design variables. There exist several methods, such as the Jacobi iteration, the block Gauss-Seidel iteration, the dynamic acceleration and scaling of the initial design, to improve the convergence properties of the series [81].

Combined Approximations Approach for Constraint Modes

In this research, the Combined Approximations (CA) approach is proposed for updating the CB constraint modes [81, 82]. The idea behind the approach is: conditioning the binomial series terms $\{\boldsymbol{\Delta}\mathbf{r}_k\}_t$ in Equation (2.31) for each residual constraint mode so that these terms are banned to get high values. Accordingly, the divergence of the series and the inaccuracies in the approximation of the residual constraint modes are prevented.

The update procedure is as follows:

The tth residual constraint mode $\{\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\}_t$ is approximated in the space spanned by the vectors of the basis $\mathbf{H}_t$,

$$\mathbf{H}_t = [\{\boldsymbol{\Delta}\mathbf{r}_1\}_t, \{\boldsymbol{\Delta}\mathbf{r}_2\}_t, \ldots, \{\boldsymbol{\Delta}\mathbf{r}_{N_b}\}_t], \quad t = 1, 2, \ldots, N_{\mathrm{s}} \tag{2.32}$$

where

$$\{\boldsymbol{\Delta}\mathbf{r}_1\}_t = \mathbf{K}_{\mathrm{ii}}^{-1}\mathbf{R}_t, \qquad \{\boldsymbol{\Delta}\mathbf{r}_k\}_t = -\mathbf{B}\{\boldsymbol{\Delta}\mathbf{r}_{k-1}\}_t \quad k = 2, 3, \ldots, N_{\mathrm{b}},$$

$\mathbf{R}_t$ is the tth column of $\mathbf{R}$ and N_{b} is the total number of the binomial series terms used in the approximation. It is important to point out that each binomial series term $\boldsymbol{\Delta}\mathbf{r}_k, \quad k = 2, 3, \ldots, N_{\mathrm{b}}$ is calculated using the knowledge of the previous one. Hence, when a new term has to be added into the available basis, the last term of this set is sufficient to compute the new one. This is a nice computational advantage. To prevent numerical errors due to the large elements of $\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}$, the basis $\mathbf{H}_t$ should be normalized. A simple way of doing that is dividing each column of $\mathbf{H}_t$ by an arbitrary reference entry of the corresponding column [83].

Having defined the basis $\mathbf{H}_t$, $\{\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\}_t$ can be approximated as

$$\{\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\}_t \approx \{\boldsymbol{\Delta}\mathbf{r}_1\}_t y_{t,1} + \{\boldsymbol{\Delta}\mathbf{r}_2\}_t y_{t,2} + \ldots + \{\boldsymbol{\Delta}\mathbf{r}_{N_{\mathrm{b}}}\}_t y_{t,N_{\mathrm{b}}} = \mathbf{H}_t\mathbf{y}_t \tag{2.33}$$

where $\mathbf{y}_t^{\mathrm{T}} = \{y_{t,1}, y_{t,2}, \ldots, y_{t,N_{\mathrm{b}}}\}$ is a vector of unknown coefficients. Substituting Equation (2.33) into Equation (2.29) and pre-multiplying the equation by $\mathbf{H}_t^{\mathrm{T}}$ gives a linear system of N_{b} equations

$$[\mathbf{H}_t^{\mathrm{T}}(\mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}})\mathbf{H}_t]\mathbf{y}_t = \mathbf{H}_t^{\mathrm{T}}(-\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}\{\boldsymbol{\Psi}_{\mathrm{ib}}^0\}_t - \{\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}}\}_t) \tag{2.34}$$

where $N_{\mathrm{b}} \ll N_{\mathrm{i}}$.

When Equation (2.34) is solved for $\mathbf{y}_t$ and the solution is inserted back into Equation (2.33), the tth residual constraint mode $\{\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\}_t$ is computed approximately. When the above defined operations are performed for each residual constraint mode, the CA approach of the residual constraint mode matrix $\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}$ is defined as

$$\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} = [\{\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\}_1, \{\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\}_2, \ldots, \{\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\}_{N_s}]. \tag{2.35}$$

Hence, the approximate constraint mode matrix is given by

$$\boldsymbol{\Psi} \approx \begin{bmatrix} \boldsymbol{\Psi}_{\mathrm{ib}}^0 + \boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} \\ \mathbf{I}_{\mathrm{bb}} \end{bmatrix}. \tag{2.36}$$

The above procedures do not change the number of the constraint modes which means no extra basis elements are added to the CB transformation matrix as in the ECB method. The advantage of the CA approach is that efficient local approximations (series expansion) are combined with accurate global approximations (the reduced basis method) for an effective solution procedure. The solutions of the CA approach are exact when an included new basis vector is a linear combination of the previous basis vectors [83, 86, 87].

The number of **FL**oating-point **OP**erations (FLOPs) required by the exact analysis and the CA approach are roughly defined in Table 2.1. The symmetry and the sparsity of the matrices are not taken into account during the FLOP count.

Table 2.1: Number of FLOPs in exact analysis and the CA approach.

	Exact Analysis	CA Approach
Matrix Factorization	$\frac{2}{3}(N_\mathrm{i})^3$ [Eq. (2.29)]	$\frac{2}{3}N_\mathrm{s}(N_\mathrm{b})^3$ [Eq. (2.34)]
Forward-Backward substitution	$2\,N_\mathrm{s}(N_\mathrm{i})^2$ [Eq. (2.29)]	$2\,N_\mathrm{s}(N_\mathrm{b})^2 + N_\mathrm{s}N_\mathrm{i}(2N_\mathrm{b}-1)$ [Eqs. (2.33,2.34)]
Basis calculation	-	$N_\mathrm{s}N_\mathrm{i}(N_\mathrm{b}-1)(2N_\mathrm{i}-1)$ [Eq. (2.32)]
Reduction	-	$N_\mathrm{s}N_\mathrm{b}(2N_\mathrm{i}-1)(1+N_\mathrm{i}+N_\mathrm{b})$ [Eq. (2.34)]

Table 2.1 can be used to have a general idea about the computational costs assigned to both the exact analysis and the CA approach. The number of FLOPs is governed by three measures: The size N_i of the full component stiffness matrix corresponding to the internal nodes, the number of the constraint modes, N_s, and the number of the binomial series terms (basis vectors), N_b.

It is possible to identify the number of the basis vectors in $\mathbf{H}_t$ which makes the computation time of the CA approach equal to that of the exact analysis by the formulas in Table 2.1. This relationship is shown in Figure 2.1 for large range of N_i values. The legends in the figure stand for various N_i values. As observed in the figure, the correlation between $N_\mathrm{i}/N_\mathrm{s}$ and N_b is linear. The dominating factor on the computational efficiency is the ratio between the size of the internal stiffness matrix and the number of the constraint modes. The higher this ratio gets, the more basis vectors can be used in the CA approach. If the number of the binomial series terms, which provides satisfactory approximation, is below the linear curve, the CA approach is computationally more efficient than the exact analysis. If it is on the curve, the computation time of the constraint modes by the exact analysis and the CA approach are similar. It is important to emphasize again that the number of FLOPs strongly depends on the type of the solver. Hence, Table 2.1 only gives an idea about the computational requirements of these methods.

Generally, a few basis vectors are enough to give a good approximation of the

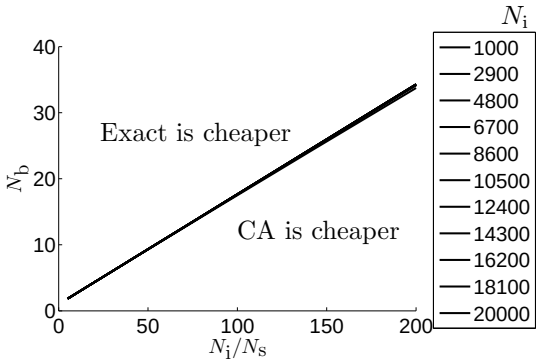


Figure 2.1: Comparison of the computational efficiency of the exact analysis and the CA approach. Relation between the size of the internal full stiffness matrix, the number of the constraint modes and the number of the binomial series terms.

constraint modes. Therefore, for the applications in which the ratio $N_\mathrm{i}/N_\mathrm{s}$ is high, the computational efficiency of the CA approach is more pronounced.

Based on the selected application, only the constraint modes can be updated using the CA approach as well as both the ECB and the CA methods can be employed at the same time. In the latter case the transformation matrix of the modified substructure gets the form of

$$\mathbf{T}_{\mathrm{ECB\&CA}} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Psi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_\mathrm{i}^0 & \tilde{\mathbf{R}}_\mathrm{D} & \boldsymbol{\Psi}_\mathrm{ib}^0 + \boldsymbol{\Delta\Psi}_\mathrm{ib} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_\mathrm{bb} \end{bmatrix} \tag{2.37}$$

Evaluation of the Error & Automatizing the Reanalysis

The accuracy of the approximated constraint modes can be analyzed by inserting Equation (2.35) into Equation (2.29). If $\boldsymbol{\Delta\Psi}_\mathrm{ib}$ is exact, it has to validate the right-hand side of the equation and the residuals must be zero. If not, the residuals are

$$\boldsymbol{\epsilon}_\mathrm{CA} = (\mathbf{I} + \mathbf{K}_\mathrm{ii}^{-1}\boldsymbol{\Delta K}_\mathrm{ii})\boldsymbol{\Delta\Psi}_\mathrm{ib} + \mathbf{K}_\mathrm{ii}^{-1}(\boldsymbol{\Delta K}_\mathrm{ii}\boldsymbol{\Psi}_\mathrm{ib}^0 + \boldsymbol{\Delta K}_\mathrm{ib}) \tag{2.38}$$

which define the errors due to the approximation.

It is possible to automatize the calculation of the constraint modes. To do that, first, a value is assigned to the initial number of the basis vectors in the CA approach. Next, the number of FLOPs is counted using Table 2.1. This number is compared with the number of FLOPs of the exact analysis. The CA approach is used only when it requires less FLOPs than the exact analysis. If it is computationally efficient to employ, the residual constraint mode matrix $\boldsymbol{\Delta\Psi}_\mathrm{ib}$ is calculated using CA. The accuracy of the approximation is verified by Equation (2.38). If the accuracy is not satisfactory and the number of FLOPs of CA is still less than that of the exact analysis when a new vector is added to the basis $\mathbf{H}_t$; $\mathbf{H}_t$ is extended with this vector. The

reanalysis is performed again. Otherwise, the constraint modes are computed with the exact analysis.

2.4 Demonstration of the Concepts and Discussions

For the demonstration of the introduced concepts, an academic test problem shown in Figure 2.2(b) is selected. The structure is composed of 8 identical components and it is free at the boundaries. A component consists of a cylinder skin including a floor panel, frames and stiffeners whose geometry is as illustrated in Figure 2.2(a).

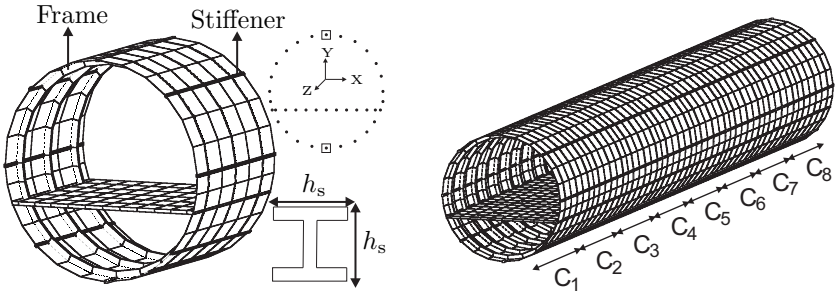


Figure 2.2: Test problem. (a) Component model and its design variable, (b) Structure model and the corresponding components C_i, $i = 1, 2, \ldots, 8$.

The reduced system matrices of the entire structure are obtained by only modeling one component. The FE model of a component is generated in the commercial FE software ANSYS. Its system matrices are calculated for the defined design variables and then they are transferred to MATLAB. For obtaining the reduced system matrices of the components, first of all, the transformation matrices are computed and afterwards, condensation is performed. In the transformation matrices 18 fixed interface normal modes are used. The number of the nodes on one interface of a component is 37. After the reduced matrices of all the components are obtained, these matrices are assembled and the reduced system matrices of the entire structure are gathered.

The skin, floor and frames are modeled using a 4-node shell element which has 6 d.o.f. at each node and is suitable to analyze thin to moderately thick shell structures [7]. The stiffeners with *I cross-section* are modeled with a three dimensional beam element which has 6 d.o.f. at each node. It allows different cross sections and permits the end nodes to be offset from the centroidal axes of the beam [7]. The cross section width and height of the stiffeners (h_s) in the components (see Figure 2.2(a)) are defined as the design variables and all the stiffeners of a component are assumed to have the same design values. Therefore, there exist 8 design variables in total in the overall structure. Each component has one design variable. For the initial design h_{s_i}, $i = 1, 2, \ldots, 8$ are set to 0.05m. The design variables are modified with the same amount of perturbation in the interval of [0.01 0.15].

Since the structure is free at the boundaries, its first 6 modes are rigid body modes. The 1st, 3rd and 4th dynamic modes are bending modes and the 2nd one is a torsion mode which are shown in Figure 2.3.

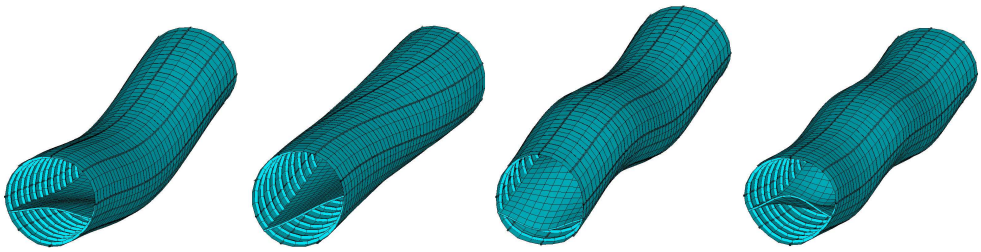


Figure 2.3: The first four flexible dynamic modes.

The highest rigid body frequency and the frequencies of the first 4 flexible dynamic modes that correspond to several design configurations are presented in Table 2.2. These results are computed by the full FE analysis.

Table 2.2: Frequencies corresponding to different design configurations.

h_{s_i}(m), $i = 1, 2, \ldots, 8$	ω_6(Hz)	ω_7(Hz)	ω_8(Hz)	ω_9(Hz)	ω_{10}(Hz)
0.01	$6.69\,10^{-4}$	16.40	26.49	29.42	38.836
0.05 (Initial Design)	$6.27\,10^{-4}$	21.65	26.69	32.96	41.14
0.1	$6.57\,10^{-4}$	25.01	28.21	35.73	43.36
0.15	$6.10\,10^{-4}$	27.49	30.67	37.85	44.98

2.4.1 Validation of the Update Scheme

The proposed reanalysis method is validated on the basis of the first four flexible dynamic modes. Since rigid body modes are defined by the constraint modes, their correct representation is also inspected.

The natural frequencies and the corresponding mode shapes are computed for different values of the design variables using the methods:

- **Full**: The full FE analysis. All d.o.f. are considered in the analysis without any reduction.
- **CB**: The Craig-Bampton method. The total number of d.o.f. of the modified components are reduced using the CB reduction basis. The fixed interface normal modes and the constraint modes are computed by the exact analysis.

- $\mathbf{T}^0_{\mathrm{CB}}$: The Craig-Bampton method. The total number of d.o.f. of the modified components are reduced using the reduction basis of the initial ones.

- **ECB**: The Craig-Bampton method. An approximate reduction basis is employed for the modified components. The fixed interface normal modes are approximated using the ECB method. The constraint modes of the initial substructure are used as the constraint modes of the modified one.

- **ECB**&**CM**$_{\mathrm{Exact}}$: The Craig-Bampton method. An approximate reduction basis is employed for the modified components. The fixed interface normal modes are approximated using the ECB method. The constraint modes are computed using the exact analysis.

- **ECB**&**BS**: The Craig-Bampton method. An approximate reduction basis is employed for the modified components. The fixed interface normal modes are approximated using the ECB method. The constraint modes are approximated using a three term binomial series expansion.

- **ECB**&**CA**: The Craig-Bampton method. An approximate reduction basis is employed for the modified components. The fixed interface normal modes are approximated using the ECB method. The constraint modes are approximated using the CA approach. For CA three cases; 3 binomial series terms (B3), 4 binomial series terms (B4) and 7 binomial series terms (B7); are taken into account in the approximation.

The size of the system matrices of the components is summarized in Table 2.3.

Table 2.3: The size of the system matrices of the components.

	Component $1, 8$	Component $2, \ldots, 7$
# of interface d.o.f.	222	444
Full	1986×1986	1986×1986
CB, $\mathbf{T}^0_{\mathrm{CB}}$	240×240	462×462
ECB	258×258	480×480
ECB&CM$_{\mathrm{Exact}}$	258×258	480×480
ECB&BS	258×258	480×480
ECB&CA	258×258	480×480

When the number of FLOPs is counted using the equations given in Table 2.1, it is observed that the selected structure is actually not suitable for demonstrating the computational efficiency of the CA approach. FLOPs required in the exact analysis are almost the same as those required in the CA approach with 3 binomial series terms. Therefore, in this example only the accuracy of the CA approach will be discussed.

The accuracy of the results is compared on the basis of the relative frequency error (ε_ω) and the Modal Assurance Criterion (MAC).

For the calculation of ε_ω, the full FE results are compared with those of the rest of the methods using

$$\varepsilon_\omega = \frac{|[\mathbf{\Lambda}_{\text{Full}}]_{jj}^{\frac{1}{2}} - [\mathbf{\Lambda}_{\text{M}}]_{jj}^{\frac{1}{2}}|}{[\mathbf{\Lambda}_{\text{Full}}]_{jj}^{\frac{1}{2}}} \quad j = 1, \ldots, 4$$

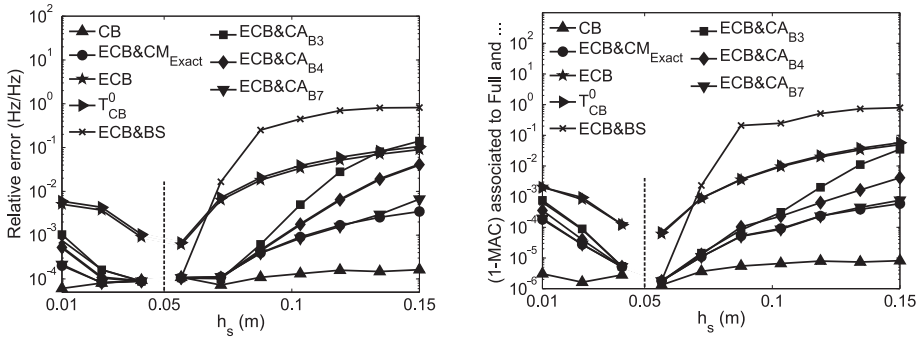
where M is a generic abbreviation used for representing the methods defined above, $|\cdot|$ is the absolute value and $\mathbf{\Lambda}_{jj}$ is the jth diagonal entry of the spectral matrix $\mathbf{\Lambda}$.

MAC is a scalar value between 0 and 1, and it represents the correlation number between the two mode shapes. A MAC value close to 1 indicates a high degree of correlation between two mode shapes. For its calculation, first the reduced model solutions are expanded to their full forms and then these eigenvectors are compared with the eigenvectors calculated by full FE analysis using

$$\text{MAC} = \frac{([\tilde{\Phi}_{\text{Full}}]_j^{\text{T}}[\tilde{\Phi}_{\text{M}}]_j)^2}{([\tilde{\Phi}_{\text{Full}}]_j^{\text{T}}[\tilde{\Phi}_{\text{Full}}]_j)([\tilde{\Phi}_{\text{M}}]_j^{\text{T}}[\tilde{\Phi}_{\text{M}}]_j)} \quad j = 1, \ldots, 4 \tag{2.39}$$

where $[\tilde{\Phi}_{\text{Full}}]_j, [\tilde{\Phi}_{\text{M}}]_j$ are the jth eigenvectors corresponding to the full FE analysis and one of the above-mentioned methods, respectively.

The results of the test problem are presented in Figures 2.4 - 2.7. The highest computed rigid body frequencies using the above defined methods, for varying design values, are plotted in Figure 2.8.

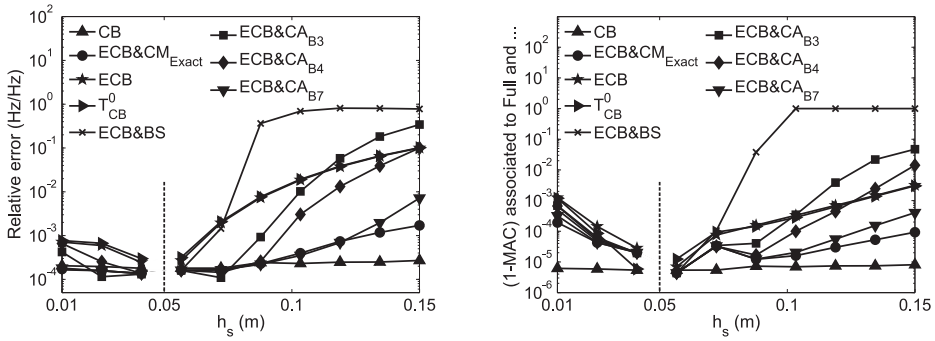


(a) Relative frequency error.

(b) Correlation of the mode shapes.

Figure 2.4: 1st flexible dynamic mode errors for varying designs. The dashed line shows the location of the initial design value.

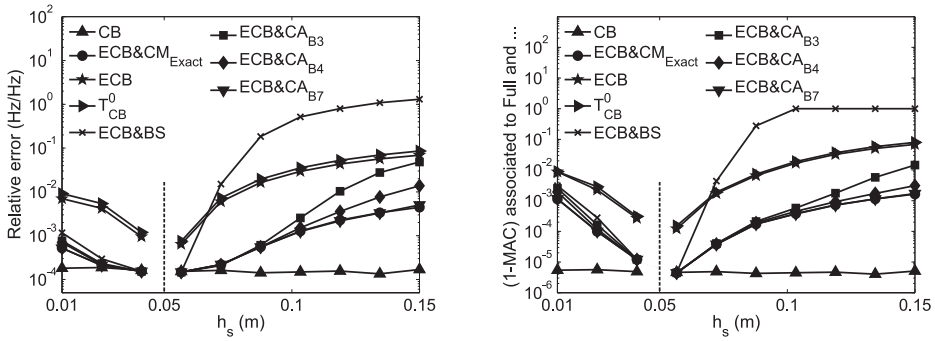
As observed from the results, the most accurate method is **CB**. This is expected because the fixed interface normal modes and the constraint modes are computed by exact analyses.



(a) Relative frequency error.

(b) Correlation of the mode shapes.

Figure 2.5: 2nd flexible dynamic mode errors for varying designs. The dashed line shows the location of the initial design value.



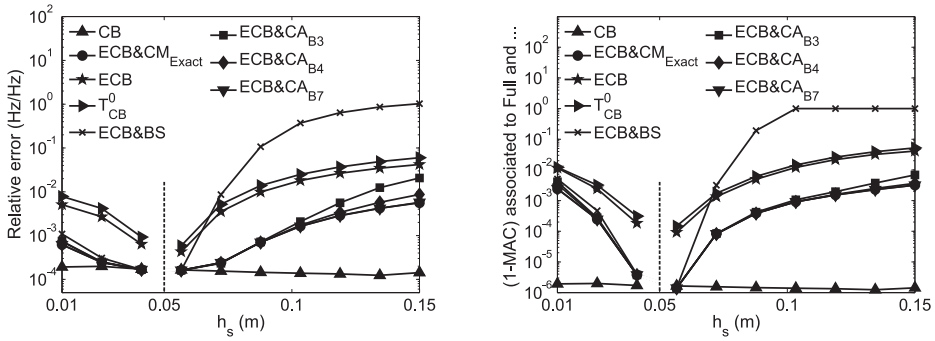
(a) Relative frequency error.

(b) Correlation of the mode shapes.

Figure 2.6: 3rd flexible dynamic mode errors for varying designs. The dashed line shows the location of the initial design value.

The difference between the relative error of **CB** and **ECB**&**CM**$_{\text{Exact}}$ is due to the ECB approximation. As seen in the figures, the higher the perturbation on the initial design gets, the less accurate the approximation becomes. This is because of the assumptions made during the computation of the additional basis vectors $\tilde{\mathbf{R}}_{\text{D}}$ (see Section 2.3.1 for details). As the constraint modes are exact in **CB** and **ECB**&**CM**$_{\text{Exact}}$, the rigid body displacements can be represented correctly by these modes.

The accuracy of the natural frequencies computed by the **ECB** and the $\mathbf{T}^0_{\text{CB}}$ methods deteriorates with the amount of perturbation on the initial design values. In this example, the geometry does not change hence rigid body modes remain accurate for the entire perturbation range.



(a) Relative frequency error.

(b) Correlation of the mode shapes.

Figure 2.7: 4th flexible dynamic mode errors for varying designs. The dashed line shows the location of the initial design value.

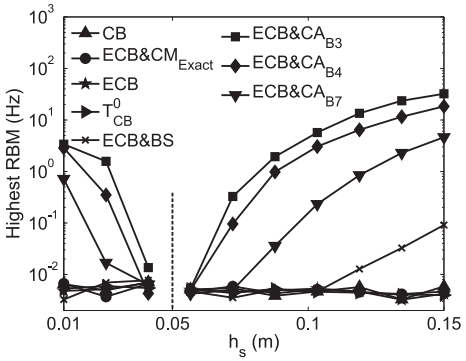


Figure 2.8: Maximum rigid body mode values for varying designs. The dashed line shows the location of the initial design value.

For the considered test problem, updating the constraint modes has a significant effect on the accuracy of the results. The fixed interface normal mode sets do not involve any bending modes that may contribute to describe the global dynamic behavior. In defining the bending modes of the structure, the influence of the constraint modes is more dominant. The torsion mode is determined by the motion of interface nodes which also requires knowledge of the constraint modes.

In the **ECB**&**BS** method, even though the constraint modes are updated, the accuracy of the results gets inadequate with the amount of perturbation on the initial model. This is because large changes in the initial stiffness matrix make the binomial series expansion diverge from the actual result as addressed in Section 2.3.2. On the other hand, the representation of rigid body modes are acceptable. This is related with the geometry not being modified with the perturbations. In the ideal situation, i.e. when there are no numerical errors, the calculation of rigid body modes is exact

with BS (see Appendix A.1 for details).

The accuracy of the natural frequencies calculated by **ECB**&**CA** is satisfactory for a wide range of perturbations. Unfortunately, it is not possible to make the same statement for rigid body modes as seen in Figure 2.8. However, with an increasing number of the basis vectors in CA, the representation of these modes as well as the accuracy of the flexible dynamic modes gets better. Consequently, evaluation of the error presented in Section 2.3.2 is very important for obtaining reliable results. As explained in Appendix A.2, when this error converges to zero, the constraint modes computed by the CA approximation converge to the exact ones. Hence, the rigid body displacements can be computed accurately. This is investigated in Section 2.4.2.

2.4.2 Results of the Automated Update Scheme

Exact analysis of the constraint modes is computationally more efficient than the CA approach for the selected structure. In order to still show the influence of the number of the binomial series terms in CA on the accuracy of the results and to demonstrate the automated decision of this number, the FLOP count option is ignored in the automated update scheme (see Section 2.3.2 for details). The constraint modes are computed only with the CA approach and the minimum number of the basis vectors is set to 3. The accuracy of the modes is verified using Equation (2.38). If the accuracy is not satisfactory, the set of the CA basis vectors is extended with a new vector and the reanalysis step is repeated. The tolerance value for the maximum norm of $\boldsymbol{\epsilon}_{\mathrm{CA}}$ in Equation (2.38) is set to 0.001 which is relative with respect to the maximum norm of the displacements defined on the interface nodes.

The solutions corresponding to the automated update scheme are plotted in Figures (2.9) - (2.10).

The number of the basis vectors utilized in $\mathbf{CA}_{\mathrm{automated}}$ in order to reach the requested tolerance is summarized in Table 2.4 for several designs. As observed from these results, in the neighborhood of the initial design, the required number is small and it increases with the magnitude of perturbation.

The highest accuracy that can be obtained using $\mathbf{ECB}\&\mathbf{CA}_{\mathrm{automated}}$ can only be as good as that of the $\mathbf{ECB}\&\mathbf{CM}_{\mathrm{Exact}}$. As seen in the figures, the errors in these methods are very close. This indicates that the constraint modes computed by the CA approximation converge to the exact ones when the error $\boldsymbol{\epsilon}_{\mathrm{CA}}$ in Equation (2.38) is driven to zero. Hence, rigid body modes can be represented correctly by $\mathbf{ECB}\&\mathbf{CA}_{\mathrm{automated}}$ as shown in Figure 2.10.

2.5 Summary and Conclusions

In this chapter, the Component Mode Synthesis (CMS) technique was introduced. Special attention was paid to the Craig-Bampton (CB) method. Next, the reanalysis methods that can be used under the framework of CB were discussed. The Enriched CB (ECB) method was studied in order to update the fixed interface normal mode

Table 2.4: The number of the basis vectors used in $\mathbf{CA}_{\text{automated}}$ for different designs.

h_{s_i}(m), $i = 1, 2, \ldots, 8$	Component $1, 8$	Component $2, \ldots, 7$
0.01	22	11
0.026	8	7
0.041	3	3
0.057	3	3
0.072	6	5
0.088	9	8
0.103	12	11
0.119	15	13
0.134	20	15
0.15	29	17

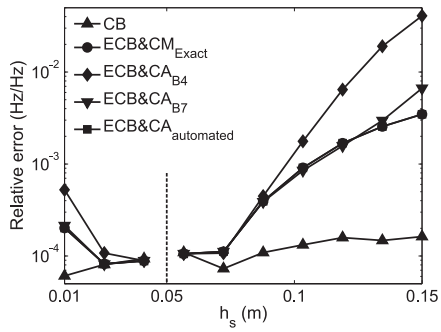
set of the initial CB reduction basis. A new method based on the Combined Approximations (CA) approach was proposed for approximating the constraint mode set of the modified substructures. An academic test problem was utilized for the demonstration of the presented concepts.

With the CB method, the analysis time can be reduced considerably. Additionally, it presents a number of interesting features that can be benefited in an optimization strategy. These are summarized in the following:

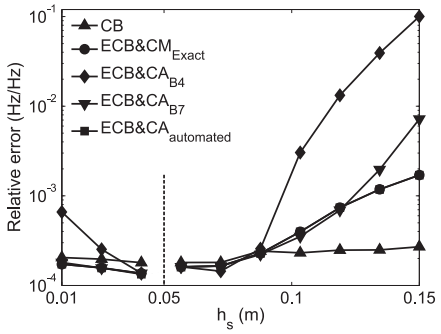
- A complex structure is divided into several substructures. The analysis of each component can be assigned to different groups and/or computers. Hence, parallel processing opportunities are highly supported.
- The number of d.o.f. in a large FE model is reduced significantly while the accuracy of the analysis is preserved within a low frequency range.
- Analyses of component models are independent from each other. When modifications are required in a certain component, only reanalyzing this part is sufficient to determine the modified matrices of the structure. The unmodified substructures do not have to be analyzed again.
- It is possible to use one parametric FE model for all the similar substructures. Thus, for structures which have repeating components in its geometry, the complete model need not to be generated.

The assembly of the reduced component models in CB are based on the compatibility of the interface nodes. Therefore, for the coupling of the substructures, it has to be ensured that the models have matching meshes on the interface surface. This can be considered as a drawback of the method.

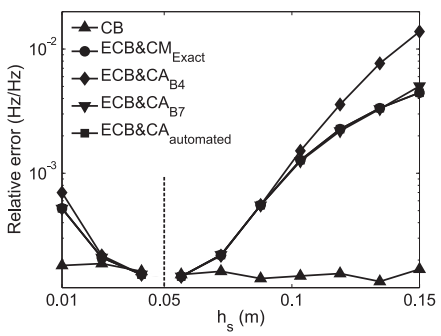
The solution of the free vibration problem can be avoided for the modified substructures using the ECB method. This saves computation time during repeated



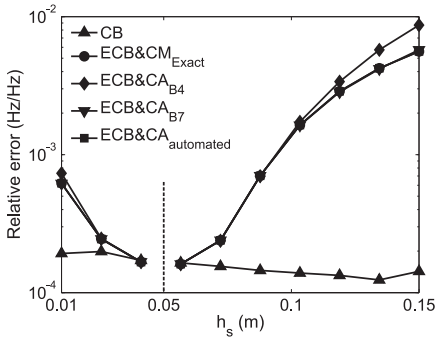
(a) 1st dynamic mode.



(b) 2nd dynamic mode.



(c) 3rd dynamic mode.



(d) 4th dynamic mode.

Figure 2.9: Relative frequency errors for varying designs. The dashed line shows the location of the initial design value.

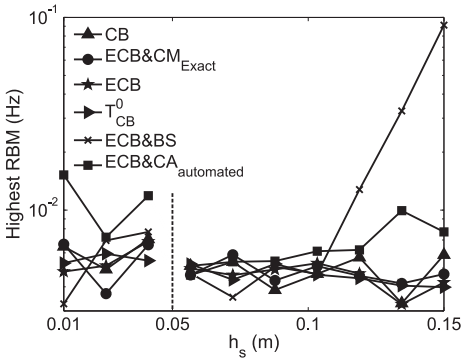


Figure 2.10: Maximum rigid body mode values for varying designs. The dashed line shows the location of the initial design value.

analysis of components. ECB adds extra basis vectors into the initial fixed interface normal mode set. Accordingly, the number of the generalized d.o.f. in the modified substructure is larger than that of the initial one. The assumptions that are made during the calculation of the additional basis vectors may hinder the accuracy of the ECB at higher frequencies and for large perturbations. There is no proposed automated update scheme in the literature that switches ECB with the exact analysis when the accuracy of the approximations are not satisfactory.

In many applications, the influence of the constraint modes is more dominant than that of the fixed interface normal modes on identifying the global dynamic behavior of a structure. As observed in the test problem, the reanalysis of only the fixed interface normal modes is not sufficient in terms of accuracy. Updating the constraint modes has a significant impact on the accuracy of the solutions.

The binomial series expansion leads to incorrect update of the constraint modes when the introduced perturbations on a substructure are large.

The results of the proposed method based on the CA approach are satisfactory for a wide range of perturbations. The method allows to calculate the residual errors due to the approximation of the constraint modes. With an increasing number of the basis vectors in CA, the method converges to the exact modes. The computational efficiency of the method depends on the properties of the FE models of the components. More specifically, the ratio between the size of the internal stiffness matrix and the number of the constraint modes is the driving factor. As this ratio gets larger, the method becomes more effective compared to the exact analysis. The presented automated update scheme allows to switch CA with the exact analysis automatically when the computational efficiency is lost.

Rigid body displacements of a structure are described by the constraint modes. For modifications in which the FE mesh is preserved (e.g. sizing optimization problems), the constraint modes of the initial substructures still represent rigid body modes of the new design. Even when the constraint modes are not updated correctly in the binomial series approximation, rigid body modes are still predicted accurately. In the CA method, rigid body modes are captured when a sufficient number of basis vectors are utilized. This implies that the calculated constraint modes have to be close to their exact values.

3

Numerical Optimization

In this chapter, first, the general formulation of the optimization problems concerned in the research is given. Next, the conditions that have to be fulfilled by any optimum corresponding to these problems are discussed. Afterwards, two numerical methods are presented and the strategies for finding the global optimum are introduced. Numerical test problems are used to demonstrate the concepts.

3.1 Optimization Problem Formulation

Maximization or minimization of a function subject to constraints on its variables is considered under the field of *numerical optimization*. An optimization problem can be written as

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \begin{cases} c_i(\mathbf{x}) = 0 & i \in \mathcal{E} \\ c_i(\mathbf{x}) \leq 0 & i \in \mathcal{I} \end{cases}$$

where $\mathbf{x}^{\mathrm{T}} = \{x_1, x_2, \ldots, x_n\}$ is a vector of variables, $\mathcal{I}$ and $\mathcal{E}$ are sets of indices and $f(\mathbf{x})$ is the *objective function* that is to be minimized. The *equality constraints* $c_i(\mathbf{x})$, $i \in \mathcal{E}$ and the *inequality constraints* $c_i(\mathbf{x})$, $i \in \mathcal{I}$ are the functions that the variables must satisfy. Both $f(\mathbf{x})$ and $c_i(\mathbf{x})$ return a scalar value. The set of points $\mathbf{x}$ satisfying all the constraints is called the *feasible set*, that is

$$\mathcal{F} = \{\mathbf{x} \,|\, c_i(\mathbf{x}) = 0 \;\; i \in \mathcal{E};\; c_i(\mathbf{x}) \leq 0 \;\; i \in \mathcal{I}\}$$

and all the points in this set are called the *feasible points*. For any $\mathbf{x} \in \mathcal{F}$, an inequality constraint $c_i(\mathbf{x})$, $i \in \mathcal{I}$ is said to be *active* if $c_i(\mathbf{x}) = 0$ and *inactive* if $c_i(\mathbf{x}) < 0$ [104]. The active set $\mathcal{A}(\mathbf{x})$ at this point is defined as

$$\mathcal{A}(\mathbf{x}) = \mathcal{E} \cup \{i \in \mathcal{I} \,|\, c_i(\mathbf{x}) = 0\}$$

which is the union of the indices of the equality constraints and the active inequality constraints.

A point $\mathbf{x}^*$ is referred to as a *local minimizer* if there is a neighborhood $\mathcal{N}(\mathbf{x}^*, \epsilon)$ of $\mathbf{x}^*$ and in that neighborhood $\mathbf{x}^*$ is a feasible point with the lowest objective function value:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{F} \cap \mathcal{N}(\mathbf{x}^*, \epsilon).$$

This point is a *global minimizer* if it has the lowest objective function value in the entire feasible set:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{F}.$$

An optimization problem which has multiple optimum solutions is called a *multi-modal* problem.

In some optimization problems, the variables make sense only when they take discrete values. Problems of this type are called *discrete optimization problems*. In contrast, variables of *continuous optimization problems* take real values. There are also problems which consist of both discrete and continuous valued variables. In this thesis, only the continuous optimization problems are in focus. Detailed information about other types of problems and the corresponding solution methods can be found in [15, 105].

A general formulation of the continuous optimization problems can be given as

$$\min_{\mathbf{x} \in \mathbb{R}^{n \times 1}} f(\mathbf{x}) \quad \text{subject to} \begin{cases} c_i(\mathbf{x}) = 0 & i \in \mathcal{E} \\ c_i(\mathbf{x}) \leq 0 & i \in \mathcal{I} \end{cases} \tag{3.1}$$

where $f(\mathbf{x})$ and $c_i(\mathbf{x})$ are all smooth (second order differentiable), real valued functions. There are many special cases of Equation (3.1) varying from each other by the selection of the type of the objective and the constraint functions. When at least one of these is a nonlinear function of the design variables, the problem is in the category of *Non-Linear Programming* (NLP). In this category, multi-modality often arises due to the nonlinearity of the objective and/or the constraint functions. This thesis concerns the solution of the NLP problems.

3.2 Necessary and Sufficient Conditions for Optimality

Before presenting the methods employed for the solution of Equation (3.1), first, *necessary* and *sufficient conditions* for optimality are briefly introduced. *Necessary conditions* are utilized to check whether a point $\mathbf{x}$ is the optimum of a given problem or not. Since an optimum can be a minimum, a maximum or a saddle point, *sufficient conditions* are used to define the type of it.

To be able to state that a point $\mathbf{x}^*$, selected in a feasible set

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^{n \times 1} \,|\, c_i(\mathbf{x}) = 0 \;\; i \in \mathcal{E};\; c_i(\mathbf{x}) \leq 0 \;\; i \in \mathcal{I}\},$$

is an optimum, the following conditions have to be fulfilled.

Let $\mathbf{x}^* \in \mathcal{F}$ and suppose that the gradients of the active constraints at $\mathbf{x}^*$, that is $\{\nabla c_i(\mathbf{x}^*),\, i \in \mathcal{A}(\mathbf{x}^*)\}$, are linearly independent. This condition is called *Linear*

Independence Constraint Qualification (LICQ). If $\mathbf{x}^*$ is an optimum, there exists a vector $\boldsymbol{\lambda}^* = \{\lambda_i^*,\ i \in \mathcal{E} \cup \mathcal{I}\}$ such that the following conditions are satisfied at $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) = \nabla_{\mathbf{x}} f(\mathbf{x}^*) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i^* \nabla_{\mathbf{x}} c_i(\mathbf{x}^*) = 0, \tag{3.2a}$$

$$c_i(\mathbf{x}^*) = 0, \quad \text{for all } i \in \mathcal{E} \tag{3.2b}$$

$$c_i(\mathbf{x}^*) \leq 0, \quad \text{for all } i \in \mathcal{I} \tag{3.2c}$$

$$\lambda_i^* \geq 0, \quad \text{for all } i \in \mathcal{I} \tag{3.2d}$$

$$\lambda_i^* c_i(\mathbf{x}^*) = 0, \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I} \tag{3.2e}$$

where $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \in \mathbb{R}^{n \times 1}$ is the gradient of the Lagrange function

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\mathbf{x}) \tag{3.3}$$

of Equation (3.1) and $\boldsymbol{\lambda}$ is a *Lagrange multiplier* vector.

The conditions defined in Equation (3.2) are known as the *Karush-Kuhn-Tucker* (KKT) conditions [104] and provide a foundation for many algorithms [49, 58, 114]. Derivation of these conditions can be found in [104].

Equation (3.2a) with the LICQ condition state that $\boldsymbol{\lambda}^*$ is uniquely defined at $\mathbf{x}^*$ and the gradient of the Lagrange function is zero for this pair. The constraints have to be fulfilled by $\mathbf{x}^*$ which is given in Equations (3.2b) and (3.2c). The fourth condition puts restrictions on the Lagrange multipliers of the inequality constraints. It ensures that there exists no feasible direction that can both reduce the objective function value and satisfy the constraints. According to the last condition, if any of the constraints are not active at $\mathbf{x}^*$, the corresponding Lagrange multiplier λ_i^* is zero. Hence, inactive constraints are ignored in the Lagrange function formulation.

Geometric interpretation of the KKT conditions is illustrated in Figure 3.1 for the case of three constraints. Let ∇c_1 and ∇c_2 denote the gradients of the constraints $c_1(\mathbf{x})$ and $c_2(\mathbf{x})$ which are perpendicular to the respective constraint surfaces at $\mathbf{x}^*$. The gradient of the objective function at $\mathbf{x}^*$ is represented by ∇f in the figure. In order to obtain a point which reduces the objective function value better than $\mathbf{x}^*$ and does not violate any of the constraints, a feasible direction vector $\mathbf{v}$ needs to be selected. This vector has to have an obtuse angle with ∇c_1 and ∇c_2 to stay in the feasible region. It must have an acute angle with $-\nabla f$ for decreasing the value of the objective function value even further. As clearly seen in the figure, there is no such direction. Hence, $\mathbf{x}^*$ is an optimum point. The signs of the Lagrange multipliers, λ_1 and λ_2, are important to have that conclusion which must be positive. Additionally, since the constraint function $c_3(\mathbf{x})$ is not active at $\mathbf{x}^*$, it is not taken into account for the decision of a feasible direction.

To summarize, if the KKT and the LICQ conditions hold at a point $\mathbf{x}^*$, any point selected in the neighborhood of this point will either violate the constraints or increase the value of the objective function.

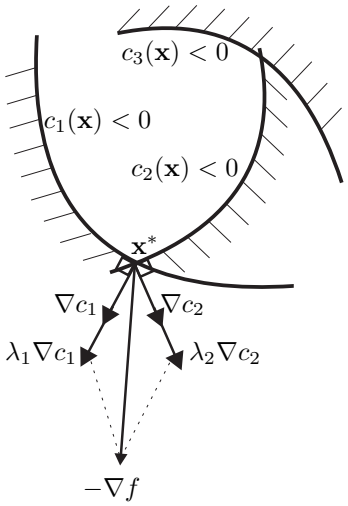


Figure 3.1: Geometric interpretation of KKT conditions.

A point $\mathbf{x}^*$ satisfying the KKT conditions is an optimum but does not necessarily have to be a minimum. If *sufficient conditions*; stated next; hold, this point is then a strict local minimum.

Suppose that for $\mathbf{x}^* \in \mathcal{F}$, there is a Lagrange multiplier vector $\boldsymbol{\lambda}^*$ such that the KKT conditions are satisfied. Suppose also that

$$\mathbf{d}^{\mathrm{T}} \nabla_{\mathbf{xx}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) \mathbf{d} > 0 \tag{3.4}$$

for every vector $\mathbf{d} \neq \mathbf{0}$ which satisfies $\nabla c_i(\mathbf{x}^*)^{\mathrm{T}} \mathbf{d} = 0$ for all $i \in \mathcal{A}(\mathbf{x}^*)$ with $\lambda_i^* > 0$. Then $\mathbf{x}^*$ is a strict local minimum [104].

3.3 Solution Algorithms

Numerical methods utilized for solving optimization problems can be divided into two categories, *gradient-based* and *derivative-free* [25]. Gradient-based methods are reliable, fast and robust, and commonly employed for solving continuous optimization problems [31, 104]. They can handle large numbers of variables and constraints. The computed optimum is exact and at least a local one. They are called *gradient-based* because the direction of the search is guided using the gradient information of the objective and the constraint functions. Derivative-free methods, as their name suggests, do not use gradient information during the search of an optimum. They are particularly attractive for solving problems having discrete variables, discontinuous or noisy function responses, or disconnected feasible domains. These further can be divided into *deterministic* and *stochastic* methods. Derivative-free deterministic methods always produce the same results given the same starting point [32]. Conversely, the stochastic ones use random numbers during their search [60] and

hence might lead to different solutions of the same problem. They search the design space globally, therefore they are also well-suited to solve multi-modal problems.

In the following sections, the Sequential Quadratic Programming (SQP) method from the gradient-based class and Genetic Algorithms (GAs) from the stochastic derivative-free class will be introduced briefly. The task of these methods during the solution phase of the NLP problems will be clarified in Section 3.4.

3.3.1 Sequential Quadratic Programming

The essential idea behind the Sequential Quadratic Programming (SQP) method is to model Equation (3.1) at the current iterate $\mathbf{x}_{(k)}$ by a Quadratic Programming[1] (QP) subproblem, and use the minimizer of this subproblem to define a new iterate $\mathbf{x}_{(k+1)}$ and accordingly a new subproblem. By repeating these steps, a sequence of iterates $\{\mathbf{x}_{(k)}\}_{k=0}^{\infty}$ is obtained which is terminated when either no more progress is obtained or a solution point has been approximated with sufficient accuracy.

To explain the theory behind the SQP method, first, the equality constrained problem

$$\min_{\mathbf{x}\in\mathbb{R}^{n\times 1}} f(\mathbf{x}) \quad \text{subject to } \mathbf{c}(\mathbf{x}) = 0 \tag{3.5}$$

is considered where $\mathbf{c}(x) = \{c_i(\mathbf{x}) = 0,\, i \in \mathcal{E}\}$ is a vector function. This will be extended for the general NLP problems next.

The simplest derivation of the SQP method can be realized as the application of the Newton's method to the KKT conditions given in Equation (3.2). An optimum solution of Equation (3.5) has to fulfill the system of equations

$$F(\mathbf{x}, \boldsymbol{\lambda}) = \begin{bmatrix} \nabla f(\mathbf{x}) + \mathbf{A}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\lambda} \\ \mathbf{c}(\mathbf{x}) \end{bmatrix} = \mathbf{0} \tag{3.6}$$

where $\mathbf{A}(\mathbf{x})^{\mathrm{T}} = \{\nabla c_i(\mathbf{x}),\, i \in \mathcal{E}\}$ is a matrix whose columns are the gradients of the constraints $c_i(\mathbf{x})$. Equation (3.6) is simply the KKT conditions defined for the problems with equality constraints where the Lagrange function is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \mathbf{c}(\mathbf{x})^{\mathrm{T}}\boldsymbol{\lambda}. \tag{3.7}$$

In Equation (3.6), $\mathbf{x}$ and $\boldsymbol{\lambda}$ are the unknowns that have to be solved. To find these unknowns, Newton's method linearizes $F(\mathbf{x}, \boldsymbol{\lambda})$ at $(\mathbf{x}_{(k)}, \boldsymbol{\lambda}_{(k)})$ using the first order Taylor series approximation, i.e.,

$$\begin{aligned} &F(\mathbf{x}_{(k+1)}, \boldsymbol{\lambda}_{(k+1)}) = F(\mathbf{x}_{(k)} + \mathbf{v}_{(k)}, \boldsymbol{\lambda}_{(k)} + \boldsymbol{\delta}_{(k)}) \\ &\approx \begin{bmatrix} \nabla f(\mathbf{x}_{(k)}) + \mathbf{A}(\mathbf{x}_{(k)})^{\mathrm{T}}\boldsymbol{\lambda}_{(k)} \\ \mathbf{c}(\mathbf{x}_{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{W}_{(k)} & \mathbf{A}(\mathbf{x}_{(k)})^{\mathrm{T}} \\ \mathbf{A}(\mathbf{x}_{(k)}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{(k)} \\ \boldsymbol{\delta}_{(k)} \end{bmatrix} = \mathbf{0} \end{aligned} \tag{3.8}$$

[1] In Quadratic Programming, the constraints are the linear and the objective function is the quadratic functions of the optimized variables.

and finds $(\mathbf{v}_{(k)}, \boldsymbol{\delta}_{(k)})$ that solves Equation (3.8). In the formulation $\mathbf{W}_{(k)}$ is either the Hessian of the Lagrangian function with respect to $\mathbf{x}$; that is $\mathbf{W}_{(k)} = \nabla_{\mathbf{xx}}\mathcal{L}(\mathbf{x}_{(k)}, \boldsymbol{\lambda}_{(k)})$; or an approximation of it. The next iteration point, then, is defined as

$$\begin{bmatrix} \mathbf{x}_{(k+1)} \\ \boldsymbol{\lambda}_{(k+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{(k)} \\ \boldsymbol{\lambda}_{(k)} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{(k)} \\ \boldsymbol{\delta}_{(k)} \end{bmatrix}.$$

The solution of the successive linearized problems converges to the solution only when the KKT matrix

$$\begin{bmatrix} \mathbf{W}_{(k)} & \mathbf{A}(\mathbf{x}_{(k)})^{\mathrm{T}} \\ \mathbf{A}(\mathbf{x}_{(k)}) & \mathbf{0} \end{bmatrix}$$

is nonsingular.

Nonsingularity is a consequence of the following conditions [104]:

1. $\mathbf{A}(\mathbf{x}_{(k)})$ has full row rank. This is the LICQ condition.
2. $\mathbf{W}_{(k)}$ is positive definite on the null space of the gradients of the active constraints. That is, $\mathbf{d}^{\mathrm{T}}\mathbf{W}_{(k)}\mathbf{d} > 0$ for all $\mathbf{d} \neq \mathbf{0}$ such that $\mathbf{A}(\mathbf{x}_{(k)})\mathbf{d} = 0$. This condition holds when $(\mathbf{x}_{(k)}, \boldsymbol{\lambda}_{(k)})$ is close to a minimum point $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$.

There is an alternative way to reach Equation (3.8). Instead of linearizing Equation (3.6) around a given iterate, in Equation (3.5) the constraints are replaced by their linear first order Taylor series approximation and the objective function by its second order Taylor series approximation augmented by a second order information from the constraints at $(\mathbf{x}_{(k)}, \boldsymbol{\lambda}_{(k)})$. Hence, a QP problem

$$\begin{aligned} &\min_{\mathbf{v}_{(k)}} \left[\frac{1}{2}\mathbf{v}_{(k)}^{\mathrm{T}}\mathbf{W}_{(k)}\mathbf{v}_{(k)} + \nabla f(\mathbf{x}_{(k)})^{\mathrm{T}}\mathbf{v}_{(k)}\right] \\ &\text{subject to } \mathbf{A}(\mathbf{x}_{(k)})\,\mathbf{v}_{(k)} + \mathbf{c}(\mathbf{x}_{(k)}) = \mathbf{0} \end{aligned} \tag{3.9}$$

is obtained. If the above defined conditions are satisfied, Equation (3.9) has a unique solution $(\mathbf{v}_{(k)}, \boldsymbol{\mu}_{(k)})$ satisfying the KKT conditions

$$\begin{aligned} &\mathbf{W}_{(k)}\mathbf{v}_{(k)} + \nabla f(\mathbf{x}_{(k)}) + \mathbf{A}(\mathbf{x}_{(k)})^{\mathrm{T}}\boldsymbol{\mu}_{(k)} = \mathbf{0} \\ &\mathbf{A}(\mathbf{x}_{(k)})\,\mathbf{v}_{(k)} + \mathbf{c}(\mathbf{x}_{(k)}) = \mathbf{0}. \end{aligned} \tag{3.10}$$

When Equation (3.10) is compared with Equation (3.8), it can be observed that $(\mathbf{v}_{(k)}, \boldsymbol{\mu}_{(k)})$ can be identified from Equation (3.8) only by adding $\mathbf{A}(\mathbf{x}_{(k)})^{\mathrm{T}}\boldsymbol{\lambda}_{(k)}$ to the first line of Equation (3.8) which gives

$$\begin{bmatrix} \nabla f(\mathbf{x}_{(k)}) \\ \mathbf{c}(\mathbf{x}_{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{W}_{(k)} & \mathbf{A}(\mathbf{x}_{(k)})^{\mathrm{T}} \\ \mathbf{A}(\mathbf{x}_{(k)}) & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{(k)} \\ \boldsymbol{\lambda}_{(k+1)} \end{bmatrix} = \mathbf{0}. \tag{3.11}$$

From the assumption of the nonsingularity of the KKT matrix, Equation (3.11) has a unique solution which implies $\boldsymbol{\mu}_{(k)} = \boldsymbol{\lambda}_{(k+1)}$. Thus, an optimum for Equation (3.5) can also be found by sequentially minimizing the QP problem defined in Equation (3.9).

For the general constrained problems, the QP problem is defined as

$$\min_{\mathbf{v}_{(k)}} \left[\frac{1}{2} \mathbf{v}_{(k)}^{\mathrm{T}} \mathbf{W}_{(k)} \mathbf{v}_{(k)} + \nabla f(\mathbf{x}_{(k)})^{\mathrm{T}} \mathbf{v}_{(k)} \right]$$
$$\text{subject to} \begin{cases} \nabla c_i(\mathbf{x}_{(k)})^{\mathrm{T}} \mathbf{v}_{(k)} + c_i(\mathbf{x}_{(k)}) = 0 & i \in \mathcal{E} \\ \nabla c_i(\mathbf{x}_{(k)})^{\mathrm{T}} \mathbf{v}_{(k)} + c_i(\mathbf{x}_{(k)}) \leq 0 & i \in \mathcal{I} \end{cases}$$

where both of the constraint functions are the first order Taylor series approximations of the original constraints at an iterate $\mathbf{x}_{(k)}$.

To summarize, the SQP method is in essence an extension of the Newton's method to the constrained optimization problems. Hence, one would expect that it shares the characteristics of the Newton's method such as rapid convergence when the iterates are close to the solution but an erratic behavior when the iterates are far from a solution [17, 110]. To establish a global convergence for the NLP problems from remote starting points, a way of measuring progress towards a solution is needed. For the SQP method, this is done by employing *merit functions* which quantifies the balance between reducing the objective function value and satisfying the constraints.

The construction of the QP subproblems are the same for all SQP strategies. They differ from each other by the selection of a QP solver and a merit function. In this research, the considered SQP implementation [2] uses the method proposed by Gill et al. [50] for solving the subproblems and utilizes the merit function defined in [2, 65].

SQP methods are proven to converge to an optimum solution for continuous and smooth functions [104]. Its disadvantage is that depending on the selected initial point it might be trapped in a local optimum.

3.3.2 Genetic Algorithms

Genetic Algorithms (GAs) are concerned with the solution of optimization problems in the global sense by imitating the principles of natural evolution. The working principle of the method can be summarized as follows: First, GA is initialized with a random set of points (population or generation). Next, the value of the objective function is calculated for each element of this set. Then GA selects some of these points as *candidates* based on their objective function values and feasibility, and creates a new set of points using them. It does this by keeping some of the candidate points the same (elite), making random changes in some of them (mutation) and combining the vector entries of randomly selected pairs (cross-over). Afterwards it replaces the previous population with the new one (new generation) and follows the same procedure until there is no improvement in the population. The best feasible point (one with minimum objective function value for a minimization problem) of the last population is defined as the optimum solution.

During its process, GA does not require any derivative information of the objective or the constraint functions. A solution provided by it is likely to be close to the global optimum. Unfortunately, unlike the gradient-based methods, there is no optimality check for GA. A solution is declared as optimum when some predefined termination

criteria are met. Hence, the solutions are based on estimations and might not be exact. It is also important to mention that, compared to the gradient-based methods GAs need many function evaluations. This is because of the lack of gradient information during the search of an optimum.

The GA implementation considered in this research treats linear and bound constraints different than the nonlinear ones [1]. Since a region restricted by the bound and the linear constraints defines a convex set[2], it is not difficult to generate points in the feasible domain throughout the optimization process. This is generally not an easy task for the nonlinear constraints, which is the reason why they are treated separately. A brief introduction is as follows.

Bound and Linear Constraints

An optimization problem with the linear and the bound constraints can be written in the form

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \begin{cases} \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \mathbf{C}\mathbf{x} = \mathbf{d} \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{cases} \tag{3.12}$$

where the ith row of $\mathbf{A}$, $\mathbf{b}$, $\mathbf{C}$ and $\mathbf{d}$ corresponds to the coefficients of the ith inequality and equality constraints, respectively. The vectors $\mathbf{l}$ and $\mathbf{u}$ define the lower and the upper bounds on $\mathbf{x}$. Among the many ways to handle these constraints in GA [26], the one used in this research solves Equation (3.12) by generating a feasible population at each of its generations using suitable cross-over and mutation operators.

Since the feasible region $\mathcal{F}$ restricted by the linear and the bound constraints defines a convex set, the cross-over point $\mathbf{x}_{\mathrm{CO}}$ is generated as

$$\mathbf{x}_{\mathrm{CO}} = \beta \mathbf{x}_m + (1 - \beta)\mathbf{x}_p$$

where $\mathbf{x}_m$ and $\mathbf{x}_p$ are a randomly selected pair from the set $\mathcal{S}_{\mathrm{P}}$. This set consists of the candidate points of the previous population which are used for generating the current population. The coefficient β is selected arbitrarily in the range [0,1]. The illustration of a convex set and a cross-over point is shown in Figure 3.2. As seen in the figure, any point that lies on the line, connecting the arbitrarily selected two points, is still in the feasible set.

As mentioned before, a mutation point $\mathbf{x}_{\mathrm{M}}$ is obtained by introducing a random change to a randomly selected point $\mathbf{x}$ in $\mathcal{S}_{\mathrm{P}}$. There are two possibilities for the location of $\mathbf{x}$, either it lies strictly inside $\mathcal{F}$ or it lies on the boundaries as illustrated in Figure 3.2. If it is the first case, any direction selected in $\mathcal{F}$ with a sufficiently small step size α produces a mutation point

$$\mathbf{x}_{\mathrm{M}} = \mathbf{x} + \alpha \mathbf{v}$$

where $\mathbf{v}$ is a feasible direction vector. However, for the second case, some of the constraints are active for $\mathbf{x}$. Consequently, only the directions tangent to them and

[2] A set $\mathcal{S}$ is called convex if for all $\mathbf{x}_m$, $\mathbf{x}_p$ in $\mathcal{S}$ and all β in the interval $[0, 1]$, the point $\beta\mathbf{x}_m + (1 - \beta)\mathbf{x}_p$ is in $\mathcal{S}$.

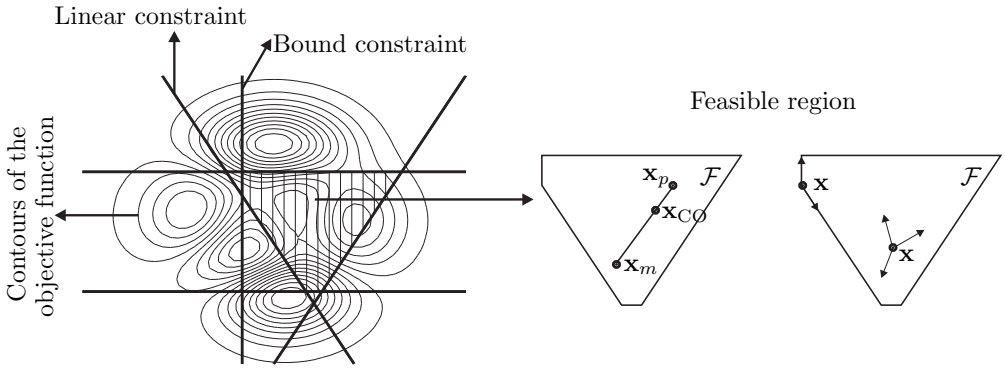


Figure 3.2: (Left) Contours of a function restricted with linear and bound constraints. (Middle) Generating cross-over point. (Right) Generating mutation point.

the ones that lie in the subspace positively spanned[3] by these directions can produce feasible mutation points as shown in Figure 3.2. Details about defining a positive spanning set and selecting a feasible direction vector can be found in [9, 90].

The same step size is used for generating all the mutation points in a population which is updated for a new population based on the success of the previous one. If the best point of the previous population has improved the objective function value, the step size is increased, otherwise it is decreased. Therefore, the search process is adapted to the progress of the generations for the sake of not missing any valuable information about the location of an optimum [1, 9].

Example 3.3.1 For the demonstration of the GA algorithm, the following problem is selected.

$$\min_{\mathbf{x}} \Big[3(1-x_1)^2 \exp\left(-x_1^2-(x_2+1)^2\right) - 10\left(\frac{x_1}{5}-x_1^3-x_2^5\right) \exp\left(-x_1^2-x_2^2\right) - \frac{1}{3}\exp\left(-(x_1+1)^2-x_2^2\right) \Big] \quad \text{subject to} \begin{cases} -2x_1 - x_2 \leq 0.5 \\ 2x_1 - x_2 \leq 3 \\ -0.5 \leq x_1 \leq 2 \\ -1.5 \leq x_2 \leq 1 \end{cases}$$

It has two variables, a nonlinear objective function, 2 linear inequality and 2 bound constraints. In the algorithm, the number of the points in a population and the maximum number of the generations are set to 50. The first 20 points of the previous population are selected as candidates for generating the current population. Among these points 2 of them are used as elite points and the rest is used for generating the cross-over and the mutation points.

[3] A set $\mathbf{V}$ is called a positive spanning set for a space $\mathbb{H}$, if positive linear combinations of its elements are spanning $\mathbb{H}$.

Each population, excluding the elite ones, consists of 60% cross-over and 40% mutation points. As seen in Figure 3.3, first a random feasible population is selected. At each generation, the population is improved so that it evolves towards the optimum point. Finally for the 19th population, the algorithm is terminated due to the similarity of the points with each other. The objective function value is found as -5.7586 for the best value of the last population, that is $\mathbf{x}_{\mathrm{GA}} = [0.495, -1.4909]$. The exact optimum is at $\mathbf{x}_{\mathrm{E}} = [0.5, -1.5]$ with an objective function value of -5.7616. The results clearly reveal that the solution provided by GA is restricted by the performance of the populations calculated at each generation and the total number of the generations. Furthermore, the exact solution is never guaranteed.■

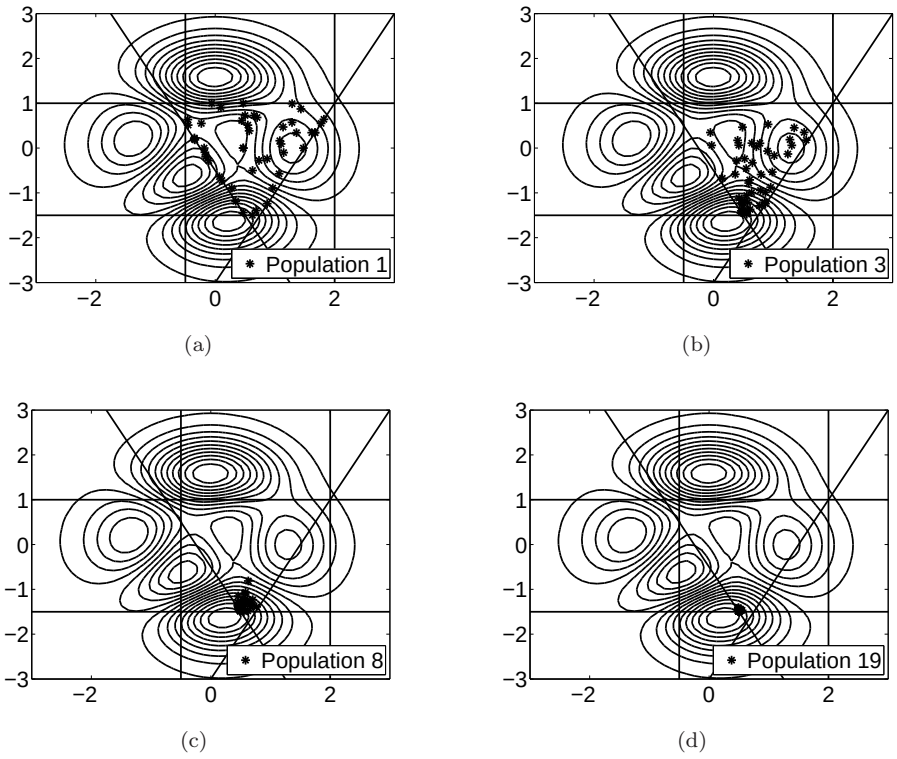


Figure 3.3: Populations 1, 3, 8 and 19

.

Nonlinear Constraints

When Equation (3.12) is extended with nonlinear constraints, the problem becomes

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \begin{cases} \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \mathbf{C}\mathbf{x} = \mathbf{d} \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \\ g_i(\mathbf{x}) \leq 0 \quad i = 1, 2, \ldots, m \\ g_i(\mathbf{x}) = 0 \quad i = m+1, m+2, \ldots, M \end{cases} \tag{3.13}$$

where $\mathbf{g}(\mathbf{x})^{\mathrm{T}} = [g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_M(\mathbf{x})]$ consists of the nonlinear inequality and equality constraints. An important class of methods, used for the optimization of these type of problems, seeks the solution by sequential minimization of subproblems [58, 104]. These subproblems are obtained by combining the objective function with the constraints in an appropriate way.

In the utilized GA implementation [1], the *Composite Lagrangian Barrier-Augmented Lagrangian* (CLB-AL) algorithm of Conn et al. [29, 30] provides a framework for handling the nonlinear constraints. The solver uses the subproblem formulation

$$\min_{\mathbf{x}} \Phi(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}, \mu) \quad \text{subject to} \begin{cases} \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ \mathbf{C}\mathbf{x} = \mathbf{d} \\ \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{cases} . \tag{3.14}$$

where the new objective function is

$$\begin{aligned} \Phi(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}, \mu) &= f(\mathbf{x}) + f_{\mathrm{B}}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s}) + f_{\mathrm{AL}}(\mathbf{x}, \boldsymbol{\lambda}, \mu) \\ &= f(\mathbf{x}) + \left[-\sum_{i=1}^{m} \lambda_i s_i \ln(s_i - g_i(x)) \right] + \left[\sum_{i=m+1}^{M} \lambda_i g_i(\mathbf{x}) + \frac{1}{2\mu} \sum_{i=m+1}^{M} [g_i(\mathbf{x})]^2 \right], \end{aligned} \tag{3.15}$$

the entries of the vector $\boldsymbol{\lambda}^{\mathrm{T}} = \{\lambda_1, \lambda_2 \ldots, \lambda_M\}$ are the Lagrange multiplier estimates, the vector $\mathbf{s}^{\mathrm{T}} = \{s_1, s_2, \ldots, s_m\}$ involves the non-negative shifts and μ is the penalty parameter.

The nonlinear equality constraints are introduced into $f(\mathbf{x})$ using the augmented Lagrangian function $f_{\mathrm{AL}}(\mathbf{x}, \boldsymbol{\lambda}, \mu)$ in which the second term is used to penalize $f(\mathbf{x})$ when the constraints are violated. By driving μ to zero, the constraint violations are penalized with increasing severity. The first term of $f_{\mathrm{AL}}(\mathbf{x}, \boldsymbol{\lambda}, \mu)$ prevents μ from getting very small values thereby, ill-conditioning and numerical problems that may arise during the solution of the subproblem are avoided [104].

The nonlinear inequality constraints are introduced into $f(\mathbf{x})$ by the shifted barrier function $f_{\mathrm{B}}(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{s})$. This function introduces a barrier along the constraint boundaries so that the value of the subproblem approaches infinite as $\mathbf{x}$ converges to the boundaries. The shift term $\mathbf{s}$ relaxes the barrier which is useful if some of $g_i(\mathbf{x})$ are active at a minimum of Equation (3.13).

An optimum is sought by minimizing a sequence of the subproblems defined in Equation (3.14) where the GA approach introduced in Section 3.3.2 is used for the

minimization of the subproblems. First, the initial values $\mathbf{x}_{(0)}$, $\boldsymbol{\lambda}_{(0)}$, $\mathbf{s}_{(0)}$ and $\mu_{(0)}$ are defined. The initial point $\mathbf{x}_{(0)}$ has to be selected in the feasible region. Afterwards, the subproblem is solved by GA. The first population of GA consists of the copies of $\mathbf{x}_{(0)}$ which is distributed into the feasible domain during the optimization of the subproblem. When the subproblem is minimized to a required accuracy and satisfies the feasibility conditions, the *Lagrange* multiplier estimate $\boldsymbol{\lambda}$ is updated utilizing the optimum point while keeping the penalty parameter fixed at its current value. Otherwise the *penalty* parameter μ is reduced by a positive factor while keeping $\boldsymbol{\lambda}$ fixed at its current value. The shift vector $\mathbf{s}$ is modified using $\boldsymbol{\lambda}$ and μ. All these modifications lead to a new subproblem formulation and minimization. Until the termination conditions are met these steps are repeated. Technical details of the algorithm can be found in [1, 30].

Although the algorithm proposed by Conn et al. [30] aims to converge to a point for which the KKT conditions are satisfied, it only formulates the subproblems and adjusts the parameters $\boldsymbol{\lambda}$, $\mathbf{s}$ and μ. The accurate update of the parameters depends on the intermediate solutions provided by the subproblem solver GA. Consequently, the robustness of the CLB-AL framework is strongly related with the progress of GA.

3.4 Multi-modality and Finding Global Optimum

As introduced earlier, if an optimization problem has multiple optimum solutions, it is called multi-modal. Multi-modality causes difficulties for optimization methods in terms of finding the global optimum. In order to increase the chance of getting the global solution, two approaches can be considered; *Multi-Level Hybrid Optimization* (MLHO) and *Multi-Start Local Optimization* (MSLO).

In the MLHO scheme, a sequence of different optimization methods is used. The motivation is to utilize the strong features of the selected methods through different stages of the optimization process. In this research, the global search technique GA and the gradient based SQP method are considered for the hybrid optimization scheme. The possibility of getting in the neighborhood of the global optimum is high with GA, although it is a low-fidelity method. It never guarantees an exact optimum. Thus, GA is used in the global search phase for finding a good starting point for SQP. A high-fidelity method, SQP, is employed in the local search phase for finding an exact solution.

In the MSLO scheme, a local search algorithm is run with different initial points. If the locations of the multiple optima are approximately known, this knowledge can be used to define the initial points. Otherwise these points are selected randomly in the design space. The best calculated optimum is assigned as the global one. This scheme does not guarantee the global optimum; however, the chance increases with an increasing number of the initial points. In this study, the SQP method is also used for finding the local solutions.

3.5 Numerical Results and Discussions

MLHO and MSLO are demonstrated on multi-modal test problems which are taken from [132]. Computational efficiency of the schemes and their progress in finding the global optimum are evaluated based on the obtained results.

In the MSLO scheme, 500 initial points are used which are chosen by the Latin Hypercube Sampling (LHS) (see Section 4.1.1 for details) within the variable range. The number of the starting points is selected arbitrarily. The scheme is executed only once for each example. The results, presented in tables, summarize the computed optima by each SQP run corresponding to one starting point. However, there is only one optimum found as the result of the scheme which is the lowest of the local optima.

MLHO is executed 10 times for each problem to test its robustness on finding the global optimum. Non-robustness may arise due to the fact that random number generators are involved in the scheme as well as the issues associated with the handling of the nonlinear constraints.

MATLAB implementation of SQP method [2] is utilized with its default settings for the local search. Derivatives are computed by the forward finite difference approximation.

The algorithm defined in Section 3.3.2 belongs to the GA solver of MATLAB [1] which is used with few changes in its default settings. The total number of the points in a population is selected as 100 with the motivation of having a detailed search in the feasible domain. This number could have been selected larger with the consequences of more function evaluations. The maximum number of the generations is defined as 30. That means, if GA has not found a solution until this generation, it is terminated with the best result of the last population. The algorithm is also stopped when the best design values obtained in 7 subsequent generations are the same. The rest of the stopping conditions are used in their default settings. It is important to note that there is no predefined optimal settings for GA in the literature. Since in this study GA is used at the global search phase of MLHO for finding a good starting point for SQP, a point provided by it in the region of an optimum solution (hopefully in the region of the global solution) is sufficient. With the defined termination settings of GA, the computational efforts for searching the exact global optimum are prevented.

Example 3.5.1 The first test problem

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \sum_{i=1}^{4}(5x_i - 5x_i^2) + \sum_{i=5}^{12}(-x_i) - 3$$

$$\text{subject to} \begin{cases} 2x_1 + 2x_2 + x_{10} + x_{11} \leq 10 \\ 2x_1 + 2x_3 + x_{10} + x_{12} \leq 10 \\ 2x_2 + 2x_3 + x_{11} + x_{12} \leq 10 \\ -8x_1 + x_{10} \leq 0 \\ -8x_2 + x_{11} \leq 0 \\ -8x_3 + x_{12} \leq 0 \\ -2x_4 - x_5 + x_{10} \leq 0 \\ -2x_6 - x_7 + x_{11} \leq 0 \\ -2x_8 - x_9 + x_{12} \leq 0 \\ 0 \leq x_i \leq 3, \quad i = 1, 2, \ldots, 12 \end{cases} \tag{3.16}$$

has 12 variables, a nonlinear objective function, 9 linear inequality and 12 bound constraints. The global optimum $f(\mathbf{x}^*) = -104.25$ is defined at $\mathbf{x}^* = [2.5, 2.5, 2, 5, 3, 3, 3, 3, 3, 3, 0, 0, 0]$ and the first three constraints are active at this point.

The results of the MSLO scheme are presented in Table 3.1. The SQP solver converged to a large number of optima including the global one. The average number of the function evaluations for one SQP execution is 58. This number should be multiplied by the number of the initial points to calculate the total number of the function evaluations in MSLO which is $29,000$.

Table 3.1: Computed optima for each individual SQP run within the MSLO scheme.

Optimum value $f(\mathbf{x}^*)$	♯ of times obtained
-104.25	213
-89.83	143
-89.75	13
-81.91	32
-81.70	13
-74.25	43
-59.83	25
-59.75	1
-53.48	3
-51.91	11
-51.70	3

The results of MLHO are as shown in Table 3.2. The scheme provided the global optimum each time it is executed. The average number of the function evaluations is 3150.■

Table 3.2: Computed optima for each execution of the MLHO scheme.

Optimum value $f(\mathbf{x}^*)$	♯ of times obtained
-104.25	10

As realized from these results, using a global search scheme requires a high number of function evaluations compared to a local search scheme SQP. When the nonlinear constraints are involved in the problem formulations, the cost of computing the global optimum may get more expensive. This is demonstrated in the following examples.

Example 3.5.2 The second example

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} 4(x_1 - 1.5)(x_3 - 0.5) - (x_2 - 1)^2 - 2(x_4 - 1)^2$$
$$\text{subject to} \begin{cases} -2x_1 - x_2 \leq -0.75 \\ -x_3 x_4 + 1 = 0 \\ 0 \leq x_i \leq 2, \quad i = 1, 2, 3, 4 \end{cases} \tag{3.17}$$

has 4 variables, a nonlinear objective function, a linear inequality and a nonlinear equality constraint. The global optimum is at $\mathbf{x}^* = [0, 2, 2, 0.5]$ with $f(\mathbf{x}^*) = -10.5$. There are two other local optimum solutions. One is at $\mathbf{x}^* = [0, 0.75, 2, 0.5]$ with $f(\mathbf{x}^*) = -9.56$. The other one $f(\mathbf{x}^*) = -3$ can be found by a unique selection of x_3^* and x_4^* as 0.5 and 2, respectively, by setting x_2 as either 0 or 2 and leaving x_1 free to take any value as long as the inequality constraint is satisfied [132].

The results of the MSLO scheme are summarized in Table 3.3. All three optima are converged successfully. The average number of the function evaluations for one SQP execution and for MSLO are 27 and 13,500, respectively.

Table 3.3: Computed optima for each individual SQP run within the MSLO scheme.

Optimum value $f(\mathbf{x}^*)$	♯ of times obtained
-10.5	151
-9.56	173
-3	176

Table 3.4 shows the results of MLHO. The global optimum is obtained each time the scheme is executed. The average number of the function evaluations is 10,760.■

Table 3.4: Computed optima for each execution of the MLHO scheme.

Optimum value $f(\mathbf{x}^*)$	♯ of times obtained
-10.5	10

In the example, MLHO requires a considerably high number of function evaluations compared to one execution of SQP. This is because of the GA solver. As explained in Section 3.3.2, an optimum solution is approximated by solving a sequence of subproblems when there are nonlinear constraints. This means that GA has to be run multiple times in these types of problems unlike the ones with the linear and the bound constraints.

So far the global optimum is obtained in all ten runs of the MLHO scheme. The next example shows that this is not always true.

Example 3.5.3 The third test problem

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \sum_{i=1}^{14} (x_i - 3)^2$$

$$\text{subject to} \begin{cases} (\sin(x_3^{-2}) + x_4^2)x_5^{-2} - 1 \leq 0 \\ (x_8^2 + x_9^2)x_{11}^{-2} - 1 \leq 0 \\ (x_{11}^2 + \sin(x_{12}^{-2}))x_{13}^{-2} - 1 \leq 0 \\ (x_5^2 + x_6^{-2})x_7^{-2} + (x_8^{-2} + x_{10}^2)x_{11}^{-2} + (x_{11}^2 + x_{12}^2)x_{14}^{-2} - 3 \leq 0 \\ (x_3^2 + x_4^{-2} + x_5^2)x_1^2 - 1 = 0 \\ (\sin(x_{11}^2) + x_{12}^2 + x_{13}^2 + x_{14}^2)\sin(x_6^2) - 1 = 0 \\ (x_5^2 + x_6^2 + x_7^2)x_2^2 + (x_8^2 + x_9^{-2} + x_{10}^{-2} + x_{11}^2)x_3^{-2} - 2 = 0 \\ 0.1 \leq x_i \leq 5, \quad i = 1, 2, \ldots, 14 \end{cases} \tag{3.18}$$

has 14 variables, a nonlinear objective function, 4 nonlinear inequality, 3 nonlinear equality and 14 bound constraints. The global optimum is $f(\mathbf{x}^*) = 17.56$ and all the constraints are active at the optimum point $\mathbf{x}^* = [0.23, 0.13, 3.26, 2.78, 2.80, 3.06, 3.21, 2.25, 2.38, 2.73, 3.27, 2.76, 3.29, 3.47]$.

Table 3.5 summarizes the results of MSLO where the initials "NC" stands for *No Convergence*. The SQP solver converged to a large number of optima including the global one. It also could not converge to a solution for 26 initial points. The average number of the function evaluations for each execution of SQP is 230. This number is $115,000$ for MSLO.

MLHO experienced difficulties in identifying the global optimum which is found only 2 times out of 10 executions as presented in Table 3.6. The average number of the function evaluations of the scheme is $38,611$.■

Table 3.5: Computed optima for each individual SQP run within the MSLO scheme.

Optimum value $f(\mathbf{x}^*)$	♯ of times obtained
17.56	54
17.75	73
17.89	48
18.54	36
19.00	78
19.46	43
20.52	32
25.58	1
26.04	1
26.47	1
26.76	1
27.31	1
30.57	103
34.15	1
35.05	1
NC	26

As observed from the test problems, pursuit of the global optimum requires a high number of function evaluations. When the objective and/or the constraint functions are based on long running simulations (e.g. FE simulations), execution of even a local optimization method can be very challenging [121].

Concerning MLHO, the chance of obtaining the global optimum is highly correlated with the progress of GA. As realized from the last example, the larger the number of the nonlinear constraints gets, the lower the fidelity of the scheme is. The CLB-AL framework, employed for the nonlinear constraint problems, is originally meant for the gradient-based methods. In the framework, the convergence check between the subproblems and the termination conditions are all based on the gradient information. Since this is not used in GA, all these conditions are replaced by less strict non-gradient termination conditions. This can cause inaccurate update of the subproblem parameters which may then lead to convergence problems. Additionally, in the framework, the solutions of the subproblems are supposed to converge to the main global optimum in the limit. Finding a local optima instead of the global one for some subproblems during the optimization process may have a negative influence on the correct update of the parameters. This may then hinder the convergence of the solution.

To summarize, the fidelity of the MLHO scheme on providing the global optimum depends on the progress of GA. A promising point near the global optimum may not be guaranteed especially for problems with a high number of nonlinear constraints. When the number of the starting points is sufficient to explore the feasible domain,

Table 3.6: Computed optima for each execution of the MLHO scheme.

Optimum value $f(\mathbf{x}^*)$	♯ of times obtained
17.56	2
17.75	1
17.89	1
19.00	2
30.57	4

the possibility of finding the global optimum is high with MSLO.

3.6 Summary and Conclusions

This study only considers continuous optimization problems. Therefore, in this chapter, first, the general formulation of these problems were given. Next, the necessary and the sufficient conditions for optimality were introduced. Afterwards, the solution methods that can be used to solve the most general class of continuous optimization problems; Non-Linear Programming (NLP); were explained briefly. Attention was paid to the gradient based Sequential Quadratic Programming (SQP) method and the stochastic derivative-free method Genetic Algorithms (GAs). Since NLP problems are generally multi-modal, two optimization schemes were studied to find the global optimum solution. Their performances were demonstrated by numerical test problems.

The SQP method is fast and robust. It is suitable for solving any continuous optimization problem. The computed optimum is exact. Unfortunately, it can be a local one depending on the initial point.

The GA method searches the design space globally. Therefore, it is suitable to solve multi-modal problems. Its convergence rate is low and it never guarantees finding an exact optimum. When the nonlinear constraints exist in the problem formulation, the fidelity of the method may get lower.

The objective of employing the Multi-Start Local Optimization (MSLO) and the Multi-Level Hybrid Optimization (MLHO) schemes is to find the global optimum of the multi-modal NLP problems. Both of these schemes require more function evaluations than the gradient-based local search methods. Thus, they can only be used when the objective and the constraint functions are not computationally expensive to evaluate.

The chance of finding the global optimum gets high in MSLO with an increasing number of the starting points.

The intention of studying MLHO was defining an initial point for the local search method consciously instead of selecting arbitrary points from the domain. In that

way the search of the global optimum would be independent from the number of the initial points which is highly correlated with the multi-modality of the problem. The results reveal that the selected GA algorithm may not always be reliable to provide a point in the neighborhood of the global optimum especially for problems with a large number of nonlinear constraints.

4

Surrogate Modeling

One of the main applications of surrogate modeling is to develop fast mathematical approximations to the long running simulations. In engineering applications, they are generally utilized to replace the Finite Element (FE) models. Once the surrogate models are built, they are many orders of magnitude faster to evaluate than the FE models and can effectively be used in the global optimization schemes.

To preserve the generality of the subject, the term *true model* will be used instead of *finite element model* in the chapter.

The aim of surrogate modeling is to predict the trend of the true model $f(\mathbf{x})$ with n-design variables $\mathbf{x}^{\mathrm{T}} = \{x_1, x_2, \ldots, x_n\}$ by a limited amount of available data

$$\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(Q)}, y^{(Q)})\}$$

generated from $f(\mathbf{x})$. The pairs $\{(\mathbf{x}^{(i)}, y^{(i)}),\ i = 1, 2, \ldots, Q\}$ stand for the values assigned to $\mathbf{x}$ and the corresponding responses of $f(\mathbf{x})$, respectively, i.e. $y^{(i)} = f(\mathbf{x}^{(i)})$.

In this chapter first of all, the fundamental steps of surrogate modeling are introduced briefly. Afterwards, three well-known techniques, namely Polynomial models, Kriging and Artificial Neural Networks, are highlighted. Alternative approaches such as Radial basis functions [22], Support Vector Regression [127], Multivariate Adaptive Regression Splines [67], Genetic Programming [10, 44] etc. are not focused on in this research.

The performance of a selected surrogate model is highly application dependent. Hence, it is not an easy task to define a modeling technique that is superior to the others. This chapter also discusses the advantages and the disadvantages of the considered modeling techniques in order to gain an understanding on their feasibility for different problem formulations.

4.1 The Fundamental Steps of Surrogate Modeling

In this section, the key steps in surrogate modeling are discussed which can be listed as:

1. Design of Experiments (DOE),
2. Numerical simulations,
3. Surrogate model selection,
4. Parameter estimation and model validation.

Furthermore, the pre-processing and the post-processing steps that can be applied before and after obtaining a surrogate model are also discussed.

4.1.1 Design of Experiments

Design of Experiments (DOE) aims to identify the input points (samples) $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(Q)}\}$ that may contain valuable information about the true model $f(\mathbf{x})$ while keeping the number of the samples, Q, as small as possible.

The experimental design strategies can be grouped into *Classical Design of Experiments* (CDOE) and *Design of Computer Experiments* (DOCE) [52, 98].

The first group involves sampling strategies developed for physical experiments. The typical characteristics of these experiments are the existence of a random error due to measurements. Furthermore, the investigated trend of the true model is usually known. Therefore, samples are mainly selected near or on the boundaries of the design space so as to minimize the influence of the random error in subsequent measurements. Additionally, replicated samples are used in order to have a measure of the error.

The second group involves sampling strategies developed for deterministic computer experiments which are not affected by a random error. That is, the repeated experiments for identical samples will give exactly the same results. There is generally limited knowledge on the investigated trend. Consequently, the strategies utilized for the physical experiments do not perform efficiently for this group. As opposed to CDOE, in DOCE the samples are selected such that they fill the design space in order to extract as much information as possible about the true model. This approach is referred to as *space-filling* and in this way, the *bias*[1] *error* is intended to be minimized.

In this research, since the true model is based on the FE simulations, the results of the experiments are deterministic. Thus, the sampling plan of this study falls in the category of DOCE. Among the methods that are available in the literature [52, 98, 106, 122, 140], this study considers one of the most commonly employed ones, the *Latin Hypercube Sampling* (LHS) scheme.

Latin Hypercube Sampling

The objective of LHS is to generate sample points whose projections on the variable axes (i.e. $x_1, x_2, \ldots, x_n$) are evenly distributed.

[1] "Bias quantifies the extent to which the surrogate model outputs differ from the true values calculated as an average over all possible data set" [111].

In order to generate Q samples in a design space of n variables: First, Q^n hypercubes (bins) are built. Next, the samples are randomly placed in the bins by making sure that there is only one sample in all one dimensional projections of the bins.

The following example demonstrates how the samples are generated.

Example 4.1.1 An LHS plan, including 10 samples, of 3 design variables is illustrated in Figure 4.1. In the figure, the location of the samples in a three dimensional space and their projections on the two dimensional space are plotted. There exists 10^3 hypercubes (bins) in total. Each row and the columns of the bins contain only one sample.■

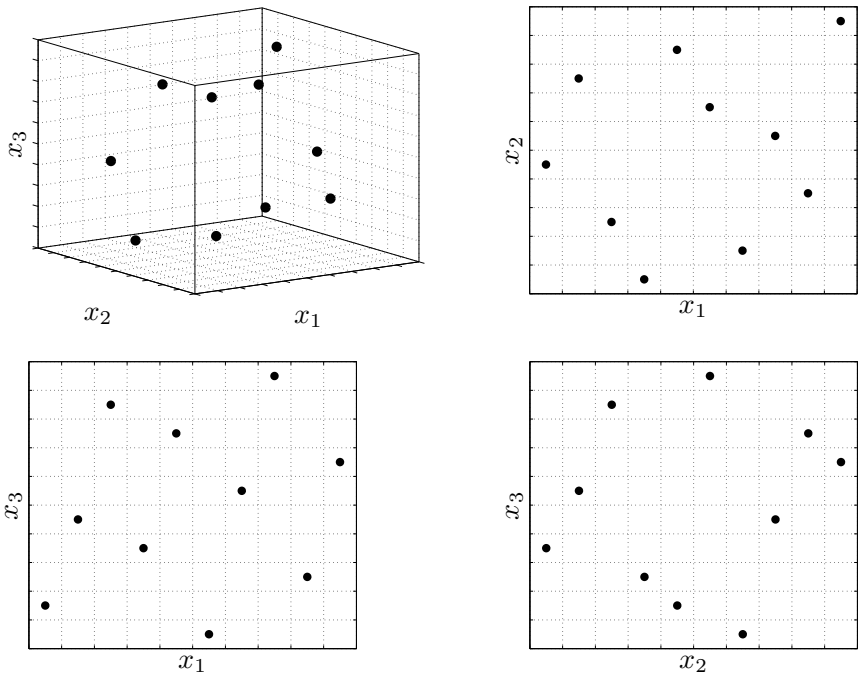


Figure 4.1: Three variable LHS plan with ten samples.

It is important to note that the selection of the samples in the hypercubes is not unique. The shown sampling plan presented in the example is one of many possibilities.

An LHS with its simplest form yields a sampling plan whose projections on the axes are uniform (evenly distributed) whereas it does not guarantee a space-filling design. For instance, projections of the samples selected on the main diagonal of the design space are also evenly distributed, although the space is covered poorly. Therefore, a measure of uniformity is required to distinguish the quality of the LHS plans, which

can then be used to select an optimal sampling plan. One of the widely used measures is the *maximin* metric, that is also the one employed in this research. The uniformity of the sampling is assured by maximizing the minimal distance between the available samples while minimizing the number of the sample pairs separated by the same distance [101]. The details of the algorithm for selecting optimal space-filling samples can be found in [46].

One positive feature of LHS is that the total number of the samples is not restricted with specific multipliers or powers of the number of the design variables like in the *factorial designs* [41].

4.1.2 Numerical Simulations

Having determined the sample points where data will be generated, the next step is to execute the true model for each of these points and to obtain the corresponding outputs $\{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$.

The pairs $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(Q)}, y^{(Q)})\}$ are called the *training data*.

4.1.3 Surrogate Model Selection

After obtaining the training data, a suitable approximation method is required to build a surrogate model. The approximation is built up with these data through the *learning* process. The aim is to define a model whose predictions are reasonably consistent with the data points. Most of the methods are successful in reproducing the training data but this is not a sufficient indicator of the quality of the surrogate. The model should also give acceptable predictions on the unseen data. This is referred to as the *generalization capability* of the model.

4.1.4 Parameter Estimation and Model Validation

The selected surrogate model $\hat{f}(\mathbf{x}, \mathbf{w})$ is based on a vector of parameters, $\mathbf{w}$, which determine its response.

For a given training data $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(Q)}, y^{(Q)})\}$, one approach to find $\mathbf{w}$ is solving the *Least Squares Problem* (LSP):

$$\min_{\mathbf{w}} f_{\mathrm{E}}(\mathbf{w}) = \min_{\mathbf{w}} \left[\sum_{i=1}^{Q} [y^{(i)} - \hat{f}(\mathbf{x}^{(i)}, \mathbf{w})]^2 \right] \tag{4.1}$$

where $f_{\mathrm{E}}(\mathbf{w})$ is the error function. In this approach, the parameter values that provide the most accurate description of the data are sought.

It is an indication of *over-fitting* when a surrogate model exhibits serious oscillations between the data points [16] as shown in Figure 4.2. An over-fitting model has poor generalization capability, therefore it should not be employed as a surrogate model. Two techniques that can be used to prevent over-fitting are *cross-validation* and *regularization*.

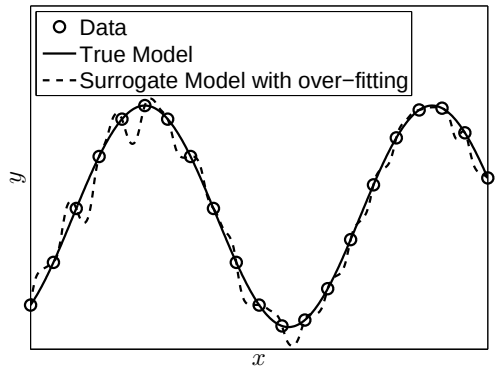


Figure 4.2: An example for over-fitting.

Cross-validation

Cross-validation is carried out, first, by randomly splitting the data into k approximately equal subsets. If a mapping $\varphi : \{1, 2, \ldots, Q\} \to \{1, 2, \ldots, k\}$ is utilized to describe the allocation of the Q data points into one of the k subsets, $\varphi_{(j)}$, $j = 1, 2, \ldots, k$ represent the jth subset. Next, each of these subsets are removed from the data in turn. The parameters of the surrogate model are calculated using the remaining $k - 1$ subsets. The response of the surrogate $\hat{f}^{-\varphi_{(j)}}(\mathbf{x}, \mathbf{w})$ is computed for each point $\mathbf{x}$ in the removed set $\varphi_{(j)}$. When this procedure is completed for all the subsets, the predictions $\{\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(Q)}\}$ of the observed data $\{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$ are obtained. Finally, the cross-validation error is calculated as

$$\epsilon_{\mathrm{cv}} = \frac{1}{Q} \sum_{i=1}^{Q} (y^{(i)} - \hat{y}^{(i)})^2. \tag{4.2}$$

This error gives an indication of the generalization quality of the selected surrogate model. For obtaining ϵ_{cv}, the model parameters have to be calculated for k different data sets. This means that the parameter estimation problem has to be solved k times. Once it is decided that the selected surrogate model is suitable to predict the data, the final model parameters are computed using all the available data points. Generally, k is selected as either 5 or 10 [67].

Regularization

Another technique to control over-fitting is regularization. It is based on the idea of adding a penalty to the error function $f_{\mathrm{E}}(\mathbf{w})$ (see Equation (4.1)),

$$\tilde{f}_{\mathrm{E}}(\mathbf{w}) = f_{\mathrm{E}}(\mathbf{w}) + \lambda f_{\mathrm{R}}(\mathbf{w}) \tag{4.3}$$

so that the unwanted oscillations in the model can be suppressed.

The *regularization coefficient* λ governs the relative importance of the *regularization function* $f_{\mathrm{R}}(\mathbf{w})$ over $f_{\mathrm{E}}(\mathbf{w})$. Large values of λ prevent the surrogate model from becoming finely tuned to the data.

The choice of $f_{\mathrm{R}}(\mathbf{w})$ and λ are crucial for obtaining a surrogate model with a good generalization capability. A summary of various proposed methods for this choice can be found in [129].

4.1.5 Pre-processing and Post-processing Steps

Before estimating the parameters of the selected surrogate model, it is always useful to scale the data, i.e. the pairs $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(Q)}, y^{(Q)})\}$, so that they fall within a specified range (e.g. $[-1\ 1]$, $[0\ 1]$).

After obtaining the parameters, if the model response is desired to be explored in detail, *linear regression analysis* can be performed between the model predictions $\hat{\mathbf{y}} = \{\hat{y}^{(1)}, \hat{y}^{(2)}, \ldots, \hat{y}^{(Q)}\}$ and the observed data $\mathbf{y} = \{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$. A linear curve with a slope 1 and a y-intercept 0 indicates a perfect fit between $\hat{\mathbf{y}}$ and $\mathbf{y}$.

A coefficient r^2 can also be computed which is a measure of the strength of the relationship between the observations and the predictions. It is calculated as

$$r^2 = \left[\frac{\mathrm{Cov}(\hat{\mathbf{y}}, \mathbf{y})}{s_{\hat{\mathbf{y}}} s_{\mathbf{y}}}\right]^2 = \frac{1}{s_{\hat{\mathbf{y}}}^2 {s_{\mathbf{y}}}^2}\left[\frac{1}{Q-1}\sum_{i=1}^{Q}(\hat{y}^{(i)} - \bar{\hat{y}})(y^{(i)} - \bar{y})\right]^2$$

where

$$s_{\hat{\mathbf{y}}} = \sqrt{\frac{1}{Q-1}\sum_{i=1}^{Q}(\hat{y}^{(i)} - \bar{\hat{y}})^2}, \quad s_{\mathbf{y}} = \sqrt{\frac{1}{Q-1}\sum_{i=1}^{Q}(y^{(i)} - \bar{y})^2}$$

are the *sample standard deviation* of $\hat{\mathbf{y}}$ and $\mathbf{y}$, respectively, $\mathrm{Cov}(\cdot, \cdot)$ stands for the covariance and

$$\bar{\hat{y}} = \frac{1}{Q}\sum_{i=1}^{Q}\hat{y}^{(i)}, \quad \bar{y} = \frac{1}{Q}\sum_{i=1}^{Q}y^{(i)}$$

are the mean value of $\hat{\mathbf{y}}$ and $\mathbf{y}$, respectively [24]. The coefficient r^2 varies in the range $[0, 1]$ with 1 representing a perfect fit.

It is important to emphasize that, all these measures give an idea about the correlations between the observations and the predictions. They do not supply any information about the generalization quality of the surrogate model.

4.2 Polynomial Models

A polynomial model is the sum of the weighted basis functions where these functions are selected from a set of all possible combinations of the design variables $\{x_1, x_2, \ldots, x_n\}$. The order of the basis functions is limited to the order of the polynomial model.

4.2.1 Modeling

A polynomial model

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{n_\mathrm{b}} w_j \, \Psi_j(\mathbf{x}) \tag{4.4}$$

fitted to a data set $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(Q)}, y^{(Q)})\}$ can be written as

$$y^{(i)} = \sum_{j=1}^{n_\mathrm{b}} w_j \, \Psi_j(\mathbf{x}^{(i)}) + \epsilon^{(i)} \quad i = 1, 2, \ldots Q \tag{4.5}$$

where n_b is the number of the basis functions and $\Psi_j(\mathbf{x})$ is the jth basis function. The vector ${\mathbf{x}^{(i)}}^\mathrm{T} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}\}$ is the ith sample, the vector $\mathbf{y}^\mathrm{T} = \{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$ defines the observations collected at the input points $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(Q)}\}$, $\mathbf{w}^\mathrm{T} = \{w_1, w_2, \ldots, w_{n_\mathrm{b}}\}$ is a vector of the model parameters (coefficients) and $\boldsymbol{\epsilon}^\mathrm{T} = \{\epsilon^{(1)}, \epsilon^{(2)}, \ldots, \epsilon^{(Q)}\}$ is a vector of the prediction errors. The basis functions $\Psi_j(\mathbf{x})$, $j = 1, 2, \ldots, n_\mathrm{b}$ are picked from a catalogue of all possible combinations of the design variables of an order not larger than the order of a polynomial approximation. For instance, for a third order polynomial with two design variables $\mathbf{x}^\mathrm{T} = \{x_1, x_2\}$, $\Psi_j(\mathbf{x}) \in \{1, x_1, x_2, x_1x_2, x_1^2, x_2^2, x_1^2x_2, \ldots, x_2^3\}$.

Equation (4.5) is represented in matrix form as

$$\mathbf{y} = \boldsymbol{\Psi}\mathbf{w} + \boldsymbol{\epsilon} \tag{4.6}$$

where the matrix $\boldsymbol{\Psi} \in \mathbb{R}^{Q \times n_\mathrm{b}}$

$$\boldsymbol{\Psi} = \begin{bmatrix} \Psi_1(\mathbf{x}^{(1)}) & \Psi_2(\mathbf{x}^{(1)}) & \ldots & \Psi_{n_\mathrm{b}}(\mathbf{x}^{(1)}) \\ \Psi_1(\mathbf{x}^{(2)}) & \Psi_2(\mathbf{x}^{(2)}) & \ldots & \Psi_{n_\mathrm{b}}(\mathbf{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_1(\mathbf{x}^{(Q)}) & \Psi_2(\mathbf{x}^{(Q)}) & \ldots & \Psi_{n_\mathrm{b}}(\mathbf{x}^{(Q)}) \end{bmatrix}$$

is called the *Vandermonde* matrix. The unknown coefficients $\mathbf{w}$ in Equation (4.6) can be determined using the least squares estimation defined in Equation (4.1) as:

$$\min_{\mathbf{w}} \; [(\mathbf{y} - \boldsymbol{\Psi}\mathbf{w})^\mathrm{T}(\mathbf{y} - \boldsymbol{\Psi}\mathbf{w})]. \tag{4.7}$$

Differentiating Equation (4.7) with respect to $\mathbf{w}$ and setting the results to zero gives

$$\mathbf{w} = \boldsymbol{\Psi}^\dagger \mathbf{y} \tag{4.8}$$

where $\boldsymbol{\Psi}^\dagger = (\boldsymbol{\Psi}^\mathrm{T}\boldsymbol{\Psi})^{-1}\boldsymbol{\Psi}^\mathrm{T}$ is the *Moore-Penrose pseudo inverse* of $\boldsymbol{\Psi}$.

Example 4.2.1 In order to gain insight on the logic of the polynomial approximations, a model with one design variable of an order 3 is considered. The set of the selected basis functions is $\{1, x, x^2, x^3\}$.

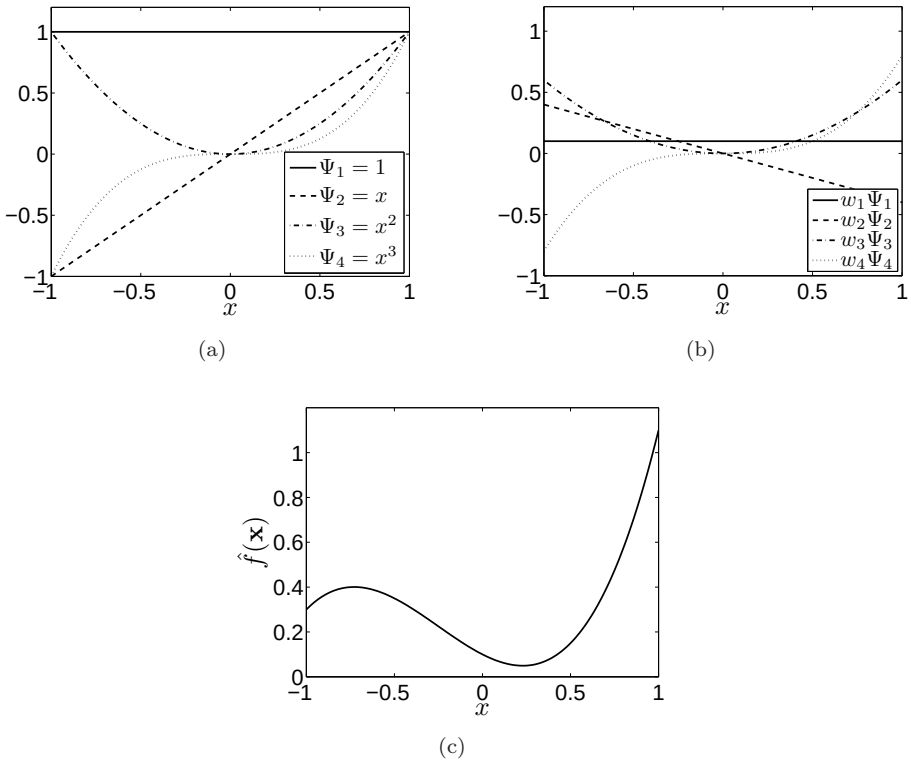


Figure 4.3: A 3rd order polynomial model and its basis functions.

The variation of these functions for $-1 \leq x \leq 1$ is shown in Figure 4.3(a). They are then weighted using the coefficients $\mathbf{w}^{\mathrm{T}} = [0.1, -0.4, 0.6, 0.8]$ as plotted in Figure 4.3(b) and summed up to form the complete polynomial approximation as plotted in Figure 4.3(c).

As seen in the figures, multiplying the basis functions with weight terms only shifts the responses on the y-axes and changes the direction of the functions but the shape of the basis functions remain the same. It is not possible either to move the locations of the curvatures (for higher order terms) or the width of them. For increasing the nonlinearity, higher order basis functions should be considered while making sure that the total number of the data is still larger than the total number of the parameters. If this criterion is violated, infinitely many polynomial models can be fitted to the data which will result in poor generalization capability.■

4.2.2 Results and Discussion

The practical performance of the polynomial models is discussed on two numerical test problems.

Example 4.2.2 In order to demonstrate the generalization capability of the polynomial models, a single variable test function

$$f(x) = x^2 + 5x - 6\sin(8x - 5)$$

is selected. A data set of 20 samples is generated and the corresponding responses of $f(x)$ are evaluated. These responses are then modified with a normally distributed noise. The generated data is as plotted in Figure 4.4.

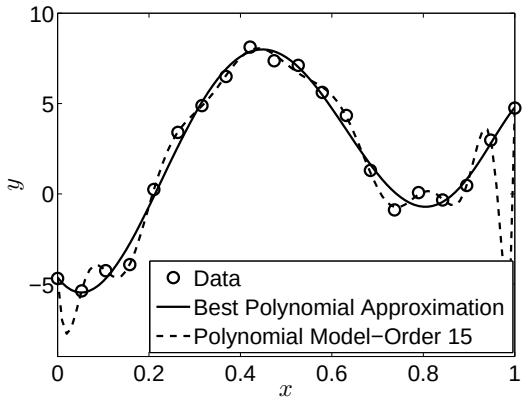


Figure 4.4: Results of the first test problem.

To obtain the best polynomial model that fits the data, different models with an order of up to 15 are taken into account. The data set is divided into $k = 5$ subsets for cross-validation.

It is found that a polynomial model of order 6 fits the data best with a cross-validation error of 2.73. This is presented in Figure 4.4. The corresponding model parameters $\mathbf{w}^{\mathrm{T}} = [7.56, -5.16, -14.16, 6.91, 6.39, -1.46, -0.79]$ are obtained using the complete data set. As seen in the figure, the selected model is successful in predicting the underlying behavior.

Increasing the number of the basis functions to tune the model to the data leads to a model with a poor generalization capability as plotted in Figure 4.4.■

When the number of the design variables is small, finding the best order polynomial is not an expensive task; although the appropriate model function, accordingly the basis functions, have to be chosen in advance. Polynomial models can also be preferred

when the complexity of the data and the interaction between the design variables are approximately known. On the other hand, in most engineering design problems, the available data come from the numerical simulations. These problems are generally multi-dimensional and there is limited knowledge on the trend of the data. When a polynomial approximation is utilized for these types of problems, finding the best polynomial model can be expensive due to the rapidly increasing number of the required data with the number of the design variables [78].

Polynomial models can also be used for the identification of the promising regions in a design space which may find its application in the field of structural design optimization [51]. This feature is demonstrated with the following example.

Example 4.2.3 As a second test problem, a two variable test function

$$f(\mathbf{x}) = \frac{x_1^6}{3} + 4x_1^2 - 2.1x_1^4 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{4.9}$$

is selected which is plotted in Figure 4.5.

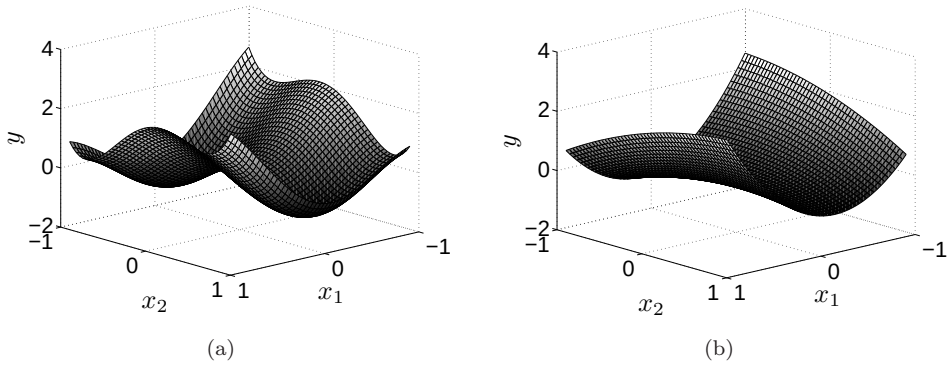


Figure 4.5: Numerical test problem. (a) True model, (b) 2nd order polynomial approximation.

A data set including 20 samples and the corresponding responses of $f(\mathbf{x})$ is generated using Equation (4.9). For the identification of the general trend of $f(\mathbf{x})$, a second order polynomial is used which is plotted in Figure 4.5(b). This order is selected in order to demonstrate how the low order polynomials perform when the approximated underlying behavior is more complex.■

As realized from Figure 4.5(b), the polynomial model provides information about the general trend of the true function, although it is very inflexible to capture the details. This feature can be used in design optimization problems to identify the promising regions of the design space which can then be delved into details for finding an optimal design. On the other hand, it is important to keep in mind that as the underlying

behavior gets more nonlinear, low order polynomials lose their reliability in locating promising regions.

Theoretically, the function defined in Equation (4.9) can be predicted exactly with polynomial approximation. To be able to do that, all the basis functions up to 6th order must be used which makes 28 basis functions, accordingly 28 parameters. In this case, the available data is not sufficient to yield a unique solution for the model parameters. On the other hand, if the basis functions $\{x_1x_2, x_1^2, x_2^2, x_1^4, x_2^4, x_1^6\}$ are known beforehand the exact solution can be found.

To summarize, when a problem gets multidimensional, choosing the order of the best polynomial model as well as deciding on the governing basis functions becomes more difficult. Additionally, the data has to be sufficient to compute a unique solution. Hence, the polynomial approximation turns into a cumbersome task as the number of the design variables increases.

4.2.3 Conclusions

Polynomial models are suitable in cases where the trend of the true model is approximately known or the number of the design variables are small. Low order polynomial models can be preferred to identify the promising regions in a design space, although they can be very inflexible to capture the local behavior of the true model. A positive feature of polynomial models is their transparency; that is, they are capable of illustrating explicit relationships between the input variables and the corresponding responses. However, the basis functions have to be chosen in advance.

The majority of the engineering design problems are multidimensional and limited information is available on the input-response relationship. Building a polynomial approximation for these types of problems may have some drawbacks like the large amount of required data and the unsatisfactory prediction capability. Consequently, polynomial models are not frequently preferred as surrogate models for such problems.

4.3 Kriging

In surrogate modeling, the main idea is to represent a true model by linear combinations of the weighted predefined basis functions. These functions can be very simple, as used in the polynomial approximation, or more flexible as discussed in the following.

Kriging involves a defined *base function* which is a very rough representation of the trend in the data. In the model, this function is combined with the *basis functions* to compute the exact predictions at the available sample points.

In this study, the exponential function

$$\Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{d}) = \prod_{j=1}^{n} \exp(-\theta_j \, |d_j|^{p_j}), \quad 0 < p_j \leq 2 \tag{4.10}$$

is considered among the available Kriging basis (correlation) functions [118]. In Equation (4.10), $\mathbf{d}^{\mathrm{T}} = \{d_1, d_2, \ldots, d_n\}$ is a vector of n input variables and $|\cdot|$ denotes the absolute value. The entries of the vectors $\boldsymbol{\theta}^{\mathrm{T}} = \{\theta_1, \theta_2, \ldots, \theta_n\}$ and $\mathbf{p}^{\mathrm{T}} = \{p_1, p_2, \ldots, p_n\}$ determine the structure of the basis function $\Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{d})$.

In this chapter focus will be on the *Ordinary Kriging*. Detailed information about the other Kriging approaches can be found in [8, 55, 77].

4.3.1 Modeling

An *Ordinary Kriging* model can be defined as

$$\hat{f}(\mathbf{x}) = \mu + Z(\mathbf{x}) \tag{4.11}$$

where μ is a constant term that needs to be determined and it describes the general trend of the data roughly. The stochastic function $Z(\mathbf{x})$ guarantees the interpolation of the observations at the sampling points. Hence, the predictions of the Kriging model at the sample points $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(Q)}\}$ are equal to the available observations $\mathbf{y}^{\mathrm{T}} = \{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$.

The function $Z(\mathbf{x})$ is defined by the normal distribution with mean μ and variance σ^2. For Q observation points $\mathbf{y}^{\mathrm{T}} = \{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$ which are assumed to be correlated with each other, the function that defines the likelihood of these observations is given by

$$L(\mu, \sigma^2, \boldsymbol{\theta}, \mathbf{p}) = \frac{1}{(2\pi\sigma^2)^{\frac{Q}{2}} \det(\boldsymbol{\Psi})^{\frac{1}{2}}} \exp\left[-\frac{(\mathbf{y} - \mathbf{1}\mu)^{\mathrm{T}} \boldsymbol{\Psi}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}\right]. \tag{4.12}$$

In Equation (4.12), $\boldsymbol{\Psi} \in \mathbb{R}^{Q \times Q}$ is the correlation matrix which explains the relationship between the observation points and $\det(\cdot)$ denotes the determinant. In this study, the correlation function defined in Equation (4.10) is used for computing the matrix $\boldsymbol{\Psi}$ which is symmetric with diagonal entries of 1. Each entry ψ_{il} of $\boldsymbol{\Psi}$ is calculated by

$$\begin{aligned} \psi_{il} &= \Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{x}^{(i)} - \mathbf{x}^{(l)}) \\ &= \exp(-\sum_{j=1}^{n} \theta_j \, |x_j^{(i)} - x_j^{(l)}|^{p_j}) \quad i, l = 1, 2, \ldots, Q. \end{aligned} \tag{4.13}$$

The *likelihood function* $L(\mu, \sigma^2, \boldsymbol{\theta}, \mathbf{p})$ is based on the parameters μ, σ^2, $\boldsymbol{\theta}$ and $\mathbf{p}$. Maximizing this function with respect to these parameters determines the likelihood function that is most likely to produce the observations $\mathbf{y}$. To simplify the optimization, generally the logarithm of Equation (4.12), i.e.

$$\begin{aligned} \ln(L(\mu, \sigma^2, \boldsymbol{\theta}, \mathbf{p})) = -\frac{Q}{2}\ln(2\pi) - \frac{Q}{2}\ln(\sigma^2) \\ - \frac{1}{2}\ln\left(\det(\boldsymbol{\Psi})\right) - \frac{(\mathbf{y} - \mathbf{1}\mu)^{\mathrm{T}} \boldsymbol{\Psi}^{-1} (\mathbf{y} - \mathbf{1}\mu)}{2\sigma^2}, \end{aligned} \tag{4.14}$$

is maximized.

The maximum likelihood estimates $\hat{\sigma}^2$ and $\hat{\mu}$

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^{\mathrm{T}} \boldsymbol{\Psi}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{Q}, \quad \hat{\mu} = \frac{\mathbf{1}^{\mathrm{T}} \boldsymbol{\Psi}^{-1} \mathbf{y}}{\mathbf{1}^{\mathrm{T}} \boldsymbol{\Psi}^{-1} \mathbf{1}} \tag{4.15}$$

of the variance σ^2 and the mean μ of the predictions $\{Z(\mathbf{x}^{(1)}), Z(\mathbf{x}^{(2)}), \ldots, Z(\mathbf{x}^{(Q)})\}$ are found by setting the partial derivative of $\ln(L(\mu, \sigma^2, \boldsymbol{\theta}, \mathbf{p}))$ with respect to σ^2 and μ to zero. Equation (4.15) is then inserted back into Equation (4.14) and the ln-likelihood function now is maximized for $\boldsymbol{\theta}$ and $\mathbf{p}$. When the constant terms that do not have any contribution on computing these parameters are removed, $\boldsymbol{\theta}$ and $\mathbf{p}$ are found by maximizing the *concentrated ln-likelihood function* [47]

$$\max_{\boldsymbol{\theta}, \mathbf{p}} L(\boldsymbol{\theta}, \mathbf{p}) = \max_{\boldsymbol{\theta}, \mathbf{p}} \left[-\frac{Q}{2} \ln(\hat{\sigma}^2) - \frac{1}{2} \ln(\det(\boldsymbol{\Psi})) \right]. \tag{4.16}$$

In the literature, the proposed lower and the upper bounds for θ_j are 10^{-3} and 10^2, respectively [46]. The problem defined in Equation (4.16) is a constrained optimization problem which may have multiple local maxima [95]. In this research, the MLHO scheme (see Section 3.4 for details) is utilized for its solution.

Once the parameters $\mathbf{p}$ and $\boldsymbol{\theta}$ are estimated, the prediction of the Kriging model $\hat{y}$ for a new input point $\mathbf{x}_{\mathrm{new}}$ is calculated by maximizing the likelihood of the observations $\mathbf{y}$ and the new prediction $\hat{y}$ [46]. The maximum likelihood estimate of $\hat{y}$ is

$$\hat{y} = \hat{\mu} + \mathbf{r}^{\mathrm{T}} \boldsymbol{\Psi}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}) \tag{4.17}$$

where $\mathbf{r}^{\mathrm{T}} = \{r^{(1)}, r^{(2)}, \ldots, r^{(Q)}\}$ is a vector containing the correlations between $\mathbf{y}$ and $\hat{y}$, that is,

$$\mathbf{r} = \begin{Bmatrix} \Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{x}_{\mathrm{new}} - \mathbf{x}^{(1)}) \\ \Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{x}_{\mathrm{new}} - \mathbf{x}^{(2)}) \\ \vdots \\ \Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{x}_{\mathrm{new}} - \mathbf{x}^{(Q)}) \end{Bmatrix} = \begin{Bmatrix} r^{(1)} \\ r^{(2)} \\ \vdots \\ r^{(Q)} \end{Bmatrix}.$$

As mentioned before, a Kriging model consists of a base function and a linear combination of the weighted basis functions. The base function is the estimated mean $\hat{\mu}$. In the model, each sample point is assigned to a basis function whose center is aligned with the corresponding sample (see Figure 4.8(b)). The parameters $\boldsymbol{\theta}$ and $\mathbf{p}$ of all these functions are the same. In Equation (4.17), the basis functions are contained in $\mathbf{r}$ where each of its entries $r^{(i)}$ defines the values of the basis functions at $\mathbf{x}_{\mathrm{new}}$. If $\Psi_i(\mathbf{x})$ is assumed to be the basis function of the ith sample,

$$\Psi_i(\mathbf{x}_{\mathrm{new}}) = r^{(i)} = \Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{x}_{\mathrm{new}} - x^{(i)}), \quad i = 1, 2, \ldots, Q.$$

The term $\boldsymbol{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})$ can be considered as a weight term which defines the contribution of each basis function to the prediction $\hat{y}$ at $\mathbf{x}_{\mathrm{new}}$ such that

$$\mathbf{w} = \boldsymbol{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu}).$$

Having weighted each basis function, they are added to the mean base term $\hat{\mu}$ to obtain $\hat{y}$. As a result, Equation (4.17) can be rewritten as

$$\hat{y} = \hat{\mu} + \sum_{i=1}^{Q} w_i \, \Psi_i(\mathbf{x}_{\text{new}})$$

where w_i is the ith term of the weight vector $\mathbf{w}^{\mathrm{T}} = \{w_1, w_2, \ldots, w_Q\}$.

The Kriging model is built in such a way that it interpolates the data. For instance, for a selected sample $\mathbf{x}^{(i)}$ in $\mathbf{X}$, $\mathbf{r}^{\mathrm{T}}\boldsymbol{\Psi}^{-1}$ is a unit vector with 1 on its ith entry. Therefore, the Kriging prediction at $\mathbf{x}^{(i)}$ is $\hat{y} = \hat{\mu} + y^{(i)} - \hat{\mu} = y^{(i)}$ which is exactly the observation $y^{(i)}$.

Example 4.3.1 In order to gain insight into the correlation function $\Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{d})$ and to see the effects of the parameters $\mathbf{p}$ and $\boldsymbol{\theta}$ on $\Psi(\boldsymbol{\theta}, \mathbf{p}, \mathbf{d})$, a Kriging model with one design variable is considered.

The variation of $\Psi(1, p, \mathbf{x}^{(i)} - \mathbf{x}^{(l)})$ for different values of p is plotted in Figure 4.6.

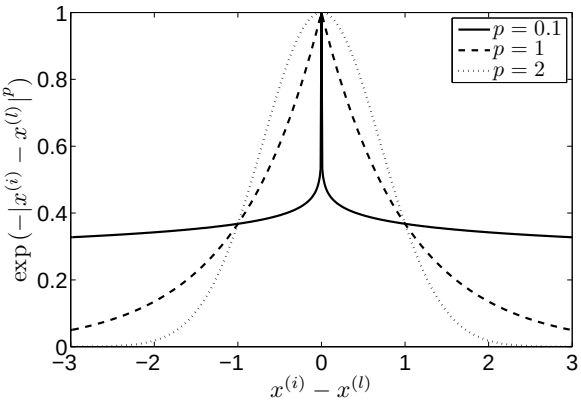


Figure 4.6: Correlations with varying p.

As observed in the figure, when the values of the design variable move close to each other, i.e. $(x^{(i)} - x^{(l)}) \to 0$, the expected correlations of the observations increase, i.e. $\Psi(1, p, \mathbf{x}^{(i)} - \mathbf{x}^{(l)}) \to 1$. When they move apart, i.e. $(x^{(i)} - x^{(l)}) \to \infty$, the correlations get weaker, $\Psi(1, p, \mathbf{x}^{(i)} - \mathbf{x}^{(l)}) \to 0$.

The exponent term p has an effect on the smoothness of the correlation function. Figure 4.6 shows that, for $p = 2$, the correlation function is smooth and differentiable whereas for smaller values of p (i.e $p < 2$), a sudden drop in the correlations with varying separation and non-differentiability are observed.

The variation of $\Psi(\theta, 2, \mathbf{x}^{(i)} - \mathbf{x}^{(l)})$ for different values of θ is plotted in Figure 4.7.

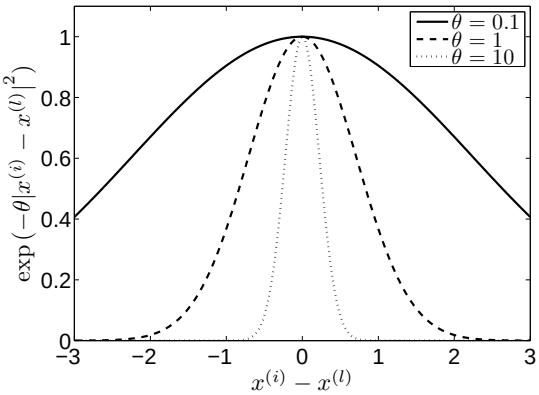
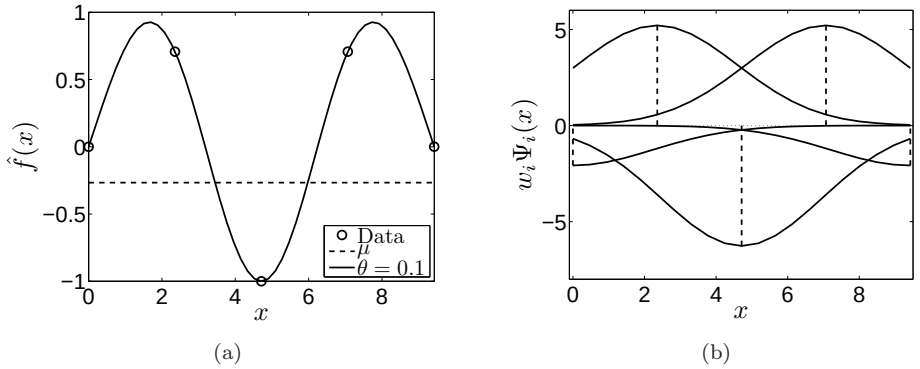


Figure 4.7: Correlations with varying θ.

It is clear from the figure that θ has an effect on the width of the correlation function and determines how far a sample point's influence reaches. For small values of θ, the correlations increase and the variance decreases. This means that a variation in the corresponding design variable does not have a significant effect on the response of the true function [46]. On the other hand, high values of θ indicate that the response of the true function is sensitive to the changes in the design variable that θ corresponds to.

A Kriging model with $\theta = 0.1$ and $p = 2$ is illustrated in Figure 4.8(a).



(a) (b)

Figure 4.8: A Kriging model and its basis funtions.

The dashed line in Figure 4.8(a) represents the maximum likelihood estimate $\hat{\mu}$ of the mean. To interpolate the data, a basis function is assigned to each sample point whose center is aligned with this point as shown in Figure 4.8(b). When the basis functions are weighted and then added to the base term $\hat{\mu}$, the Kriging model is obtained as plotted in Figure 4.8(a).■

4.3.2 Improving Generalization

As mentioned before, an important expectation from a surrogate model is a satisfactory generalization capability. When the observed data has a numerical noise, finding an interpolating model may not always be a good option [18]. While the model is tuned to these data, it may become highly nonlinear.

A Kriging model can be modified by adding a *regression constant* λ to the leading coefficient of the correlation matrix $\mathbf{\Psi}$, as $\mathbf{\Psi} + \lambda\mathbf{I}$, so that control on the interpolation feature of the model is obtained [69]. It is important to emphasize that interpolation of the data is assured by the unit diagonal entries of $\mathbf{\Psi}$ as exemplified in Section 4.3.1.

The regression constant λ is also added to the list of the parameters and they are all determined by maximizing the likelihood function. The parameter estimation problem is now defined as:

$$\max_{\boldsymbol{\theta},\mathbf{p},\lambda} \ln\left(L(\boldsymbol{\theta},\mathbf{p},\lambda)\right) = \max_{\boldsymbol{\theta},\mathbf{p},\lambda} \left[-\frac{Q}{2}\ln\left(\hat{\sigma}^2\right) - \frac{1}{2}\ln(\det(\mathbf{\Psi} + \lambda\mathbf{I})) \right] \tag{4.18}$$

where

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^{\mathrm{T}}(\mathbf{\Psi} + \lambda\mathbf{I})^{-1}(\mathbf{y} - \mathbf{1}\hat{\mu})}{Q}, \quad \hat{\mu} = \frac{\mathbf{1}^{\mathrm{T}}(\mathbf{\Psi} + \lambda\mathbf{I})^{-1}\mathbf{y}}{\mathbf{1}^{\mathrm{T}}(\mathbf{\Psi} + \lambda\mathbf{I})^{-1}\mathbf{1}}. \tag{4.19}$$

In the literature [46], the lower and the upper bounds for λ are proposed as 10^{-6} and 1, respectively.

The regression constant, found at the end of the parameter estimation step, gives an idea about the amount of smoothing used in the model. This parameter can also reduce the risk of ill-conditioning in $\mathbf{\Psi}$.

4.3.3 Results and Discussion

To discuss the practical performance of the Kriging approximation, the same numerical test problems and the same training data sets used in Examples 4.2.2 and 4.2.3 are considered.

In the Kriging model, the Gaussian basis function is utilized for which $p_j = 2$.

Example 4.3.2 To obtain the best model that fits the data, two approaches are tested: The Ordinary Kriging (Kriging interpolation) and the Ordinary Kriging with a regression constant λ (Kriging regression). The results are shown in Figure 4.9.

As observed from the figure, Kriging interpolation fits through all the data points and exhibits a highly nonlinear behavior. On the other hand, when the regression constant is introduced into the model, a good estimate of the true model is obtained. The regression constant for this example is found as $\lambda = 0.02$. Since the lower bound for this term is defined as 10^{-6}, the current value assures that the model does not interpolate the data.■

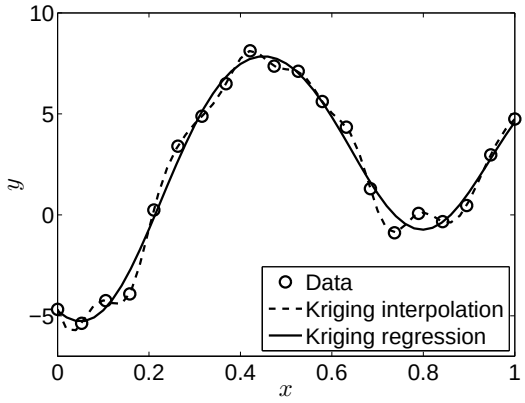


Figure 4.9: An interpolating Kriging model and a Kriging model with a regression constant.

For many engineering design problems, it is not an easy task to determine whether the available data have any fluctuations or not. Therefore, it is always safe to employ the Kriging model with a regression constant. This will increase the number of the parameters but makes Kriging more robust to be used with any kind of data.

Example 4.3.3 In this example the Ordinary Kriging model is used with a regression constant and the results are shown in Figure 4.10.

As seen in the figures, a very close representation of the true function is obtained with a rather limited amount of data.

The regression constant gets a value of $\lambda = 10^{-6}$ which is on the lower bound of the defined interval. This means that the predictions of the Kriging model are close to the exact values at the sample points.

The model parameters are found as $\theta_1 = 0.4250$ and $\theta_2 = 1.4715$ which can be interpreted as the variation of the model with respect to the second design variable is more than its variation with respect to the first design variable. To validate this, the cross sections of the Kriging model are plotted in Figure 4.11. Comparing Figure 4.11(a) and 4.11(b), it is seen that the model is more nonlinear in the x_2-direction than the x_1-direction.■

It is important to emphasize again that Equation (4.18) is a nonlinear function of the model parameters and it may have many local optima. As discussed in Chapter 3, the global optimum is not guaranteed even when the global search schemes are utilized. In this case, a Kriging model may get different model parameters each time it is trained on the same data set. In other words, its representation is not unique.

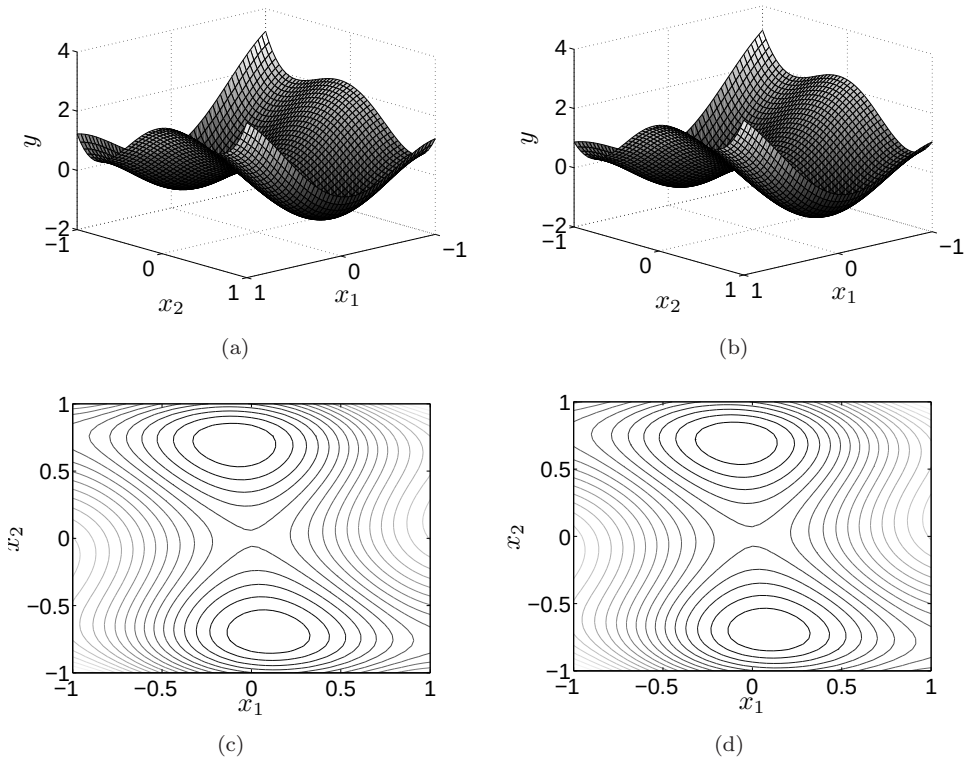
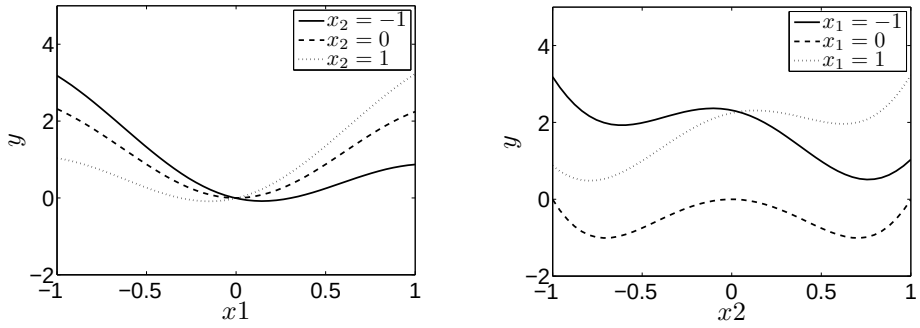


Figure 4.10: Numerical test problem. (a)-(c) True model and its contour plot, (b)-(d) Kriging model with regression constant and its contour plot.

4.3.4 Conclusions

Kriging models are suitable to approximate any nonlinear differentiable function thanks to the employed Gaussian basis functions. They do not need a predefined appropriate model function as opposed to the polynomial models. The data is interpolated and when a regression constant is introduced, the unnecessary oscillations in the data are filtered out.

The procedure for the parameter estimation (training) is not as simple as in the polynomial approximation. The concentrated ln-likelihood function is nonlinear and for each of its evaluation, the explicit estimates of the mean and the variance, as well as the determinant of the correlation matrix have to be calculated. With an increasing number of the design variables or an extending data set, this estimation process may become computationally expensive.



(a) (b)

Figure 4.11: Kriging model: Cross sections with respect to the design variables.

4.4 Artificial Neural Networks

Prediction, classification and data processing are the categories where most of the Neural Network (NN) applications fall into. Each category requires its own NN structure. Neurons, connections and basis functions are the permanent concepts in different NN architectures where the learning rules and the network topology are the main discrepancies between these architectures [6, 38, 64]. A brief discussion about these concepts is given in Section 4.4.1

In this research, NNs are employed for surrogate modeling which falls in the category of *prediction*. Two layer Feedforward Backpropagation NNs (FBNNs) are utilized for this purpose which will be the subject of the following sections. As in the case of Kriging, the employed basis function is also exponential and parametric. Before clarifying all these technical details, first the building blocks of NNs are introduced.

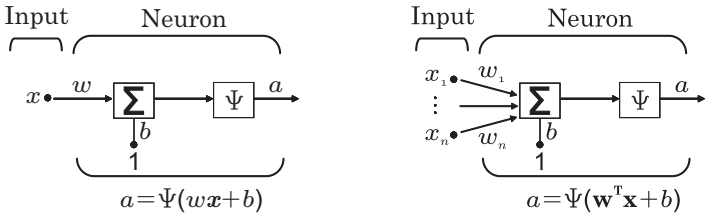
4.4.1 Building Blocks of Neural Networks

A Neuron Model

It is convenient to start explaining NNs with a single variable neuron model. Figure 4.12(a) shows the architecture of this type.

A neuron model consists of a weight w, a bias b and an output a terms and, a basis function $\Psi(d)$. The input for $\Psi(d)$ is $d = wx + b$ and its output $a = \Psi(d)$ is also the output of the neuron.

Generally, a neuron has a vector of design variables as shown in Figure 4.12(b). Each term in the vector $\mathbf{x}^{\mathrm{T}} = \{x_1, x_2, \ldots, x_n\}$ is weighted by the terms of the vector $\mathbf{w}^{\mathrm{T}} = \{w_1, w_2, \ldots, w_n\}$ and summed up to yield a scalar value. A bias term, b, is added to this value and sent to the basis function $\Psi(d)$. Finally the neuron output is



(a) Single Input Neuron (b) Multiple Input Neuron

Figure 4.12: Single input neuron and Multiple input neuron.

obtained as

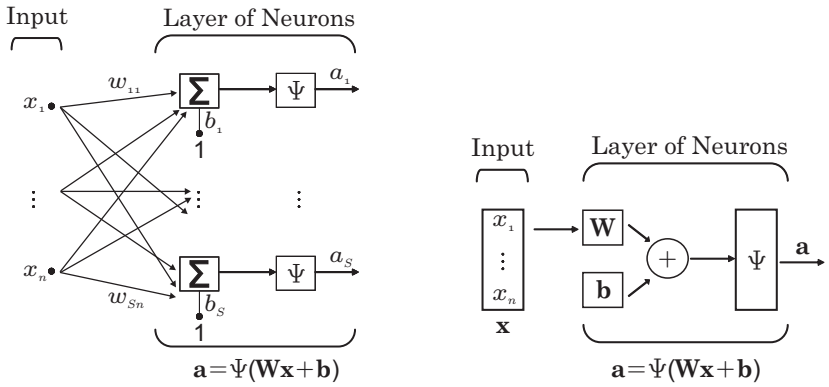
$$a = \Psi(d), \quad d = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b.$$

The terms w_i, $i = 1, 2, \ldots, n$ and b are the model parameters.

A Layer of Neurons

A single neuron is usually not sufficient to approximate the true function. Therefore, in network architectures multiple neurons are employed in parallel, which is called a *layer*.

A single layer of S neurons with n design variables $\mathbf{x}^{\mathrm{T}} = \{x_1, x_2, \ldots, x_n\}$ is illustrated in Figure 4.13(a). The corresponding abbreviated notation is as shown in Figure 4.13(b).



(a) A layer of S neurons. (b) Abbreviated notation for a layer.

Figure 4.13: One layer network.

A layer includes a weight matrix $\mathbf{W} \in \mathbb{R}^{S \times n}$, a bias vector $\mathbf{b}^{\mathrm{T}} = \{b_1, b_2, \ldots, b_S\}$, basis functions and an output vector $\mathbf{a}^{\mathrm{T}} = \{a_1, a_2, \ldots, a_S\}$. The entries of $\mathbf{W}$ and $\mathbf{b}$ are the model parameters. Each design variable x_i is connected to the jth neuron with

the weight term w_{ji}, which represents the contribution of the ith design variable to the jth neuron. In a layer, the number of the design variables is independent from the number of the model parameters as opposed to the Kriging model. Additionally, the basis functions connected to the neurons do not have to be the same in a layer.

The output of the jth neuron is:

$$a_j = \Psi_j(d_j), \quad d_j = \sum_{i=1}^{n} w_{ji} x_i + b_j \tag{4.20}$$

where $\Psi_j(d_j)$ is the corresponding basis function.

Basis Functions

The basis functions used in NNs are chosen according to the types of the problems to be solved. There are different sorts of basis functions, however, only two of them are utilized in this study. These are the *tangent sigmoid* and the *linear* basis functions.

The tangent sigmoid basis function $\Psi^{(1)}(d)$ can take any arbitrary input value $d \in \mathbb{R}$ and suppresses the output into the range $(-1, 1)$ by

$$\Psi^{(1)}(d) = \frac{2}{1 + \exp(-2d)} - 1.$$

The output of the linear basis function $\Psi^{(2)}(d)$ is equal to its input, that is,

$$\Psi^{(2)}(d) = d.$$

These basis functions are preferred because of their differentiability which will become clear when the parameter estimation step is explained.

4.4.2 Modeling

It is stated in [71] that a two layer *Feed Forward* NN (FFNN) model, including the tangent sigmoid basis functions and a sufficient number of the neurons in the first layer and the linear basis function in the second layer, can be trained to approximate any nonlinear function.

A two layer FFNN architecture with a single output is illustrated in Figure 4.14. This architecture is called *feed forward* because defined paths proceed forward and there are no feedback loops in or between the layers.

The layer whose output is also the output of the network is called the *Output Layer* (OL). The other one whose output is the input of the next layer is called the *Hidden Layer* (HL).

The formulation of this type of NN is given as:

$$\hat{f}(\mathbf{x}) = \Psi^{(2)}(d^{\mathrm{OL}}) = d^{\mathrm{OL}}, \quad d^{\mathrm{OL}} = \mathbf{a}^{\mathrm{T}} \mathbf{w}^{\mathrm{OL}} + b^{\mathrm{OL}} \tag{4.21}$$

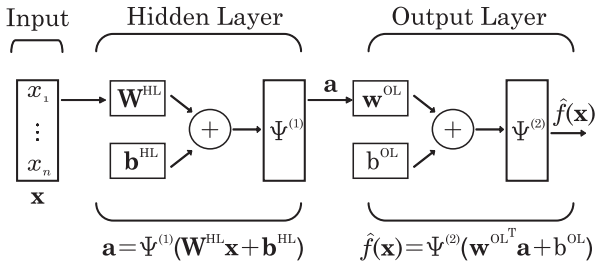


Figure 4.14: A two layer NN architecture with a single network output.

where the vector $\mathbf{a}^{\mathrm{T}} = \{a_1, a_2, \ldots, a_S\}$

$$a_j = \Psi_j^{(1)}(d_j^{\mathrm{HL}}), \quad d_j^{\mathrm{HL}} = \sum_{i=1}^{n} w_{ji}^{\mathrm{HL}} x_i + b_j^{\mathrm{HL}} \quad j = 1, 2, \ldots, S$$

is the output of the hidden layer, $\Psi^{(2)}(d)$ is the linear basis function and $\Psi_j^{(1)}(d)$ is the tangent sigmoid basis function of the jth hidden layer neuron. In Equation (4.21), b^{OL} is the bias term of the output layer, $\{\mathbf{w}^{\mathrm{OL}}\}^{\mathrm{T}} = \{w_1^{\mathrm{OL}}, w_S^{\mathrm{OL}}, \ldots, w_S^{\mathrm{OL}}\}$ is a vector with the output layer weights and $\{\mathbf{b}^{\mathrm{HL}}\}^{\mathrm{T}} = \{b_1^{\mathrm{HL}}, b_2^{\mathrm{HL}}, \ldots, b_S^{\mathrm{HL}}\}$ is a vector with the hidden layer biases. In essence, Equation (4.21) is the linear combination of the weighted tangent sigmoid basis functions.

The basis functions $\Psi_j^{(1)}(d)$ allow the NN model to learn the linear or the nonlinear relationship between the sample points $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(Q)}\}$ and the observations $\mathbf{y}^{\mathrm{T}} = \{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$. Since the output values of the sigmoid basis functions are defined in the interval of $(-1, 1)$, the output layer with a linear basis function $\Psi^{(2)}(d)$ allows the NN model to produce values outside this interval.

The unknown model parameters in Equation (4.21) can be estimated using the least squares estimation

$$\min_{\mathbf{z}} f_{\mathrm{E}}(\mathbf{z}) = \min_{\mathbf{z}} \sum_{k=1}^{Q} \left[y^{(k)} - \hat{f}(\mathbf{x}^{(k)}, \mathbf{z}) \right]^2 = \min_{\mathbf{z}} \sum_{k=1}^{Q} \left[\epsilon^{(k)}(\mathbf{z}) \right]^2. \tag{4.22}$$

In Equation (4.22),

$$\mathbf{z}^{\mathrm{T}} = \{w_{ji}^{\mathrm{HL}}, b_j^{\mathrm{HL}}, w_j^{\mathrm{OL}}, b^{\mathrm{OL}} \mid i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, S\} = \{z_1, z_2, \ldots, z_H\}$$

is a vector including all the NN model parameters and $\boldsymbol{\epsilon}^{\mathrm{T}} = \{\epsilon^{(1)}(\mathbf{z}), \epsilon^{(2)}(\mathbf{z}), \ldots, \epsilon^{(Q)}(\mathbf{z})\}$ is a vector of the errors defined as the difference between the observations and the NN predictions.

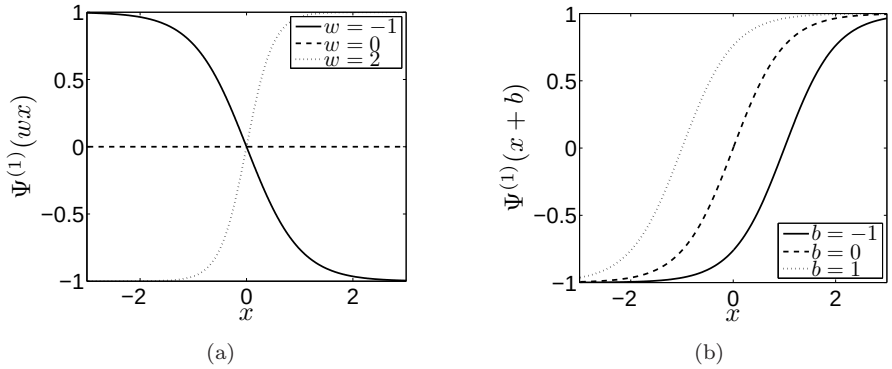
The problem defined in Equation (4.22) is an *unconstrained* optimization problem and can be solved using any suitable gradient-based iterative algorithm or a global search technique [66]. In this research, it is solved using the *Levenberg-Marquardt* (LM) iterative method (see Appendix B). The initial values for the model parameters

are selected using the Nguyen-Widrow initialization rule [103]. This rule generates the initial weight and the initial bias values of the hidden layer parameters so that the active regions (linear parts) of the tangent sigmoid basis functions are distributed evenly over the input space (see Figure 4.15(a) for the illustration of $\Psi_j^{(1)}(d)$). The output layer parameters are initialized with small random values over the interval $[-0.5, 0.5]$ [125].

As most of the traditional numerical algorithms, the LM method also requires the gradient to be known. Hence, for the solution of Equation (4.22), the partial derivatives of $\hat{f}(\mathbf{x}, \mathbf{z})$ with respect to the network parameters $\mathbf{z}$ are needed. These are calculated using the *chain rule* of calculus (see Appendix C). The continuity and the differentiability of $\hat{f}(\mathbf{x}, \mathbf{z})$ with respect to $\mathbf{z}$ are the requirements for the computation of the partial derivatives which is guaranteed by the utilized basis functions $\Psi^{(1)}(d)$ and $\Psi^{(2)}(d)$.

Example 4.4.1 In order to gain insight into the basis functions $\Psi^{(1)}(d)$, $\Psi^{(2)}(d)$ and to understand the effects of the hidden and the output layer parameters on the response of the model, an NN with one design variable is considered.

The variation of the tangent sigmoid basis function $\Psi^{(1)}(d)$ for different values of w is demonstrated in Figure 4.15(a) where $-3 \leq x \leq 3$. As observed from the illustrations, $\Psi^{(1)}(d)$ is linear in a certain interval and it saturates to -1 or 1 as x becomes large in magnitude. The weight term w has an effect on the interval size of the linear part of $\Psi^{(1)}(d)$. The larger the absolute value of this term gets, the smaller the interval size yields. The location of this interval is controlled by the bias term b as shown in Figure 4.15(b). The sign of w determines the direction of $\Psi^{(1)}(d)$.



(a) (b)

Figure 4.15: Effect of the changes in the hidden layer model parameters w and b on the basis function $\Psi^{(1)}(d)$.

To demonstrate the working principles of the NN approximation, an NN model with one design variable including 4 neurons in its

hidden layer is considered. The model parameters are selected as: $b^{\mathrm{OL}} = 10$, $\{\mathbf{b}^{\mathrm{HL}}\}^{\mathrm{T}} = \{0.2, -0.5, 0.8, 0.6\}$, $\{\mathbf{w}^{\mathrm{HL}}\}^{\mathrm{T}} = \{-0.5, 1.2, 0.4, -0.38\}$, $\{\mathbf{w}^{\mathrm{OL}}\}^{\mathrm{T}} = \{22, 36, -3, 15\}$. The hidden layer outputs without any bias terms are plotted in Figure 4.16(a). As seen in the figure, the smaller the absolute value of the weight term gets, the wider the active region of the sigmoid function becomes. It is important to note that the center of these regions coincide. In Figure 4.16(b), the hidden layer outputs including the bias terms are shown where the center of the linear regions is shifted due to the bias terms. The dominance of the hidden layer outputs are defined by the output layer weights as illustrated in Figure 4.16(c). The term b^{OL} shifts the response of the NN model on the y-axis as demonstrated in Figure 4.16(d).■

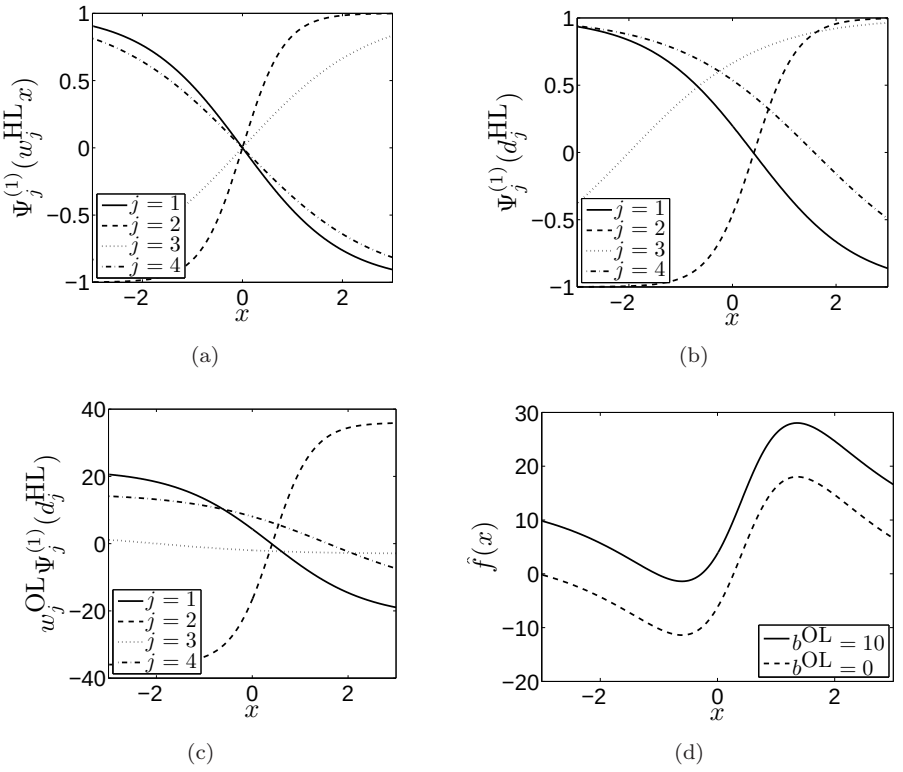


Figure 4.16: Demonstration of a 4 layer, single input NN model.

4.4.3 Improving Generalization

Increasing the number of the hidden layer neurons in an NN model leads to a highly nonlinear model with a poor generalization capability.

As in the case of Kriging, it is possible to prevent unwanted oscillations without predefining an appropriate model function. This is achieved by employing *regularization* for which the model error $f_{\mathrm{E}}(\mathbf{z})$ defined in Equation (4.22) is extended with a regularization term $f_{\mathrm{P}}(\mathbf{z})$ so that the new least squares problem becomes

$$\min_{\mathbf{z}} \tilde{f}_{\mathrm{E}}(\mathbf{z}) = \min_{\mathbf{z}} \left[\beta f_{\mathrm{E}}(\mathbf{z}) + \alpha f_{\mathrm{P}}(\mathbf{z})\right]. \tag{4.23}$$

One possible choice for $f_{\mathrm{P}}(\mathbf{z})$ stems from the observation that an over-fitted function with regions of large curvature has large model parameters. If these parameters are penalized then it is possible to obtain a smooth network response [16]. Therefore, $f_{\mathrm{P}}(\mathbf{z})$ can be selected as the mean of the sum of squares of the network weights and the biases.

The coefficients β and α are the *objective function parameters* and their relative size determines the training process. When $\alpha \ll \beta$, the model error is minimized, when $\alpha \gg \beta$, the NN model is smoothed.

In this research, the algorithm defined in [45] is used for the solution of Equation (4.23). In [45], the Bayesian framework developed by MacKay [93] is utilized to find the optimal settings for the objective function parameters. The algorithm is based on the LM iterative method where the optimum values for α and β are also calculated iteratively. The nice property of MacKay's formulation is that it provides the number of the parameters actively used in the NN model after training. Although the number of the model parameters may highly exceed the number of the data points, the upper bound for the active number of the parameters is always restricted with the total number of the data points.

4.4.4 Results and Discussion

To discuss the practical performance of the NN models, the same numerical test problems and the same training data sets employed in the previous sections are considered. In both of the examples, an FFNN model with 20 neurons in the hidden layer is used.

Example 4.4.2 To obtain the best model that fits the data shown in Figure 4.17, two approaches are tested:

First, the NN model is trained using the LM method. Since there are 20 neurons in the hidden layer, the total number of the model parameters is 61, which is much higher than the total number of the data points. Hence, there are infinitely many NN models that fit the data with very poor generalization capability. One of these models is illustrated in Figure 4.17 with a dashed line.

Second, the NN model is trained using the LM method with Bayesian regularization. The obtained NN model is plotted in Figure 4.17. Thanks to the utilized regularization during the training process, the generalization capability of the NN model is significantly improved. The number of the active

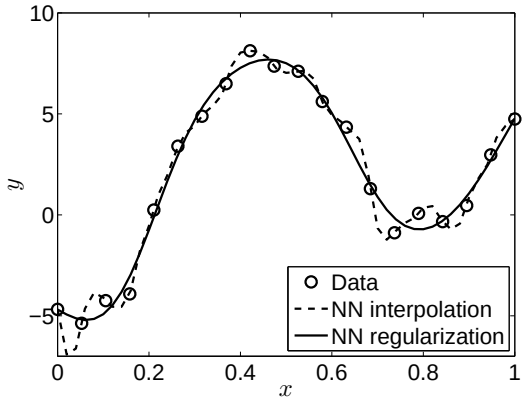


Figure 4.17: An interpolating NN model and a NN model with regularization.

model parameters is found as 9 which makes approximately 3 neurons in the hidden layer. This means that for this example an NN with 3 hidden layer neurons is sufficient to obtain a good approximation. ■

In the NN models, the nonlinearity of the model is controlled by the number of the neurons in the hidden layer. Unfortunately, it is not possible to know the effective number of the neurons beforehand. Finding this number by trial and error can be very time consuming. On the other hand, employing the Bayesian regularization method during the training process improves the generalization capability of NN. Furthermore, the number of the model parameters is allowed to be higher than the number of the data points. This feature makes the NN approximation very suitable to be used on any data without concern about the number of the parameters.

Example 4.4.3 The NN approximation of the function shown in Figures 4.18(a) and 4.18(c) is presented in Figures 4.18(b) and 4.18(d).

The number of the active model parameters is found to be 18 whereas the total number of the parameters is 81. It can be observed from the results that the approximation is satisfactory. ■

The nonlinear basis functions in the hidden layer introduce many local minima into Equation (4.23). The Levenberg-Marquardt method with Bayesian regularization, utilized for minimizing this function is gradient based. Therefore, depending on the initial point, the network solution can be trapped in one of the local minima. Accordingly, the NN model may get different parameter values each time it is trained on the same data set. Hence, its representation is not unique. Employing a global search technique may have an impact on the quality of the solution although it increases the training time significantly. It is important to mention that even the global search techniques do not guarantee the global parameter values.

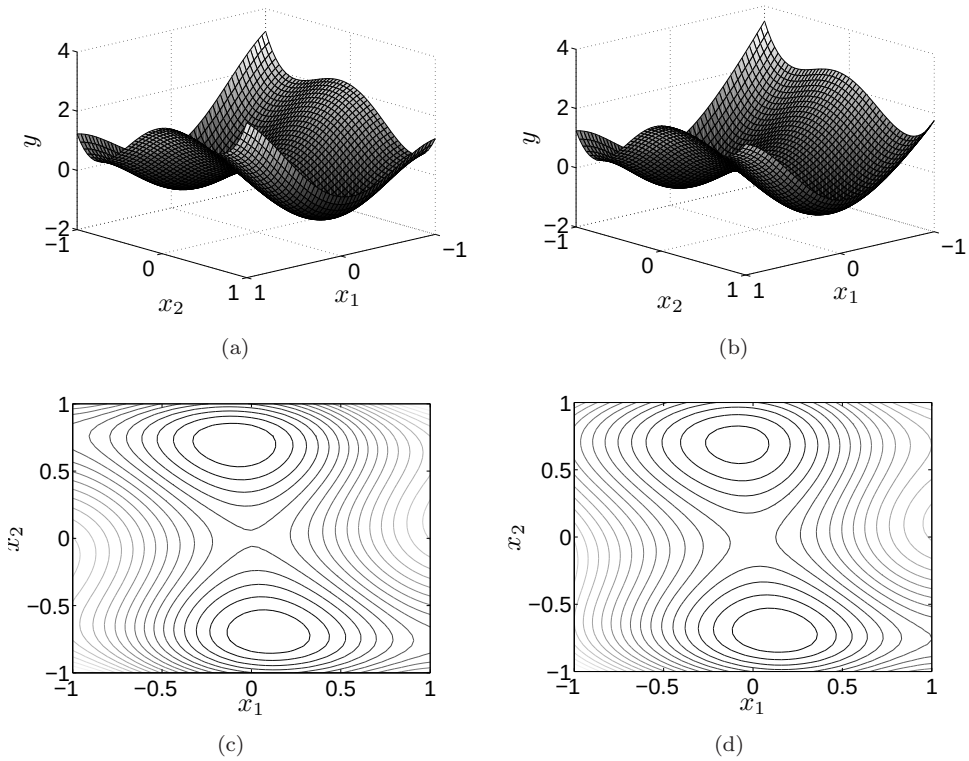


Figure 4.18: Numerical test problem. (a)-(c) True model and its contour plot, (b)-(d) NN model with regularization and its contour plot.

4.4.5 Conclusions

Two-layer FFNNs are capable of approximating any nonlinear differentiable function when a sufficient number of neurons is provided in the hidden layer with the tangent sigmoid basis functions and the linear basis functions in the output layer. NNs do not require any preliminary assumption on the shape of the surrogate model. This is automatically achieved by the utilized basis functions and the number of the hidden layer neurons in the network structure. The Levenberg-Marquardt method provides a rapid convergence to the solution. Over-fitting issues, that might be caused by the improper hidden layer neuron number selection, can be prevented using the Bayesian regularization. On the other hand, increasing the number of the neurons in the hidden layer amplifies the number of the model parameters which slows down the training process.

4.5 Summary and Conclusions

In this chapter, first, the fundamental steps for building surrogate models were presented. Next, three well-known techniques, namely Polynomial, Kriging and Neural Networks (NNs), were introduced and their positive and negative features were discussed on numerical test problems.

The prediction capability of a surrogate model is highly correlated with the complexity of the trend investigated, the number of the design variables, the quality of the available data and the effectiveness of the approaches used for the approximation and the generalization.

When the number of design variables is small, the approximations obtained by the Polynomial, the Kriging and the NN models are all satisfactory.

The majority of engineering design problems are multidimensional and there is generally not sufficient information about the relationship between the selected design variables and the response investigated. For these problems polynomial approximation may become impractical due to the difficulty in the decision of the order of the polynomial and the employed basis functions, as well as the large amount of required data. The Kriging and the NN approximations on the other hand, are much more flexible and hence more appealing for use in these problems.

Kriging and NN are similar approaches in the sense that they both use parametric exponential functions which enable them to approximate highly nonlinear data. Generally, these models require a smaller amount of data compared to the polynomial models which makes them attractive to use as surrogates for computationally expensive models. It is hard to conclude whether one of these models is superior to the other.

The flexibility of the NN approximation is that an arbitrary number of neurons can be used in the hidden layer for a good prediction capability while the generalization capability of the model can still be guaranteed by the employed Bayesian regularization. This means that the initial number of the model parameters is allowed to exceed the number of the data points. Although this is a very nice property, increasing the number of the neurons increases the number of parameters. Consequently, more time is required for solving the nonlinear least squares problem. This is why the gradient-based methods are more preferable than the global optimization methods for calculating these parameters. Since the gradient-based methods do not guarantee the global optimum and the Nguyen-Widrow initialization rule involves random number generators, an NN model may get different parameter values each time it is trained on the same data set. This leads to different model representations.

In the Kriging approximation, the nonlinearity of the model is correlated with the number of data points and their distribution in the design space. Hence, a more evenly distributed and a large number of data will lead to a more accurate surrogate. On the other hand, increasing the number of data points and the design variables slows down the training of the model. The concentrated ln-likelihood function of Kriging is nonlinear and it may have many local optima. The global parameters may not still be

assured even when a global optimization scheme is utilized. This means that different model parameters can be found each time the model is trained on the same data set.

5

Optimization Strategy

The subjects discussed in the previous chapters can be treated independently for different research purposes. In this chapter, these subjects are gathered in order to propose a strategy for optimizing the dynamic behavior of structures. Before that, first, the literature is reviewed briefly for structural optimization and the borders of the strategy are clarified.

5.1 General Concepts in Structural Optimization

Structural optimization concerns the application of numerical optimization techniques (see Chapter 3 for details) to the design of structures. A structural optimization problem can be formulated as

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{subject to} \begin{cases} c_i(\mathbf{x}) = 0 & i \in \mathcal{E} \\ c_i(\mathbf{x}) \leq 0 & i \in \mathcal{I} \end{cases} \tag{5.1}$$

for which the evaluation of the objective function $f(\mathbf{x})$ and/or the constraints $\mathbf{c}(\mathbf{x}) = \{c_i(\mathbf{x}),\, i \in \mathcal{E} \cup \mathcal{I}\}$ requires the use of structural analysis methods (generally, the Finite Element(FE) method). In the problem formulation; weight, displacements, stresses, vibration frequencies, buckling loads or any combination of these entities can be used either as an objective function or as constraints [63]. A structural optimization problem can also be defined with multiple objective functions which is then referred to as *Multiple Criteria Optimization*. Detailed information about this class of optimization problems and the proposed solution methods can be found in [39]. In this research only problems with a single objective function are addressed.

The proposed changes in the computational model under the given constraints are achieved through a vector of variables $\mathbf{x}^{\mathrm{T}} = \{x_1, x_2, \ldots, x_n\}$, called *design variables*. Design variables can be element sizes such as cross-sectional dimensions as well as parameters controlling the geometry of a structure, its material properties, etc. Although $\mathbf{x}$ may take continuous or discrete values, or the mixture of both; this study only considers continuous variations in the design variables.

Numerous structural optimization problems use the same formulation defined in Equation (5.1); however, they can be classified into three main categories: sizing, shape and topology optimization. These categories are determined by the definition of the design variables.

In *sizing optimization*, generally the cross-section properties of the beam elements, the thicknesses of the truss and the shell elements, and the material properties such as density, elastic modulus, etc. are defined as the design variables and optimized without modifying the available mesh, as illustrated in Figure 5.1.

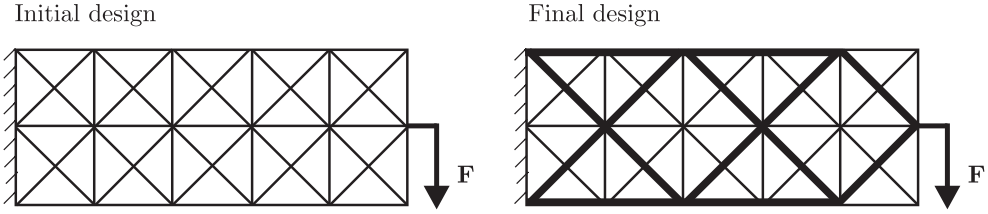


Figure 5.1: Schematic illustration of sizing optimization. Maximizing the stiffness for a given weight where the cross-sectional areas of the trusses are used as design variables. (Left) Initial design, (Right) Final design [12].

In *shape optimization*, the shape of a structure (e.g. the length of a beam or a boundary of a shell) is modified so that the FE mesh is varied with the design changes. On the other hand, new holes or connections that are not present in the initial design domain cannot be introduced during the optimization process. Thus the topology remains the same as shown in Figure 5.2.

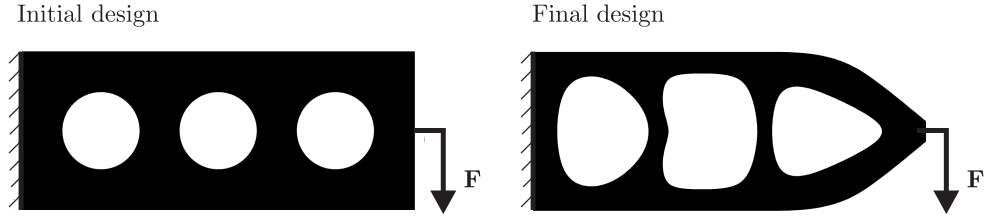


Figure 5.2: Schematic illustration of shape optimization. Maximizing the stiffness for a given weight where the dimensions of the holes and the parameters controlling the boundary of the domain are used as the design variables. (Left) Initial design, (Right) Final design [12].

In *topology optimization*, the topology of a structure is optimized so that the shape and the connectivity of a design domain are altered. Therefore, holes can be placed into the initial design as illustrated in Figure 5.3.

Topology optimization is preferred for conceptual design due to the amount of freedom in representing a product geometry and the difficulties in representing a detailed one. Shape optimization can both be effective for conceptual and detailed design

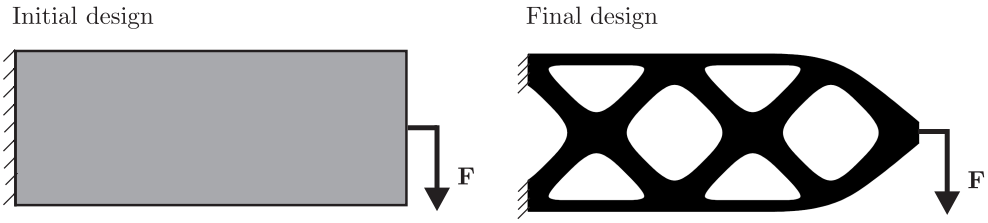


Figure 5.3: Schematic illustration of topology optimization. Maximizing the stiffness for a given weight. (Left) Initial design, (Right) Final design [12].

whereas sizing optimization is more suitable for the detailed design phase [119, 124]. The application of the introduced optimization concepts on the design of aircraft components starting from finding the conceptual design to manufacturing the prototype, and a discussion about the difficulties during optimization can be found in [89]. All three of these optimization concepts are essential to achieve more efficient designs, although this research is restricted to sizing optimization.

5.2 Challenges in Practice and Proposed Solutions

Optimization is an iterative process. The followed steps in an optimization algorithm can be summarized as illustrated in Figure 5.4. Beginning at a given initial design, an optimization algorithm generates a sequence of new designs. Each design in the sequence is a step forward to an optimal solution for which the objective and the constraint functions need to be evaluated many times. This sequence is terminated when either no more progress can be obtained in the designs or the last design in the sequence satisfies the desired accuracy.

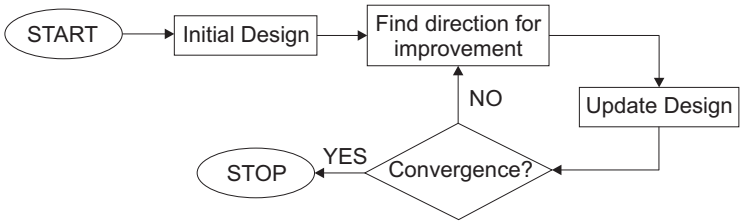


Figure 5.4: The followed steps in an optimization method.

In structural design optimization, evaluation of at least one of the constraints or the objective function is based on analysis methods, in particular the FE method. Generally, high computational costs are associated with the analysis of complex structures. Accordingly, optimization of these structures is a challenging task. For instance, analysis of a car body structure for the dynamics and nonlinearities of a crash event may require weeks for one particular design if executed on a state-of the-art single processor computer [128]. Optimization which needs such an analysis is

computationally very demanding and not feasible. Fortunately, increasing availability of computers with many processors makes the impossible of the past possible today. Although this is very comforting, careful analysis of a posed problem and awareness of the available analytical and numerical methods have a crucial role for obtaining reliable results in a shorter time.

Analysis time is a major drawback for optimization of many structural designs. For speeding up the optimization process, one may think of reducing the number of calls to an analysis method. It is shown in Chapter 3 that pursuit of the global optimum needs a high number of function evaluations compared to the local gradient-based optimization methods. Direct coupling of an analysis method with neither a global optimization scheme nor a derivative free method is efficient due to the number of the required function evaluations and the computationally demanding analysis. Hence, if it is feasible to calculate the derivatives, the gradient-based methods should be used in direct coupling. Selection of a suitable and an efficient gradient-based method is important in order to reduce the number of calls to the analysis method. This is discussed further in Section 5.2.1.

The algorithms developed for problems having a small number of design variables may not be suitable and efficient for the ones having a large number of design variables. The proposed solutions in the literature for these type of problems are introduced briefly in Section 5.2.2.

Another option for speeding up the optimization process is to employ accurate and computationally efficient analysis and reanalysis methods. Section 5.2.3 involves a discussion on these topics.

5.2.1 Gradient-based Algorithms

It is possible to find many gradient-based algorithms in the literature. Selecting a suitable and an efficient one for the considered problem has a big impact on the total number of the FE analysis calls during optimization.

A characteristic of a large class of early methods is the translation of a constrained problem to an unconstrained one by introducing the constraints (near or beyond the constraint boundary) to the objective function with the penalty methods. In this way, a constrained problem is solved using a sequence of parameterized unconstrained problems. These methods are now considered to be relatively inefficient and have been replaced by *Sequential Approximate Optimization* (SAO) methods [2, 63, 74].

The essential idea behind the SAO methods is solving an optimization problem by generating a series of subproblems in which the objective and the constraint functions are the approximations of the original ones and are easier to solve. The SAO methods can be grouped into *local* and *midrange* concepts [11].

In *local approximations*, subproblems are based on function values and derivative values with respect to the design variables computed in a single point of the design space. For instance, in *Sequential Linear Programming* (SLP), a Nonlinear Programming (NLP) problem (see Chapter 3 for details) is approximated at the

current design point $\mathbf{x}_{(k)}$ by a Linear Programming (LP) subproblem in which the objective function and the constraints are replaced with their first order Taylor series approximations at $\mathbf{x}_{(k)}$. The minimum of this subproblem is used to define a new design point $\mathbf{x}_{(k+1)}$ and accordingly a new subproblem. By repeating these steps, a sequence of these design points $\{\mathbf{x}_{(k)}\}_{k=0}^{\infty}$ approaches the optimum design point in the limit [63]. Another local approximation method, the Method of Moving Asymptotes (MMA) [130], solves NLP problems by generating convex and separable subproblems. The parameters (asymptotes) that control the convex curvatures are changed in each subproblem formulation.

In *midrange approximations*, multiple points of the design space are utilized to generate the subproblems. They can be further grouped into *single-point-path* and *multi-point-path* methods [42].

In *single-point-path* methods, subproblems are built using the function values and the derivative values computed in the current design point and also in the previous ones. Employing BFGS method [104] for accurate calculation of the approximate second order derivatives, Sequential Quadratic Programming (SQP) (see Chapter 3 for details) is a single-point-path midrange approximation method. It is stated in [61] that utilizing the second order approximations in an optimization algorithm reduces the total number of the iterations by about 10 to 50 percent compared to the first order approximations.

In a *multi-point-path* method, the objective and/or the constraint functions of each subproblem are replaced with their approximations (for instance, polynomial functions - see also Section 5.3) which are defined using a set of data generated within a given search subregion. For each subproblem, a new search subregion is selected to ensure the convergence of the method. This is done on the basis of a *move limit* strategy [135]. These methods are effective especially when the derivatives can not be computed accurately. The multi-point-path method proposed by Toropov et al. [131] uses linear combinations of predefined basis functions to approximate the objective and the constraint functions of the subproblems. Another multi-point-path method has been presented by Etman which is utilized for optimizing multi-body systems [42].

It is always desirable to converge to a solution from a remote starting point in the gradient-based SAO methods. This is referred to as *global convergence* and some of the proposed strategies for it can be found in [59, 104].

In the gradient-based optimization methods, the derivatives of the objective function and the constraints with respect to the design variables are required for each design change. These derivatives predict the changes in the structure response for each design modification and are used to select search directions in the optimization process. Therefore efficient and accurate calculation of the derivatives are very important. There are various ways for their computation such as: *finite difference approximations*, *analytical methods* and *semi-analytical methods* [62, 80, 134].

The simplest and the easiest method to implement is the finite difference approximation. Unfortunately, its calculation involves numerous repeated analysis and high computational costs particularly during the optimization of the large structural systems with many design variables. Additionally, accuracy problems can

be encountered due to the approximations and the step size selection [84].

Analytical methods provide exact solutions and reduce the number of the analysis calls drastically in optimization processes; however, they cannot be easily obtained for many complex problems [97]. Moreover, access to the source code of the FE software is required for their implementation.

To summarize, selecting a suitable optimization algorithm as well as having methods for efficient and accurate calculation of the derivatives are essential especially for optimization of structures involving expensive analysis and many design variables.

5.2.2 Large-scale Optimization

Optimization problems having a large number of design variables are treated under the concept of *large-scale optimization*.

By referring to [28, 56], the notion *large* depends on computer resources, problem definition and also the structure of the problem. Generally, an NLP problem with the order of hundred thousand variables and a similar number of constraints is considered as large, while an NLP problem up to fifty design variables is categorized as small-scale.

Small-scale gradient-based optimization methods are not directly applicable on the large-scale optimization problems because the approximation for the Hessian or its inverse is usually dense in these methods [104]. Storage requirements grow with a rate more than linear with the increasing number of the design variables and become excessive [63].

Small-scale optimization methods can be improved in several ways in order to be utilized for large-scale problems. One way is using simple and compact approximations for the Hessian matrices. Thus, only a few vectors that represent the Hessian are saved instead of keeping the full matrices [56, 104]. Accordingly, storage requirements can be reduced to a high extent. Another way is *decomposing* a large optimization problem into a series of small-scale subproblems [27, 132]. For the solution, a two stage process is carried out: first, decomposed small-scale problems are solved individually. Then, they are coordinated so as to optimize the entire problem. The coordination level is usually referred to as the top level. Subproblems are selected in a way that they are connected through a small number of design variables to the top level. Decomposed problems may correspond to either the physical systems (components of a structure) or the disciplines. The optimization process based on decomposition is also referred to as *Multi-level optimization* in the literature [63].

5.2.3 Reduction and Reanalysis Methods: Focus on FE models

In an FE model, the number of degrees of freedom (d.o.f.) grows rapidly with the amount of detail and accuracy investigated from the analysis. As stated in [62], computation time increases almost quadratically with the number of d.o.f. Therefore

a good compromise between the demanded level of accuracy from an analysis and the permissible computation time is important.

A simple approach can be either using simple geometries instead of complex ones or having a complex geometry with a coarse mesh. Analyses based on such approaches may give a rough idea about the modeled physical behavior. Their fidelity is low in general. Methods discussed under the field of *multi-fidelity approximations* concern the adaptive use of high-fidelity expensive-to-evaluate analysis models with low-fidelity fast-to-evaluate ones in optimization algorithms in order to increase the computational efficiency while preserving a certain accuracy. A detailed review on this subject can be found in [123].

For certain analysis, accuracy is the essential priority. For instance, in automobile designs dynamic analysis is a useful method for accurate prediction of dynamic stresses on the chassis during many types of maneuvers [79]. For such an analysis, the full FE model of the automobile is used. The model is often made very large in order to capture local stresses accurately. Generally, investigation of the structural behavior for static analysis requires fine meshes. Contrarily, investigations on the dynamic properties of the same structure usually include calculating a few deformation modes. These modes can easily be captured using much less d.o.f. Hence, the number of d.o.f. in an FE model can be reduced by employing suitable reduction methods without changing the mesh.

As discussed in Chapter 2, Component Mode Synthesis (CMS) is a frequently used reduction and sub-structuring method in the dynamic analysis which usually meets high standards of computational efficiency and accuracy in the low frequency range [21, 34, 35]. CMS decreases the analysis time, accordingly it reduces the overall optimization time significantly [37, 40, 102].

There are different applications of CMS in the optimization literature. In [68], it is used for developing methodologies for the calculation of the eigenvalue and eigenvector derivatives. In [72], a design optimization method is proposed where the idea behind CMS is integrated into a two-level design optimization scheme. The subproblems are defined on the component models. The modes defining the component transformation matrices are considered as the top level (intermediate) design variables. At the subsystem (component) level, the element properties of the components are the primary design variables. The intermediate design variables of each component are defined as the functions of the primary design variables of that component. Coordination of the sub-optimization problems with the main optimization problem is already defined in the CMS technique. An optimization problem based on that two-level scheme could be stated as maximizing the first global eigenvalue with a minimal change in the intermediate design variables at the top level. At the subsystem level, the objective of the subproblems can be: minimizing the error between the target and the current intermediate design variables by changing the primary design variables under a mass constraint [72]. In this scheme, in addition to the issues concerning the accuracy of the results as already reported in [72], the detrimental effect of the number of the intermediate design variables on the efficiency of the method is another concern. The number of d.o.f. of the components increases with the complexity of the modeled structure. Thus, the number of the intermediate design variables may easily

exceed the total number of the component design variables which is contradictory to the purpose of decomposition.

For reducing the analysis time during optimization, employing reanalysis methods can also be very useful. The objective of these methods is to evaluate the structural response due to the modifications in the design variables using knowledge of the initial model, thereby, solving a complete set of new equations is avoided. More information on the developed reanalysis methods for the eigenvalue problems can be found in [83, 85]. A discussion about the integration of the reanalysis methods into CMS and the proposed reanalysis method can be found in Chapter 2.

5.3 Surrogate-Based Optimization

As discussed in the previous section, direct coupling of a local or a single-path gradient-based algorithm with an analysis method still does not have any other alternatives on the optimization of large scale, non-separable problems. On the other hand, efficient and accurate calculation of the derivatives are remaining issues in their application. Moreover, the analysis of a structure may fail when some of the design values are not feasible. For instance, direct coupling of a Branch and Bound type mixed integer NLP algorithm with FEM might be problematic when non-integer values are assigned to some of the design variables.

The motivation of *Surrogate-Based Optimization* (SBO) is: *replacing expensive-to-evaluate FE models with their fast-to-evaluate approximations in optimization problems.* These approximations are known as *meta-models*, *response surfaces* or *surrogate models* in the literature. When they are defined on the overall design domain, they are also called *global approximations.* Meta-models are built to predict the trends in the data collected from an FE model. These data consist of a set of values for the selected design variables and the response of the structure for these design values. Therefore, surrogate models can be considered as highly simplified versions of FE models. Details about surrogate model selection and a discussion about the commonly employed methods, namely polynomial, Kriging and Neural Network (NN) approximations, can be found in Chapter 4.

In order to generate the training data for surrogate models, sample points are selected from a feasible domain that is constrained by the upper and the lower bounds of the design variables. This ensures that with carefully selected points, failure of the FE model is prevented. Data generation is still the challenging step of meta-modeling. For obtaining certain accuracy, the total number of the data should be sufficient. On the other hand, with an increasing number of the design variables, the required number of data grows rapidly. As discussed in Chapter 4, this number can be very high for polynomial approximations. Kriging and NN approximations, on the other hand, are more flexible so that they demand relatively less data for accurate representation of the complex underlying behaviors. The definition of "sufficient number of data points" is highly correlated with the complexity of the approximated behavior. To the author's knowledge there is no rule in the literature that defines this number.

In this research it is kept around *ten times the number of the design variables* as recommended in [91].

Once a surrogate model is built, it is many orders of magnitude faster to evaluate than an FE model. Therefore, it can be effectively employed in global optimization schemes. The number of function evaluations in an optimization algorithm is not a big issue due to the simplicity of the surrogate models.

Analytical derivatives of FE models are not required for building the surrogates. Additionally, analytical derivatives of surrogates are not essential during optimization. Derivatives of these can accurately be calculated by the finite difference approximation.

When an optimization algorithm is directly coupled with an FE model, evaluation of the model is done sequentially during the search of an optimum. On the contrary, an FE model is only required for generating data for meta-modeling in SBO. Hence, these data can be gathered all at once by parallel processing.

The application of the SBO schemes is still restricted to the small-scale optimization problems due to computationally demanding analyses and the amount of the required data for training surrogates. On the other hand, when combined with decomposition methods, using SBO for large-scale optimization problems seems very promising.

In SBO schemes where the approximations are defined on the overall design domain, the followed steps for finding an optimal solution are similar and can be summarized as shown in Figure 5.5.

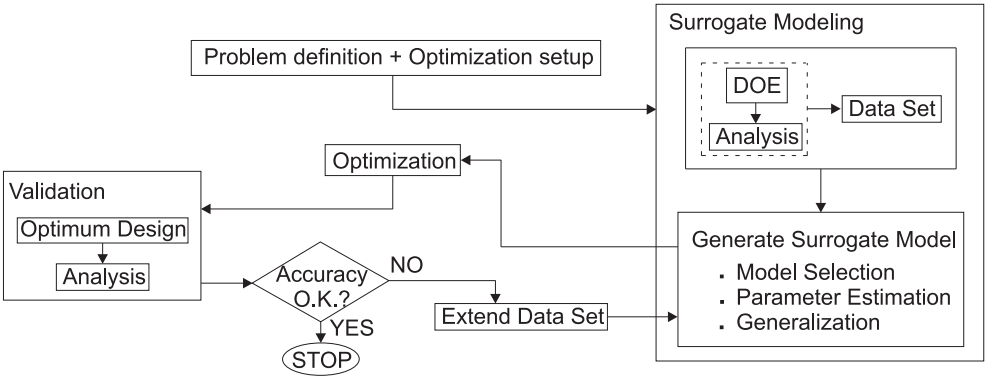


Figure 5.5: Schematic illustration of a general SBO method.

The solution process starts with the problem analysis which involves, first, understanding the problem under consideration. Then, selection of the design variables and parameterization of the computational model are carried out. Finally the objective function and the constraints are defined.

The second step is to generate the surrogate model. Here, firstly a set of sample points is selected from the design space which is called Design of Experiments (DOE) (see Section 4.1.1 for details). Afterwards, for each sample point the FE model is

run and data is gathered for training the surrogate. Then, a suitable meta-modeling approach (Polynomial, Kriging, NN, etc.) is selected and the unknown parameters of the chosen meta-model are determined using the available data (see Chapter 4 for details).

Having generated the surrogate model, the next step is the optimization which is carried out employing suitable numerical optimization methods.

Since the calculated optimum is not directly related with the FE model but the surrogate model, the results need to be validated. In order to do that, the response of the FE model is obtained by the computed optimum design values. This is then compared with the response of the surrogate model for the same design values. If the accuracy is acceptable, the scheme is stopped. Otherwise, the data set is extended with the optimum design values and the corresponding response of the FE model. When parallel processing is possible, one may also add multiple data into the set calculated at the other promising regions of the design space. New parameters for the selected surrogate model are computed using the extended data set and the optimization step is repeated. This procedure is iterated until the validation results are acceptable.

The SBO methods differ from each other by the followed approaches during the generation of the surrogate model and the utilized algorithms at the optimization step [111]. For instance, in [76], a method based on the Kriging approximation at the surrogate modeling phase and the Branch and Bound (a mixed integer nonlinear programming) algorithm for optimization is introduced. Berke et al. [14] used the NN approximations for surrogate modeling and the gradient-based algorithms for optimization. Another SBO method is the one proposed by Booker et al. [19] in which the Kriging approximations are utilized for meta-modeling and the pattern search (a derivative free optimization) method is employed during the search of an optimum.

Wind et al. [139] presented a two-level SBO method for optimizing the dynamic behavior of structures. In the scheme, both the analysis and the optimization problem are decomposed using CMS. Optimization is performed only at the component level where NN approximations are utilized with Genetic Algorithms (GAs). In the decomposition strategy, each subproblem is the same as the main optimization problem except that the design variables that do not correspond to the component of the interest are fixed at constant values. Therefore, the complications that might be encountered due to the coordination of the subproblems are prevented by eliminating the intermediate design variables. In the method, the top level is only used for the distribution of the component level solutions to each subproblem. The drawback of this strategy is the possibility of convergence problems especially when the response of one component depends on the design variables of other components [138].

5.4 The Proposed Strategy

As stated in Chapter 1, the objective of this research is to develop an effective strategy for optimizing the dynamic behavior of structures. The proposed strategy is based

on a "*Surrogate-Based Optimization*" (SBO) method that is introduced in Section 5.3 in which global approximations are utilized as surrogates. In the analysis phase of the method, a CMS technique, Craig-Bampton (CB), is used for offering solutions to one of the major difficulties, *the analysis time*, in structural optimization. Moreover, reanalysis methods are employed in CB to reduce the analysis time even further.

The benefits of employing an SBO scheme for structural optimization problems have already been discussed in the previous section. Using a CMS technique in combination with SBO has the following additional computational advantages:

- Reduction in the total number of d.o.f. leads to fast analysis of the complete structure.
- Independent condensation of each substructure encourages parallel processing even further.
- Preventing unnecessary calculations for the unmodified substructures, only the modified components can be analyzed and coupled with the already computed ones.
- For structures having repeated components (e.g. one cyclic sector of an industrial blisk), modeling of one component is sufficient.

All these discussed items have a significant effect on the reduction of the analysis time. Moreover, when reanalysis methods are utilized, analysis of the modified substructures are performed without solving the complete set of new equations. This brings supplementary benefits for saving computation time.

The steps in an SBO method are:

- DOE
- Analysis
- Surrogate Model generation
- Optimization
- Validation

as also shown in Figure 5.5. These steps are treated exclusively for introducing the details of the proposed strategy.

Design of Experiments

In the research, the data set required for generating a surrogate model is obtained from FE analyses. Therefore, Design of Experiments (DOE) is based on a space-filling strategy for which the Latin Hypercube Sampling (LHS) scheme is utilized (see Section 4.1.1 for details). The number of the sample points is selected around *ten times the total number of the design variables*. These points are generated in the overall domain restricted by the upper and the lower bounds of the design variables.

Analysis

At the *analysis* step of the proposed SBO method, the Craig-Bampton (CB) method is used as a CMS technique. Furthermore, reanalysis methods are considered for efficient calculation of the CB transformation matrices of the modified components. Details of the employed methods can be found in Chapter 2.

The followed steps at the analysis phase are shown schematically in Figure 5.6. For the dynamic analysis of a structure, first, the complete structure is divided into components. Then the parameterized FE model of each component is built. If there are similar components, only one of them is modeled. Afterwards, the design values of the complete structure are distributed to components based on the design variables captured in the component models. An FE model standing for similar components may get multiple configurations for its design variables. The next step is the calculation of the reduced system matrices of each component for the assigned design values. In the proposed scheme, libraries are used to store the information about the already analyzed components. Hence, unnecessary analyses are prevented. Before generating the system matrices of a given component design, first, the corresponding library is checked. If the requested information is not there, it is computed and stored in the library. In the computation, first of all, the transformation matrix, consisting of the normal and the constraint modes, is calculated. The normal modes can be computed either using the exact analysis methods or using the Enriched Craig-Bampton (ECB) method. Unfortunately, there is no automated switch from ECB to the exact methods based on the accuracy and/or the computational efficiency of ECB. On the other hand, calculation of the constraint modes, either by the exact or the approximate methods, can be automated as introduced in Section 2.3.2. The approximate constraint modes are calculated using a method based on the Combined Approximations (CA) approach. Details about ECB and the CA-based approach can be found in Chapter 2. After the transformation matrix is determined, the reduced component matrices are computed. The given component design, its transformation and the reduced matrices are saved in the component library. This procedure is repeated for each component and the corresponding configurations. This ensures that all the necessary information to generate the reduced system matrices of the complete structure is readily available in the libraries for further use. Thereafter, the stored reduced matrices are gathered from the libraries for the given structure design and assembled to obtain the reduced matrices of the entire structure. Finally, the dynamic analysis of the structure is performed.

Generate Surrogate Model

In the scheme, the Kriging and the Neural Network (NN) surrogate modeling approaches are used. A detailed discussion about these approaches is given in Chapter 4.

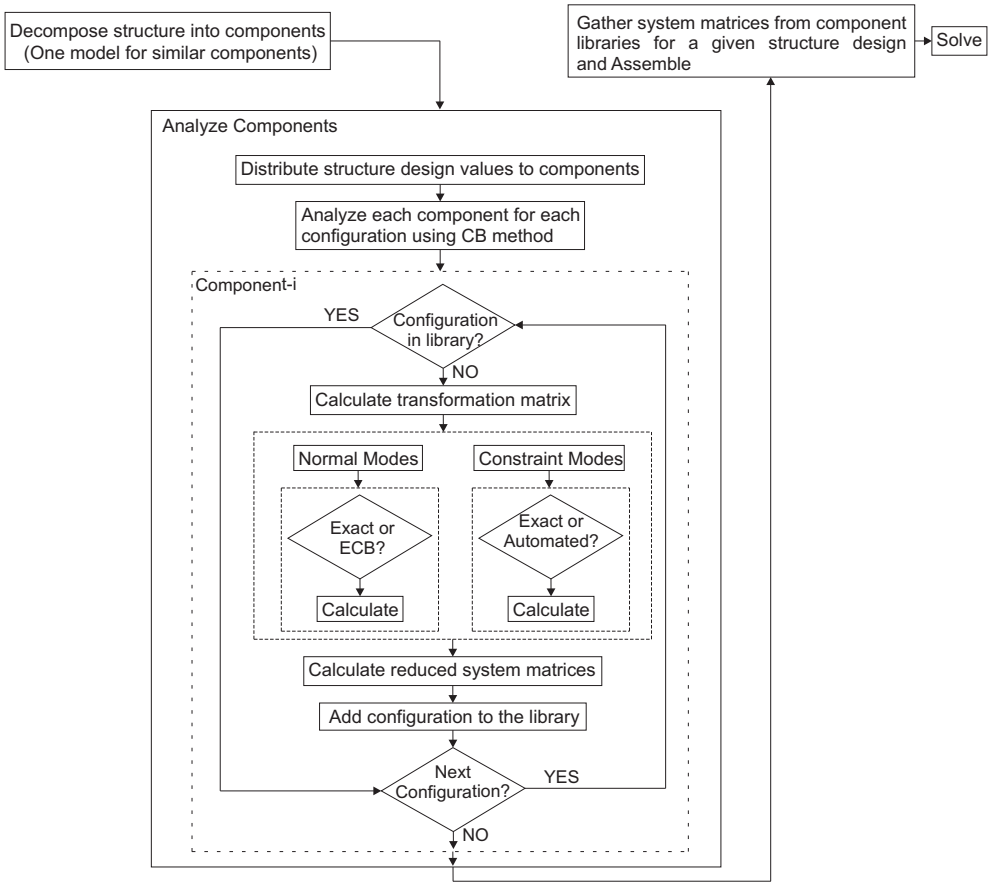


Figure 5.6: Schematic illustration of the analysis step of the proposed SBO method.

Optimization

At the *optimization* step, the global optimum can be searched using either the Multi-Level Hybrid Optimization (MLHO) or the Multi-Start Local Optimization (MSLO) schemes (see Section 3.4 for details). It is also possible to run these schemes together for increasing the chance of finding the global optimum even more.

For the MSLO scheme, designs in the DOE set are utilized as multiple starting points and a gradient-based method, Sequential Quadratic Programming (SQP) is used as a solver. In the MLHO scheme, a stochastic derivative-free global optimization method, the Genetic Algorithm (GA), is employed to locate the global optimum and SQP is initialized with the solution of GA to find an exact optimum solution. Details about these optimization algorithms and a discussion about their performance can be found in Chapter 3.

Validation

For the *validation* of the optimization results, the FE analysis is run for the calculated optimum design. If the responses of the FE model do not agree with the responses of the surrogates, the data set is extended with the optimum design values and the corresponding responses obtained from the analysis.

5.5 Discussions about the Strategy

In structural design optimization, the number of design variables is one of the main factors affecting the efficiency of the process. In some cases the actual response of the structure is not affected significantly by some of the design variables. Then it becomes reasonable to reduce the number of the design variables before solving the optimization problem. In order to assess the impact of each variable on the response, data need to be generated from the FE model for varying values of these design variables. This is usually referred to as *sensitivity analysis* and it is a very critical process. If not done correctly, "important" design variables may be eliminated due to the lack of data. This research does not cover the methods used for the identification of important design variables whereas some relevant research about this subject can be found in [70, 137].

As mentioned earlier in the chapter, the proposed method is restricted to sizing optimization problems. This is partly due to the analysis and the reanalysis methods utilized at the analysis step. The assembly phase of the CB method is based on the compatibility of the interface nodes. That is, if there is a mismatch between the interface nodes of the neighboring components, analysis with the CB method leads to inaccurate results. In sizing optimization, since the mesh does not change, the position of the interface nodes are always protected. Thus, the CB method and the reanalysis methods can safely be employed. When the analysis phase of the proposed method is extended with methods taking into account the non-matching interfaces [43, 108, 115], the SBO scheme can also be used for the shape optimization problems. On the other hand, the reanalysis methods have to be reconsidered again. Substantial changes in the geometry of components may result in completely different mode shapes which cannot be approximated accurately with the current reanalysis methods.

In the method, the ECB approach is used for the reanalysis of the fixed interface normal modes of the modified components. Unfortunately, there is no automated rule in the method that switches to the exact calculations when the accuracy and/or the computational efficiency of ECB is not satisfactory. Considering alternative update methods for the normal modes may help to automate the computation of these modes.

At the surrogate modeling step, a single meta-modeling approach is used to generate the surrogates. On the other hand, once the data set is gathered, finding the parameters of a selected surrogate is relatively cheap. This is very encouraging for utilizing multiple meta-models in an optimization scheme, although it is not studied in this research. Employing multiple surrogate models will bring the following advantages:

- When one approach fails to find an acceptable representation of the underlying behavior, the other one may compensate for the gap.
- Since different approaches may identify distinct parts of the design space better than the other, using several ones synchronously may lead to more accurate approximations of the trend investigated.

The challenge in "multiple surrogate" based optimization is to define a strategy which can evaluate multiple meta-models simultaneously and combine them so the generated model gives much better predictions than any of the individual meta-models. The related studies on this subject focus on using cross-validation error for the evaluation of the surrogates [53, 136, 141]. Hence, the meta-modeling approaches which use regularization methods are completely missed out in these evaluation schemes. Many questions have to be answered on this subject, which makes it very interesting for future research.

At the validation step, when a desired accuracy is not obtained, the data set is extended only with the calculated optimum design values and the corresponding FE model response. There are other strategies in the literature which are based on estimating the error due to the meta-model approximation and positioning the new data at the points of maximum estimated error. A detailed discussion about these strategies can be found in [46, 47].

5.6 Summary and Conclusions

Topics covered under the concept of structural optimization were introduced briefly, the proposed optimization method was presented and future research topics were discussed.

Two approaches in structural optimization were emphasized. The first approach is the direct coupling of an FE model with numerical optimization algorithms, which is inevitable for large-scale, strongly coupled optimization problems. For this case, the gradient-based algorithms are the most suitable methods because they require less function evaluations than the derivative-free algorithms. The second approach is the Surrogate-Based Optimization (SBO) scheme in which FE models are replaced with their fast-to-evaluate surrogates in the optimization problems. This scheme is very promising for multi-level and small-scale optimization problems. The search for the global optimum is affordable in SBO as opposed to the direct coupling methods. In direct coupling, evaluation of an FE model is done sequentially during the search of an optimum whereas in SBO, an FE model is utilized to generate data for meta-models. Since sequential evaluation of an FE model is not necessary, data can be generated all at once by parallel processing.

The proposed method is based on an SBO scheme where the employed approximations are defined on the overall design domain. In the method, the Craig-Bampton (CB) method is used at the analysis step which has the advantages of:

- parallel processing due to independent component analysis,

- shorter analysis time,
- freedom on not evaluating the unmodified substructures all over again during optimization,
- opportunity to generate one model that can be used for similar substructures.

Moreover, the CB method is combined with reanalysis methods to give even more reduction in the analysis and hence in the optimization time.

Kriging and Neural Networks (NNs) are the meta-modeling techniques used in the SBO method. The search for the global optimum can be done by two different schemes: The Multi-Level Hybrid Optimization (MLHO) and the Multi-Start Local Optimization (MSLO). Genetic Algorithms and Sequential Quadratic Programming are the numerical optimization methods employed in these schemes.

In the next chapter, the proposed method will be demonstrated on some academic problems and the corresponding results will be discussed in details.

6

Demonstration of the Strategy

The proposed Surrogate-Based Optimization (SBO) strategy is demonstrated using two academic test problems. In both of the problems, the selected structures have repeating patterns in their geometries. These special geometries are selected so that the modeling and the analysis opportunities of the Craig-Bampton (CB) method are fully exploited during optimization. In the problems, the harmonic response and the modal properties of the structures are intended to be modified.

6.1 Optimization of Structures with Repetitive Component Patterns

In the first test problem focus is on investigating the advantages of using the CB method in the proposed SBO scheme. In order to analyze the effectiveness of the method only, the fixed interface normal modes and the constraint modes of the CB transformation matrix are computed exactly. That is, the reanalysis methods are not considered in this problem. Furthermore, this example is used to investigate the effects of the chosen surrogate model on the final results of the SBO method.

The optimization problem involves the objective of reducing the total mass of the structure. Meanwhile, one of the natural frequencies is constrained to have a certain value where this value is chosen to be different than that of the initial design. Another constraint is to preserve the initial mode shape of the corresponding natural frequency.

Parts of this chapter are based on:
"D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, P.J.M. Hoogt van der and A. de Boer. *An optimization method for dynamics of structures with repetitive component patterns.* Structural and Multidisciplinary Optimization, 39(6):557-567, 2009" [5],
"D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek and A. de Boer. *A hybrid design optimization method using enriched Craig-Bampton approach.* In 16th International Congress on Sound and Vibration, ICSV, 2009" [3].

The selected problem formulation is generic in the sense that it is possible to enhance the formulation with multiple frequencies and the corresponding mode shapes.

Potential application areas of such problem definitions can be various, for instance, avoiding resonance conditions. A structure is said to be in *resonance* when one of its natural frequencies coincides with the frequency of a dynamic excitation force. The amplitude of vibration under resonance conditions becomes excessive. To ensure safety and efficiency, the natural frequency of a structure has to be separated from the frequency range of the excitation force. The proposed optimization method provides a robust algorithm to solve this sort of problem.

6.1.1 Problem Definition

The selected structure, as seen in Figure 6.1, resembles a fan inlet case and it consists of 24 components (substructures). The thicknesses of the strut, t_{strut}, the inner ring, t_{inner} and the outer ring, t_{outer}, of each component are selected as design variables. The substructures which have $n\frac{\pi}{2}$, $n = 0, \ldots, 3$ rotational distance between each other are imposed to have the same design variables which are shown by a unique color in Figure 6.1. This is done in order to ensure that the substructure design repeats itself in every quarter. Accordingly, there are 6 different components with 3 design variables in the overall structure which are identified with numbers in Figure 6.1. This makes 18 design variables in total. The structure is free-free i.e. there are no defined boundary conditions on the structure.

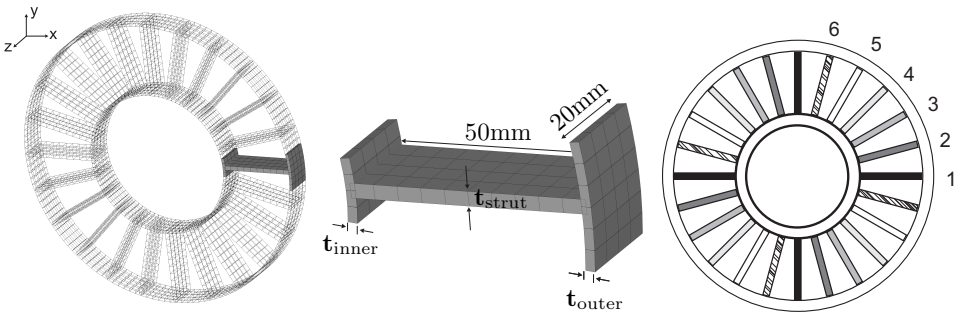


Figure 6.1: Test problem. (Left) Selected structure and its repeating component, (Middle) The three design variables of a component, (Right) The repeating geometry. Matching colors represent the components with the same design variables.

It is clear that all the substructures are identical in the local coordinates when assigned with the same design values. Accordingly, the stiffness and the mass matrices of each substructure are all the same in the local frame. This makes sure that the reduced FE model of the entire structure can be obtained using the parametric FE model of one component and rotating the computed matrices for each substructure to its global coordinate system.

In this example, the reduced system matrices of different component configurations are generated using the commercial FE software ANSYS. Assigning these matrices to the rest of the substructures by multiplying them with the corresponding rotation matrices, assembling the substructure system matrices for each structure design and solving the eigenvalue problem are performed in MATLAB.

In the FE model, a 4 node *shell* element is used which is known to be suitable for analyzing thin to moderately thick shell structures [7]. The element has 6 d.o.f. at each node which are the translations and the rotations about the x-, y-, z-axes. The in-plane vibrations of the ring structure are the only concern for this problem. Therefore, the rotations about the x-, y-axes and the translations in the z-direction are suppressed in the element.

Each component has 65 d.o.f. after reduction and there are 1560 d.o.f in the reduced model of the structure. The total number of the fixed interface normal modes and the nodes on the interfaces of a component are 5 and 20, respectively. The full FE model of the structure without reduction has 6840 d.o.f. Employing the CB method therefore results in 77% reduction in the total number of d.o.f.

The selected material properties are as follows: Young's modulus (E) is 116 GPa., Poisson's ratio (ν) is 0.3 and the density (ρ) is 4500 kg/m^3.

In the initial design, the components have an inner ring thickness, t^i_{inner}, of 2 mm, a strut thickness, t^i_{strut}, of 3 mm and an outer ring thickness, t^i_{outer}, of 2 mm where $i = 1, 2, \ldots, 6$ stands for the component number. The total mass of the initial model is 0.4936 kg and the 5th natural frequency (2nd bending frequency) is 702.23 Hz with a mode shape illustrated in Figure 6.5(a). Since this is a free-free structure in the xy-plane, the first three modes are rigid body modes. The 4th natural frequency is identical with the 5th one due to the symmetry.

In the optimization problem, the total mass of the entire structure is to be minimized by adjusting the thicknesses within the specified ranges (see Equation (6.1)). In the meantime, the 5th natural frequency has to be increased from 702.23 Hz to 750 Hz while preserving the 5th mode shape of the initial design.

The optimization problem is formulated as follows:

$$\min_{\mathbf{t}} \left[\rho \mathrm{V}(\mathbf{t})\right] \quad \text{subject to} \begin{cases} f_5(\mathbf{t}) = 750 \\ \mathrm{MAC}_5(\mathbf{t}) \geq 0.9 \\ 1 \leq t^i_{\text{inner}} \leq 5 \\ 1 \leq t^i_{\text{strut}} \leq 5 \\ 1 \leq t^i_{\text{outer}} \leq 5 \quad i = 1, 2, \ldots, 6. \end{cases} \tag{6.1}$$

In Equation (6.1), $\mathrm{V}(\mathbf{t})$ represents the volume of the entire structure which is a linear function of the thicknesses $\mathbf{t} = \{t^i_{\text{inner}}, t^i_{\text{strut}}, t^i_{\text{outer}} \,|\, i = 1, 2, \ldots, 6\}$. In order to preserve the 5th mode shape of the initial design, the *Modal Assurance Criterion* (MAC) is used (see Section 2.4.1 for details).

The structure under consideration has a rotational symmetry in its geometry. It may have an eigenvalue with multiple eigenvectors. These eigenvectors are linearly independent and any linear combination of these eigenvectors also has the same

eigenvalue [48]. Therefore, instead of using the mode shape of the 5th natural frequency directly in the MAC formulation (see Equation (2.39) for the MAC formula), first, the uniqueness of the frequency is inspected. This is achieved by comparing the 5th natural frequency with the 4th and the 6th. If it is found to be unique, the corresponding eigenvector is used in Equation (2.39). If multiple eigenvectors are associated with this frequency, then these are gathered and orthogonalized in order to obtain an orthogonal basis. Afterwards, the most complying vector with the 5th eigenvector of the initial design is sought in a subspace spanned by this basis. This latter vector is applied in the MAC constraint of Equation (2.39).

Details of the formulation on finding the most complying eigenvector and the computation of the MAC value are presented in Appendix D.

6.1.2 Surrogate Modeling

In the optimization problem, 2 surrogate models are used. They stand for $f_5(\mathbf{t})$ and $\mathrm{MAC}_5(\mathbf{t})$ constraint functions given in Equation (6.1). To analyze the effects of the chosen meta-modeling approach on the performance of the SBO method, surrogates are generated by both the Kriging and the NN approaches. In the NN model, 25 hidden layer neurons are utilized. In the Kriging model, the Multi-Level Hybrid Optimization (MLHO) scheme (see Section 3.4 for details) is used to compute the model parameters $\boldsymbol{\theta}$. The entries p_j, $j = 1, 2, \ldots, 18$ of the exponent vector $\mathbf{p}$ are selected as 2.

The training data for the meta-models are generated by the parametric FE model of one component only. First, a Design of Experiment (DOE) set, $\mathbf{D}_\mathrm{C}$, with 180 designs is generated for the component using the Latin Hypercube Sampling (LHS) scheme. Each configuration in $\mathbf{D}_\mathrm{C}$ defines a new substructure design. The reduced substructure stiffness and mass matrices are calculated for each design in $\mathbf{D}_\mathrm{C}$ and stored in the library. Each configuration in the DOE set, $\mathbf{D}_\mathrm{T}$, of the entire structure is defined by randomly selecting 6 component designs from $\mathbf{D}_\mathrm{C}$. In this example, 180 structure designs are chosen in total for $\mathbf{D}_\mathrm{T}$. Next, the reduced system matrices of the entire structure for each design in $\mathbf{D}_\mathrm{T}$ are gathered. The related component system matrices are called from the library, multiplied by the rotation matrices to transfer the components to the global coordinates and assembled. Dynamic analysis is performed on these assembled matrices to compute a set of 5th natural frequencies and the corresponding mode shapes. $\mathbf{D}_\mathrm{T}$ and the frequency set are utilized to train the surrogate model which replaces $f_5(\mathbf{t})$ in Equation (6.1). The MAC values are computed using the mode shape information and afterwards, $\mathbf{D}_\mathrm{T}$ and the MAC set are employed to train the surrogate model that will take the place of $\mathrm{MAC}_5(\mathbf{t})$ in the posed problem.

6.1.3 Optimization and Validation

The parameters of both the Kriging and the NN models are computed using methods which involve random number generators. Therefore, the calculated surrogate models

may not be unique. For investigating the effects of the surrogate modeling approaches on the robustness of the SBO strategy, the optimization problem is solved 10 times in total. 5 of the solutions are obtained using NN as the surrogate models and the remaining 5 are obtained using Kriging. The performance of the strategy is discussed based on the obtained results.

At the optimization step, the Multi-Level Hybrid Optimization (MLHO) scheme is utilized for finding the global optimum.

The search for optimum is repeated until the relative errors between the responses of the FE model and that of the surrogates are smaller than 0.005 for the computed optimum design values. The relative errors are computed with respect to the FE analysis results.

Before calculating the reduced system matrices of the components for the computed optimum design values, first the library is checked for similar component designs. These designs are sought with a relative error tolerance of 10^{-3}. The relative error is calculated with respect to the investigated optimum design.

6.1.4 Results and Discussions

The results of the first 5 executions are shown in Figure 6.2. These are obtained by utilizing the NN surrogate modeling approach within the SBO method. In the figure, the results of each execution are represented with a different marker type. The figures show the calculated final weights and the values assigned to the design variables giving these weights. The inner ring, the outer ring and the strut thicknesses of the components and the corresponding design values are presented in separate figures for making the interpretation of the results clear.

All the solutions of the first 5 executions are feasible. The average number of iterations in the SBO method is 235. The best configuration leading to the maximum reduction in the total mass is: $\mathbf{t}_{\text{inner}} = [4.1, 2.9, 2.3, 2.4, 3.7, 4.5]$, $\mathbf{t}_{\text{strut}} = [1, 1, 2.8, 1, 1, 1]$, $\mathbf{t}_{\text{outer}} = [1, 1, 1, 1, 1, 1]$. The total weight of the structure for this configuration is 0.287 kg which is 42% less than the initial weight. The values, $\mathbf{t}_{\text{inner}} = [3.4, 5, 2.8, 1.8, 3.7, 5]$, $\mathbf{t}_{\text{strut}} = [1, 1, 3, 1, 1, 1]$, $\mathbf{t}_{\text{outer}} = [1, 1, 1, 1, 1, 1]$, lead to a design which has the maximum weight, 0.2985 kg, among the computed designs. As observed from Figure 6.2, the strut and the outer ring thicknesses have almost the same values after each execution. However, there is a variation in the inner ring thickness values of the components.

The results obtained when the Kriging surrogate modeling approach utilized within the SBO method are shown in Figure 6.3. All the solutions are feasible. The average number of required iterations in the SBO method is 86. The best configuration leading to the maximum reduction in the initial weight is: $\mathbf{t}_{\text{inner}} = [3.1, 2.1, 3.4, 2.5, 3.1, 2.5]$, $\mathbf{t}_{\text{strut}} = [1, 1, 4.4, 1, 1, 1]$, $\mathbf{t}_{\text{outer}} = [1, 1, 1, 1, 1, 1]$ which leads to 39% reduction. The total weight of the structure is 0.3025 kg for these design values. The maximum computed weight, 0.3759 kg, corresponds to the configuration, $\mathbf{t}_{\text{inner}} = [2.2, 5, 5, 5, 5, 1]$, $\mathbf{t}_{\text{strut}} = [1, 1, 4.6, 1, 1, 1]$, $\mathbf{t}_{\text{outer}} = [1, 1, 5, 1, 1, 1]$.

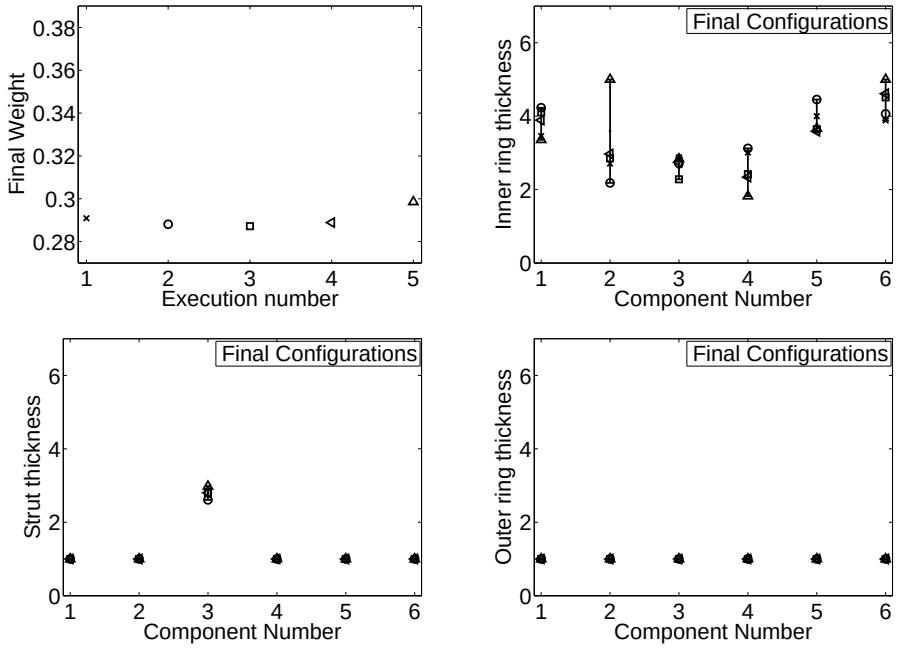


Figure 6.2: Results of the first 5 executions. The NN meta-modeling approach is used within the SBO method. For each solution, a different marker type is used.

The solutions obtained in the last 5 executions are very different from each other. The calculated weights are not gathered around a particular value as opposed to those obtained when the NN approach is utilized in the SBO method. Accordingly, the interval of the computed design values is large. Moreover, the weights are all larger than the ones computed in the first 5 executions. On the other hand, the average number of iterations is considerably less than the ones required in the SBO method with NN.

The results show that, in this example, the solutions of the SBO method with NN are more consistent and promising. On the other hand, the computational efficiency of the SBO method with Kriging is a lot better.

When the number of the sample points is small and the corresponding responses are evenly distributed around a mean with low standard deviation, a NN model may fail to capture the response accurately. This is because the regularization method (see Section 4.4.3 for details) used for increasing the generalization capability of an NN model may filter out the valuable information. Moreover, if there are any outliers, they are neglected. As a result, the generated NN model cannot capture the approximated response even in the neighborhood of the data and gives very poor predictions. Extending the data set during optimization and accordingly training the NN model with more data increases the quality of the predictions of the surrogate model. Consequently, the chance of capturing a global and a robust solution gets

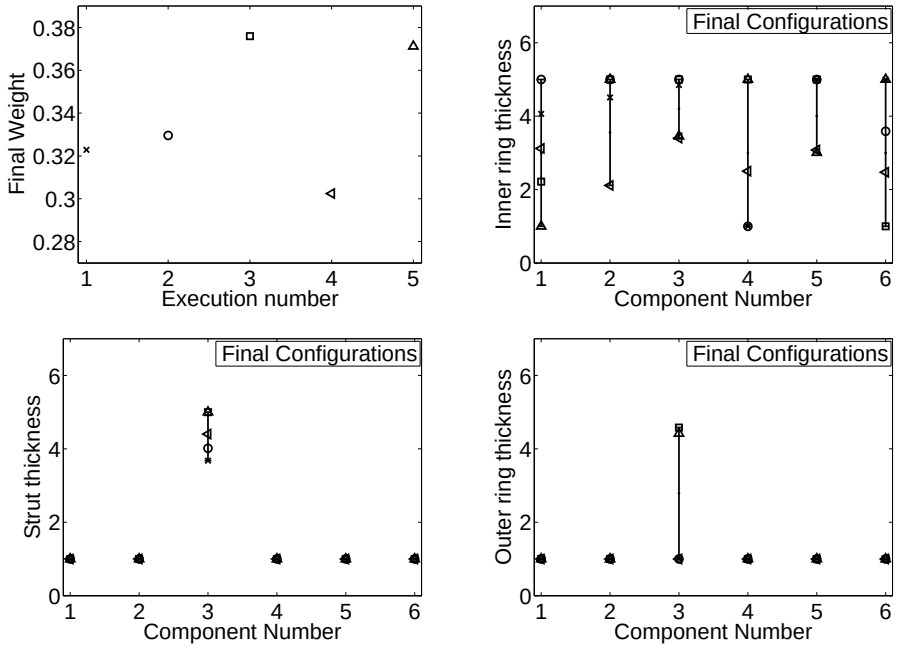


Figure 6.3: Results of the last 5 executions. The Kriging meta-modeling approach is used within the SBO method. For each solution, a different marker type is used.

higher with increasing data points.

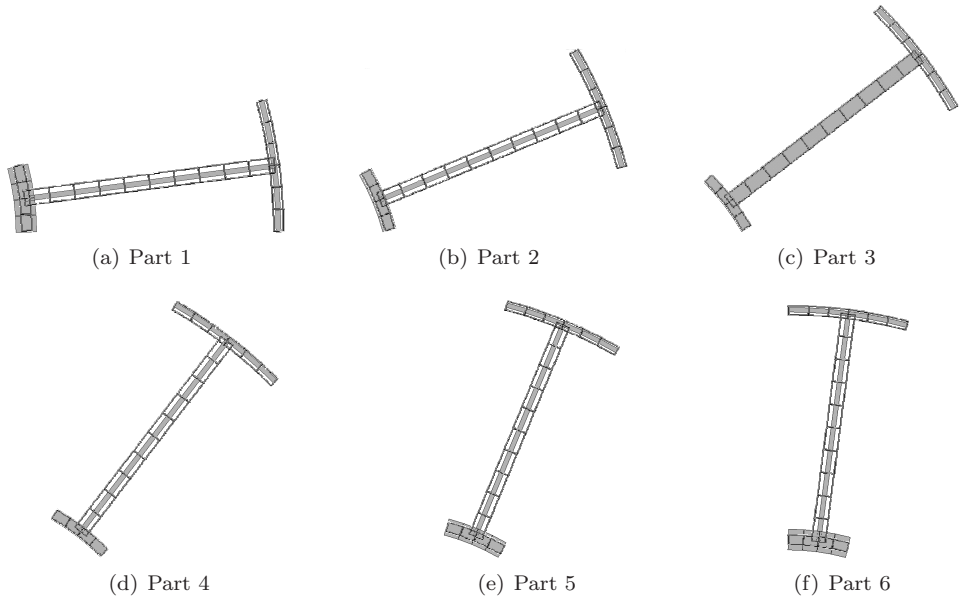
The regression constant (see Section 4.3.2 for details) used in the main diagonal of the correlation matrix does not have serious effects on the quality of the Kriging model compared to the regularization method used in the NN approximation. It becomes effective only when some of the data points start to accumulate in a region of the design space. The regression constant can be realized as a term utilized for preventing ill-conditioned correlation matrix. The data set and the values in the neighborhood of the data are approximated with high accuracy by Kriging. If a design, computed at the optimization step, is in the vicinity of one of the designs in the DOE set, there is a big chance that the FE model confirms the results of the Kriging model for this design. Hence, the SBO method is terminated with this solution. It is important to note that the predictions of the Kriging model may not be accurate for the design values which are not in the neighborhood of the available data points. If a better feasible solution lies in the domain including these design values, the SBO method with Kriging may never reach the solution due to the early termination.

As seen in Figures 6.2 and 6.3, none of the executions of the SBO method with either NN or Kriging gives the same design values. This is most likely due to the parameter estimation step of the surrogate modeling approaches (see Chapter 4). The parameters of an NN model are calculated by solving a nonlinear unconstrained optimization problem defined in Equation (4.22). This problem is solved by a

gradient-based algorithm. The initial point for the algorithm is calculated based on the Nguyen-Widrow initialization rule [103] which uses a random number generator. Therefore, the generated initial point can be different for each call of the initialization rule. Accordingly, an NN model trained on a particular data set may not get the same model parameters when it is trained again on the same data set. The above discussion also applies for Kriging. The parameters of a Kriging model are obtained by maximizing the concentrated ln-likelihood function given in Equation (4.18). This is a highly nonlinear function and finding the global solution of the function is never guaranteed even when a global search scheme is utilized. Hence, two Kriging models trained on the same data can also be different.

To summarize, there is a high possibility of not generating the same meta-models in each execution of the SBO method. Consequently, computing different design values is an expected situation. Another reason for having different results in each execution can be due to the employed search technique, the MLHO scheme. Computing a local optimum instead of the global one at an iteration of the SBO method may change the results of the next iteration. This will have consequences on the final solution.

The initial design and the best computed design are illustrated in Figure 6.4. The corresponding 5th mode shapes are as shown in Figure 6.5.



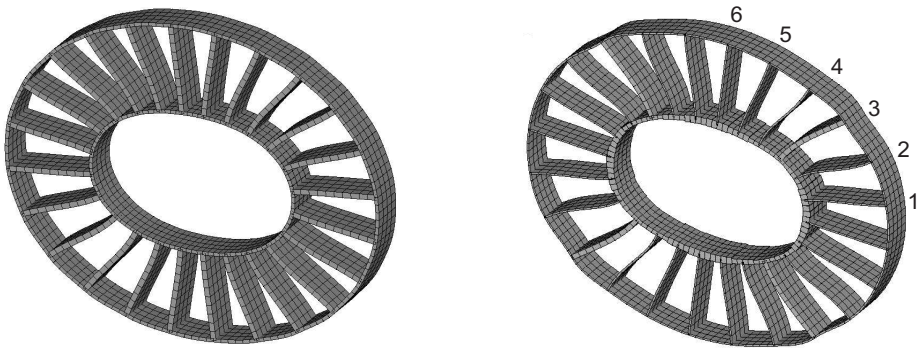
(a) Part 1 (b) Part 2 (c) Part 3

(d) Part 4 (e) Part 5 (f) Part 6

Figure 6.4: The initial and the final designs of the components where the gray color corresponds to the final design. The shown final design is obtained with the NN meta-modeling approach used within the SBO method.

Although the calculated design values are not the same, they point out a similar final design. The strut and the outer ring thicknesses have their minimum permissible

values excluding the strut thickness of the 3rd substructure. This is because these variables have a primary effect on the total mass. The objective of minimizing the total mass is achieved by assigning small values to the thicknesses of the outer ring and the strut of the components. Increasing the 5th natural frequency while preserving its mode shape is assured by modifying the inner ring thicknesses and the strut thickness of the 3rd component. The inner ring thicknesses have a dominant effect on the stiffness of the structure whereas their contribution to the total mass is relatively minor.



(a) Initial design (b) Final design

Figure 6.5: The initial and the final designs and the corresponding 5th mode shapes.

The DOE set of the complete structure is defined using the available designs of one component. Since the reduced system matrices of each component configuration are already saved in the library, it is possible to generate many different structure designs and accordingly to gather their reduced matrices without performing any extra condensation. Generating structure designs out of the available component configurations is a nice feature, especially for applications which rely on expensive analysis. On the other hand, the total number of the iterations to get an optimum solution is strongly related to the sampling strategy as well as the number of the data points. Therefore, the space-filling quality of such a sampling strategy needs to be studied further.

Another advantage of using libraries is: If a component design and the corresponding reduced matrices are already in the library, similar designs do not have to be computed again. In the test problem, on average 415 and 221 *structure configurations* are created at the end of the optimization procedure including the designs in $\mathbf{D}_{\mathrm{T}}$ when NN and Kriging are used within the SBO method, respectively. The libraries of SBO with NN and SBO with Kriging involve subsequently, 1048 and 462 *component configurations*. If no libraries had been used, 2490 and 1326 component analysis would have had to be performed. As observed from these results, using component libraries has a significant role on the computational efficiency of the SBO method.

6.2 Updating the Craig-Bampton Transformation Matrix for Efficient Structural Optimization

The objective of the second test problem is to analyze the influence of the proposed reanalysis approach on the SBO strategy. The strategy is executed both with and without the reanalysis approach. Its accuracy and computational efficiency is reviewed based on the obtained results. The NN approximations are preferred for surrogate modeling regarding the observations of the previous application.

In the test problem, the amplitude of the displacement responses is minimized by modifying the structure design under the constraints of preserving the frequency and the corresponding mode shape of the initial design. Potential application areas of such problem definitions can be for structures where large response amplifications are known to contribute to material damage which eventually led to failure [109].

6.2.1 Problem Definition

The same structure used in Section 2.4 is selected for the second test problem. The details about the design of the structure and gathering its reduced system matrices for the dynamic analysis can be found in that section.

In this problem, there is an harmonic force acting on the structure. The applied load has an amplitude of 100 kN and is in the y-direction. It is applied on the top interface node of the 4th component C_4 and the 5th component C_5 as shown in Figure 6.6. For the harmonic response analysis, structural damping with an energy dissipation of 3% is assumed which is imposed directly on the reduced stiffness matrix of the structure.

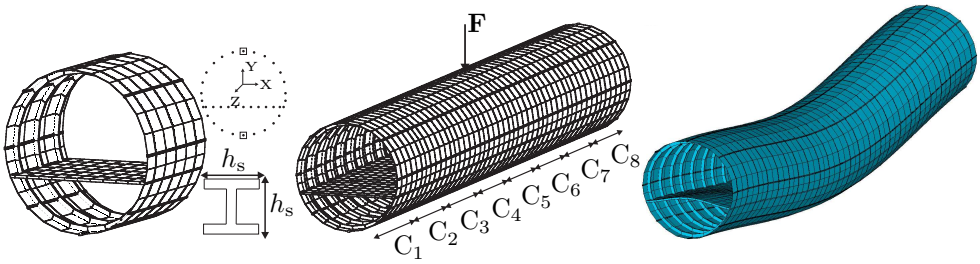


Figure 6.6: Test problem. (Left) Component model, (Middle) Selected structure under applied force, (Right) First bending mode of the initial design.

The cross-section width and height of the stiffeners, $\mathbf{h} = \{h_{s_i} \mid i = 1, 2, \ldots, 8\}$, in the components are the design variables. All the stiffeners of a component are assumed to have the same design values. Hence, there are 8 design variables in the structure which have a value of all 0.05 m in the initial design.

For the harmonic response analysis, focus is on the frequency range of $10-30$ Hz. This interval involves the first bending frequency of the initial design. Figure 6.6 shows

the mode shape of this frequency. The objective is to reduce the amplitude of the displacement response in this frequency range, thereby decreasing the displacement response of the structure for the first bending mode. The nodes that lie on the top and the bottom interface of the components are selected to prescribe the objective function. Figure 6.6 illustrates the nodes corresponding to the interface of a component. The selected nodes are identified with squares around them. The displacement magnitudes in the y-direction are computed for these nodes in the frequency range of $10-30$ Hz and then summed up. The response curve that represents the "frequency-displacement magnitude" relationship of the initial design is plotted in Figure 6.7. The results displayed in the figure are obtained by the full FE analysis performed in ANSYS.

The objective function of the problem is defined as minimizing the total area beneath the response curve, shown in Figure 6.7. The area beneath the response curve is 0.95 m.Hz for the initial design.

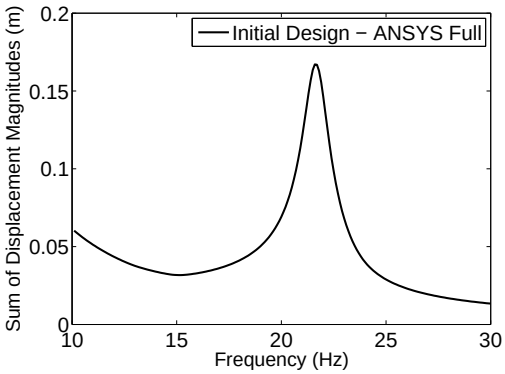


Figure 6.7: Frequency-Displacement Magnitude curve of the initial design.

The constraints of the problem are as follows:

- Keeping the first bending frequency around 22 Hz. This constraint is defined as

$$22 - \epsilon \leq f_7(\mathbf{h}) \leq 22 + \epsilon$$

  where $\epsilon = 0.02$.

- Keeping the total final mass of the stiffeners less than the total initial mass of the stiffeners. This is given as,

$$\sum_{i=1}^{8} [\rho \, \mathrm{V}_i(\mathrm{h}_{\mathrm{s}_i})] \leq 23$$

  where $\mathrm{V}_i(\mathrm{h}_{\mathrm{s}_i})$ is the total volume of the stiffeners in component C_i and ρ is the density of the stiffeners.

- Preserving the mode shape of the first bending frequency. This is assured by the MAC criterion (see Equation (2.39) for MAC formula).

$$\mathrm{MAC}_7(\mathbf{h}) \geq 0.9.$$

- Having a symmetric final configuration[1]. This constraint is prescribed by forcing the design variables of the component pairs; C_1-C_8, C_2-C_7, C_3-C_6, C_4-C_5 to have similar values. This is imposed by:

$$\begin{aligned} &\mathrm{h}_{\mathrm{s}_j} - \mathrm{h}_{\mathrm{s}_{(9-j)}} \leq 10^{-4} \\ &\mathrm{h}_{\mathrm{s}_{(9-j)}} - \mathrm{h}_{\mathrm{s}_j} \leq 10^{-4}, \quad j = 1, 2, \ldots, 4. \end{aligned}$$

- The upper and the lower bounds for the design variables are selected as

$$0.01 \leq \mathrm{h}_{\mathrm{s}_i} \leq 0.1, \quad i = 1, 2, \ldots, 8.$$

Casting above constraints in the formulation of the optimization problem gives,

$$\min_{\mathbf{h}} \left[\mathrm{A}(\mathbf{h})\right] \text{ subject to } \begin{cases} 22 - \epsilon \leq f_7(\mathbf{h}) \leq 22 + \epsilon \\ \sum_{i=1}^{8} [\rho \, \mathrm{V}_i(\mathrm{h}_{\mathrm{s}_i})] \leq 23 \\ \mathrm{MAC}_7(\mathbf{h}) \geq 0.9 \\ \mathrm{h}_{\mathrm{s}_j} - \mathrm{h}_{\mathrm{s}_{(9-j)}} \leq 10^{-4} \\ \mathrm{h}_{\mathrm{s}_{(9-j)}} - \mathrm{h}_{\mathrm{s}_j} \leq 10^{-4}, \quad j = 1, 2, \ldots, 4 \\ 0.01 \leq \mathrm{h}_{\mathrm{s}_i} \leq 0.1, \quad i = 1, 2, \ldots, 8 \end{cases} \tag{6.2}$$

where $\mathrm{A}(\mathbf{h})$ stands for the total area beneath the "frequency-displacement magnitude" curve corresponding to the frequency range of $10 - 30$ Hz.

6.2.2 Surrogate Modeling

In the optimization problem, 3 surrogate models are used. These surrogates stand for $\mathrm{A}(\mathbf{h})$, $f_7(\mathbf{h})$ and $\mathrm{MAC}_7(\mathbf{h})$ given in Equation (6.2).

The DOE set $\mathbf{D}_\mathrm{T}$ of the whole structure has 81 designs which is generated using the LHS scheme. Each design in $\mathbf{D}_\mathrm{T}$ defines a new structure configuration.

At the analysis step of the SBO method, the transformation matrix of each modified component is computed using one of the following methods:

- **Exact**: The fixed interface normal modes and the constraint modes of the Craig-Bampton (CB) transformation matrix are computed by exact analysis methods all over again.

[1] In practice, this choice may not be logical and efficient. In this academic example, by such a choice, the strategy is aimed to be tested with more design variables and constraints.

- **ECB+CA**: The initial fixed interface normal mode set is extended using the Enriched Craig-Bampton (ECB) method.

  The constraint modes are approximated by the Combined Approximations (CA) approach. The minimum number of the basis vectors are set to 3. The accuracy of the modes are verified by Equation (2.38). If the accuracy is not satisfactory, the set of the CA basis vectors is extended with a new vector and the reanalysis step is repeated.

- **ECB+CA Automated**: The initial fixed interface normal mode set is extended using the Enriched Craig-Bampton (ECB) method.

  For the calculation of the constraint modes, the automated update scheme defined in Section 2.3.2 is used. The minimum number of the basis vectors are set to 3 in the CA approach. The number of **FL**oating point **OP**erations (FLOPs) is computed to control the switch between the exact and the approximate analysis of the constraint modes. The equations given in Table 2.1 are used for estimating the number of FLOPs. The CA approach is employed only when it requires less number of FLOPs than the exact analysis. The accuracy of the approximated modes are verified by Equation (2.38). If the accuracy is not satisfactory and the computational efficiency of the CA approach is still better than the exact analysis, the set of the CA basis vectors is extended with a new vector and the reanalysis step is repeated. Otherwise, the constraint modes are computed with the exact analysis.

After the transformation matrix of a component is calculated using one of the above methods, condensation of the component matrices are performed.

The responses, $\mathrm{A}(\mathbf{h})$, $f_7(\mathbf{h})$ and $\mathrm{MAC}_7(\mathbf{h})$, of the structure for each configuration in $\mathbf{D}_\mathrm{T}$ are calculated using the assembled reduced component system matrices and the training data sets are gathered for meta-modeling.

3 separate libraries are used for storing the transformation, the reduced stiffness and the reduced mass matrices of each new component design. The first library is for component C_1, the second one is for components $\mathrm{C}_2, \mathrm{C}_3, \ldots, \mathrm{C}_7$ and the third one is for C_8. The constraint and the fixed interface normal modes of components $\mathrm{C}_2, \mathrm{C}_3, \ldots, \mathrm{C}_7$ are different than the modes of components C_1 and C_8. Therefore, the reduced system matrices of these components are not equal in the local coordinates for the same design values. Hence, the component library of $\mathrm{C}_2, \mathrm{C}_3, \ldots, \mathrm{C}_7$ is different than the libraries of C_1 and C_8. Although 2 separate libraries are defined for the components C_1 and C_8, it is also possible to utilize a single library. Since the system matrices of these components are the same in the local coordinates, multiplying the reduced system matrices of one of the components with a proper rotation matrix gives the reduced system matrices of the other component in the global coordinates. Another alternative would be utilizing a single library for all the substructures where the transformation matrices are computed by assuming that the d.o.f. on the boundaries of every component belong to an interface. This choice may have consequences on the accuracy of the analyses. Additionally, the size of the reduced system matrices of the structure is much bigger. On the other hand, it can be computationally more

efficient because of avoiding extra calculations when the substructure designs of C_1 and/or C_8 are the same as one of the components $C_2, C_3, \ldots, C_7$.

6.2.3 Optimization and Validation

The following cases are considered in this test problem.

- **Case 1:** The optimization problem is solved twice. First, the **Exact** approach is used for the calculation of the transformation matrices during the analysis step of the SBO method. In the second solution, instead of the **Exact** approach, the **ECB+CA Automated** approach is used. The performance of the SBO method is evaluated regarding the accuracy of the results and the computation time. In short, in Case 1, the computational efficiency and the accuracy of the proposed reanalysis methods are tested.

  The NN meta-modeling approach is employed for defining the surrogates. In the NN model, 25 hidden layer neurons are employed.

  It is important to emphasize that the number of FLOPs for the exact analysis of the constraint modes is smaller than that of the CA approach in the selected structure. Accordingly, in the **CA Automated** approach, the constraint modes are always computed by the exact analysis.

- **Case 2:** As mentioned in Case 1, the constraint modes are always computed by the exact methods in the **CA Automated** approach. In order to examine the accuracy of the CA method, the **ECB+CA** approach is used in the analysis step of the SBO method. The final design configuration and the corresponding analysis results are compared with the solutions found in Case 1.

  The NN meta-modeling approach is employed for defining the surrogates. In the NN model, 25 hidden layer neurons are employed.

- **Case 3:** The solutions of the SBO method are found to be sensitive to the selected surrogate modeling approach in the first test problem studied in Section 6.1. To explore the performance of the SBO method on the current test problem, the problem is solved once again using the **Exact** approach but this time the Kriging meta-modeling approach is used. In the Kriging approximation, the MLHO scheme is used to find the model parameters $\boldsymbol{\theta}$. The entries p_j, $j = 1, 2, \ldots, 8$ of the exponent vector $\mathbf{p}$ are selected as 2.

In all the test cases, the MLHO optimization scheme is used in SBO for finding the global optimum. During the calculation of the approximate residual constraint modes, the tolerance value for the maximum norm of $\boldsymbol{\epsilon}_{\mathrm{CA}}$ (see Equation (2.38) is set to 0.001.

The search for optimum is repeated until the relative errors between the responses of the FE model and that of the surrogates are smaller than 0.005 for the computed optimum design values. The relative error is computed with respect to the FE analysis results.

Before calculating the reduced system matrices of the components for the optimum design values, first the libraries are checked for similar component designs. These designs are sought with a relative error tolerance of 10^{-3}. The relative error is calculated with respect to the investigated optimum design.

6.2.4 Results and Discussions

Case 1

The results of Case 1 are summarized in Table 6.1.

Table 6.1: Summary of Case 1 results.

	Exact & NN	**ECB+CA Automated**
Final Design (m)	[0.01, 0.01, 0.058, 0.1, 0.1, 0.058, 0.01, 0.01]	[0.01, 0.01, 0.063, 0.1, 0.1, 0.063, 0.01, 0.01]
Final Total Area (m.Hz)	0.8238	0.8161
Final Mass (kg)	21.7	22.5
Total # of iterations	4	4
Computation time	5h06min	3h34min

The "frequency-displacement magnitude" curves that correspond to the final configurations are shown in Figure 6.8. To validate the results, the response of the

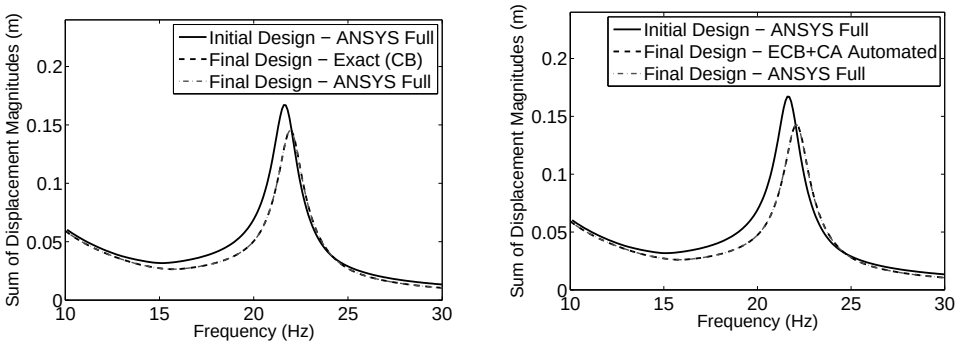


Figure 6.8: Results of Case 1. (Left) The CB transformation matrices are computed by the **Exact** approach in the SBO method, (Right) The CB transformation matrices are computed by the **ECB+CA Automated** approach in the SBO method.

structure is calculated in ANSYS using the full FE analysis for the final design values. These solutions are also presented in Figure 6.8.

Both of the final configurations are feasible. These configurations have almost the same design values. The optimal configuration is stiffest in the middle while the

stiffness decreases towards the free ends of the structure. The total area beneath the "frequency-displacement magnitude" curve is reduced by almost 14% in both **Exact** & NN and **ECB+CA Automated**.

The total required time for the optimization process decreases around 30% when the **ECB+CA Automated** approach is utilized in the SBO method.

As observed from the results, the total number of the iterations required in the SBO method are very low even though the NN approximation is used for defining the surrogates.

The accuracy and the computational efficiency of the SBO method with **ECB+CA Automated** approach is very satisfactory for the selected problem.

Case 2

The results of the SBO method with the **ECB+CA** approach are summarized in Table 6.2.

Table 6.2: Summary of Case 2 results.

	ECB+CA
Final Design (m)	$[0.01, 0.01, 0.059, 0.1, 0.1, 0.059, 0.01, 0.01]$
Final Total Area (m.Hz)	0.8215
Final Mass (kg)	21.9
Total # of iterations	6

The "frequency-displacement magnitude" curve that corresponds to the final configuration is plotted in Figure 6.9.

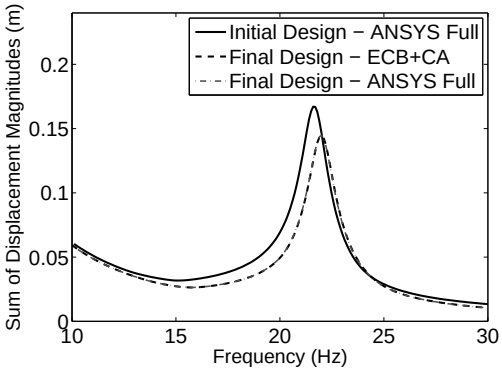


Figure 6.9: Results of Case 2. The CB transformation matrices are computed by the **ECB+CA** approach in the SBO method.

The final design is very similar to those obtained in Case 1 and it fulfills all the constraints. The total area beneath the "frequency-displacement magnitude" curve is reduced by almost 14%.

As seen in Figure 6.9, the accuracy of **ECB+CA** is satisfactory compared to the full FE analysis results for the final design values.

Case 3

The results of the SBO method with Kriging meta-models are summarized in Table 6.3.

Table 6.3: Summary of Case 3 results.

	Exact & Kriging
Final Design (m)	$[0.01, 0.01, 0.061, 0.1, 0.1, 0.061, 0.01, 0.01]$
Final Total Area (m.Hz)	0.8197
Final Mass (kg)	22.1
Total # of iterations	2

The "frequency-displacement magnitude" curve that corresponds to the final configuration is plotted in Figure 6.10.

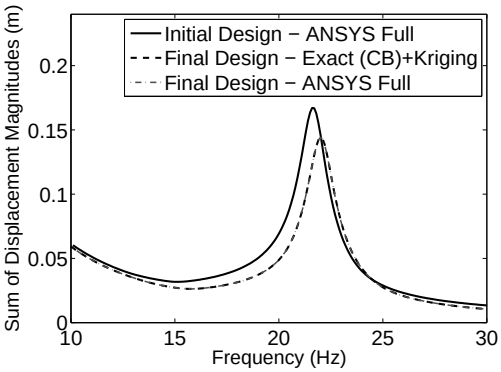


Figure 6.10: Results of Case 3. The CB transformation matrices are computed by the **Exact** approach. Kriging surrogate modeling approach is used in the SBO method.

The final configuration is feasible and it leads to the same conclusion for the optimal design as in Case 1 and Case 2. The reduction in the total area beneath the "frequency-displacement magnitude" curve is 14%.

The solutions of the SBO method with NN have already been presented in Case 1. As realized from these results, variations in the final design values are very low and the total number of the iterations in the SBO method are small for each employed surrogate modeling approach.

Consequently, the conclusions of the first test problem on the surrogate modeling approaches are certainly not valid for the current test problem. This may have several reasons: Firstly, the number of the design variables are a lot fewer than the previous problem. It is more likely to obtain a space-filling design with less design variables when the number of the training data is selected based on *ten times the number of the design variables* rule. Another reason can be the sampling scheme. In the first problem, LHS is only utilized to generate the component configurations. The structure designs are defined out of these configurations. On the other hand, in the current problem, the structure designs are defined by the LHS scheme. The total number of the component configurations is important to maintain a certain diversity among the structure designs and accordingly to get a space-filling design.

6.3 Summary and Conclusions

The proposed optimization strategy was demonstrated on two theoretical test problems in which the selected structures have repeating patterns in their geometries. In the first problem, the modeling and the analysis features of the Craig-Bampton (CB) method were exhibited. These features were exploited during the generation of the Design of Experiment (DOE) set of the whole structure and the computation of the reduced system matrices. The first problem was also used to analyze the performance of the strategy where the Neural Network (NN) and the Kriging meta-models were used as surrogates. The main objective of the second test problem was to examine the efficiency of the SBO strategy when it is executed with the proposed reanalysis methods.

It was shown that when a structure can be decomposed into repeating components, it is possible and efficient to model one component and utilize its parameterized FE model for the rest of the same components using the CB method.

Each component model is independent from each other in the CB method. This means, every component model has its own design variables. This attribute is very favorable during the selection of the DOE set of the complete structure. After generating various designs for each component and calculating the corresponding reduced matrices, it is possible to define different structure configurations out of these available component designs. The reduced system matrices of the whole structure are gathered by the assembly of the already computed reduced component matrices. Although this is a very promising and computationally efficient strategy, the validity and the applicability of the strategy still needs to be clarified.

The reduced matrices of each computed component configuration are kept in the libraries. When similar components need to be analyzed again, the corresponding reduced matrices are called from the library. Hence, unnecessary static and

dynamic analysis required for the calculation of the CB transformation matrix are prevented. Libraries especially become important at the later stages of the optimization procedure when some of the component designs are assigned with their final configurations.

It is hard to compare the influence of using the Kriging and the NN meta-modeling approaches on the performance of the SBO strategy. The influence is strongly application dependent. The large number of the iterations due to using the NN models and the non-robust solutions due to using the Kriging models are potential pitfalls that may hinder the efficiency of the strategy for certain applications.

The Enriched CB (ECB) method and the Combined Approximations (CA) approach are used for approximate computation of the fixed interface and the constraint modes of the CB transformation matrix, respectively. The automated computation of the constraint modes is a very attractive feature in which the exact or the approximate calculation of the modes are automatically decided using the number of FLOPs. The computational efficiency of the SBO method improves when the proposed reanalysis methods are used for the calculation of the CB transformation matrix at the analysis step. Moreover, the accuracy of the provided solutions are preserved.

7

Conclusions and Recommendations

The goal of this research was to develop a strategy for the optimization of the dynamic behavior of structures. The strategy was required to be robust, accurate and able to provide a solution which is as close to the global optimum as possible. Moreover, it should be suitable for optimizing complex structures whose analyses are based on computationally demanding Finite Element (FE) simulations.

In order to fulfill these demands, several research subjects have been identified. *Reduction* and *reanalysis* methods have been studied for speeding up the structural analyses during optimization. *Numerical Optimization* methods have been examined to find out an effective strategy that may lead to the global optimum. It was realized that any effort on finding the global optimum requires a tremendous number of function evaluations, i.e. FE analyses. Therefore, FE models were decided to be replaced with their surrogates whose evaluations are very fast, once generated. This necessitated a thorough understanding of the *surrogate modeling approach* which was also one of the research topics in the thesis. Finally, based on the knowledge gained, an optimization strategy was proposed.

This chapter presents the conclusions drawn from these studies and the recommendations for further research.

7.1 Reduction and Reanalysis Methods

One of the main difficulties in structural optimization is the analysis time. Therefore it is essential to consider efficiency as well as accuracy in dynamic analysis when using it in combination with an optimization method.

The Craig-Bampton (CB) method was found to be very advantageous to utilize in the optimization algorithm. The main advantages are observed to be the following.

- It enables parallel processing which can be thought in two different ways: Distribution of the analysis over different computers or to different groups of people who specialize in different categories.
- It reduces the total d.o.f. to a great extent which then leads to a significant reduction in the computation time. Meanwhile, the accuracy of the solutions is preserved in the low frequency range.
- It is possible to efficiently test the influence of different substructure configurations on the global dynamic behavior.
- One parametric model can be used for similar substructures in the analysis of a complete structure.

One drawback of the CB method is that the assembly of the reduced component models are based on the compatibility of the interface nodes. It is recommended to study suitable coupling methods which can be used for non-matching interfaces to eliminate the effort needed to have a compatible mesh on the neighboring interfaces.

It is known that when a substructure is modified, the solution of the initial model does not conform to the new system equations which causes residual forces and residual displacements to appear. This information can be used to update the solution, thereby avoiding the need to solve the complete set of new equations which include dynamic as well as static analysis. Two methods for this purpose were employed: The Enriched CB (ECB) method and a new reanalysis method.

The ECB method extends the fixed interface normal mode set by including the effects of the residual forces acting on the modified substructure. Thus, modal analysis can be avoided while preserving the accuracy to a certain extent. However, there is no proposed automated update scheme in the literature that switches the ECB method with the exact analysis when the accuracy and computational efficiency are not balanced. It is recommended to consider alternative update methods for the normal modes which enable this attribute.

For updating the constraint modes in the CB analysis, a new reanalysis method based on the Combined Approximations (CA) approach was presented. In the method, the residual displacements are approximated in a smaller space and added to the initial constraint modes. The test problem showed that while the accuracy of the method is satisfactory for a wide range of perturbations, the computational efficiency depends on the properties of the component model. Although this makes the method not applicable for every problem, an automated approach was proposed to select either the exact or the approximate analysis depending on an estimated computational cost.

7.2 Numerical Optimization

In structural optimization, the ultimate goal is to find the exact global optimum with the least number of function evaluations. Trying to achieve this goal was found to be far from trivial.

Non-Linear Programming (NLP) is the most general class of continuous optimization framework. In this research, two algorithms, namely Sequential Quadratic Programming (SQP) and Genetic Algorithms (GAs), have been analyzed in solving NLP problems.

SQP is a fast and robust gradient-based method. Although the computed optimum is exact and can be obtained with a relatively small number of function evaluations, finding the global optimum is never ensured for multi-modal problems.

GAs are stochastic derivative-free methods which explore the design space globally without using any gradient information. This makes them suitable to search for the global optimum. However, their convergence rate is very slow compared to SQP and they do not guarantee an exact optimum.

In order to increase the chance of finding the exact global optimum for multi-modal problems, two optimization schemes, the Multi-Start Local Optimization (MSLO) and the Multi-Level Hybrid Optimization (MLHO), have been considered. The numerical test problems showed that the chance of obtaining the global optimum with the MSLO scheme is correlated with the number of the randomly selected initial designs. On the other hand, the progress of the MLHO scheme is dominated by the employed GA method.

It was observed that pursuit of the global optimum requires a large number of function evaluations and unfortunately, the global optimum may still not be assured.

Because of the prohibitive number of FE analyses, direct coupling of the structural analysis methods is not recommended either with a global optimization method or with a global optimization scheme.

7.3 Surrogate Modeling

The prediction capability of a surrogate model is strongly influenced by the number of the design variables, the complexity of the approximated trend, the quality of the training data, the effectiveness of the utilized meta-model and the generalization approach.

When the investigated trend is known to be governed by a low order function or the number of the design variables are small, the polynomial models can provide satisfactory predictions. These models may also be preferred in order to identify the promising regions in the design space. Their transparency makes them favorable in certain applications.

The majority of the engineering design problems are multi-dimensional and usually limited knowledge is available on the approximated trend. In these cases the Kriging and the Neural Network (NN) approximations are superior to the polynomial models due to their flexibility in generating surrogate models. It was observed that with these approaches satisfactory models can be obtained with a limited amount of training data.

Both Kriging and NNs make use of the parametric exponential functions which enable them to approximate any nonlinear differentiable function.

The prediction capability of NNs gets better with the increasing number of the neurons used in the hidden layer. Although the number of the model parameters may exceed the number of the data, the generalization capability of the model is still assured by employing the regularization method. In order to find the model parameters, a non-linear least squares problem must be solved which may have many local minima. This is the parameter estimation step, also known as training, where an optimization algorithm is required. Generally the gradient-based methods are preferred over derivative-free stochastic methods for computational efficiency in models with a large number of parameters. Due to the random number generators involved in the initialization of the gradient-based methods, a different set of model parameters can be found for the same training data in successive computations. This reduces the robustness of the NN surrogate modeling approach, however it was found that for the demonstrated test problems the results are satisfactory.

In Kriging approximations, the nonlinearity of the model is correlated with the number of the data points and their distribution in the design space. A Kriging model can interpolate the data as well as it can filter out the fluctuations in the data if its correlation matrix is modified with a regression constant. At the parameter estimation step, a concentrated log-likelihood function needs to be maximized which is nonlinear in nature. For each evaluation of this function, the explicit estimates of the mean and the variance of the observations, as well as the determinant of the correlation matrix are required. Although it might be feasible to employ a global search scheme for finding the global Kriging parameters, different Kriging models may still be obtained each time the model is trained on the same data. It was observed in some test problems that the difference in the results can be significant.

7.4 Optimization Strategy

The studied structural analysis and numerical optimization methods were merged to define the Surrogate-Based Optimization (SBO) scheme where global approximations are utilized as surrogates. The proposed hybrid scheme consists of the CB method and the reanalysis methods for dynamic analysis, Latin Hypercube Sampling (LHS) for design of experiments, the Kriging and the NN approximations for surrogate modeling and finally the MLHO and the MSLO schemes for global optimization.

The SBO scheme is suitable for small-scale optimization problems. It does not require sequential evaluations of an FE model thanks to the employed surrogate models. Furthermore, utilizing surrogate models promotes parallelization since the training data can be generated by independent FE calculations. Parallel processing is further supported by the CB method which even allows modeling and analysis of each component independently. Utilizing the CB method further enables shorter analysis time, freedom of not evaluating the unmodified substructures during optimization, ease in modeling of structures having repeating components and effective bookkeeping of the already analyzed component configurations via libraries.

It was shown that for structures having repeating geometries, it is feasible and efficient to utilize one parametric model for all similar substructures during optimization.

Another advantage of utilizing CMS in optimization is the possibility of generating a large DOE set using the combinations of the available component configurations. Although it has not been covered in this research, tests on the validity and applicability of this feature are important to define more efficient strategies.

Results obtained from the test problems revealed that the proposed reanalysis methods improve the computational efficiency of the SBO method while keeping the accuracy of the solutions to a significant extent.

The influence of the Kriging and the NN approximations on the efficiency of the SBO method was found to be application dependent. It was observed that a large number of iterations might be required when the NN surrogates are used in the proposed optimization scheme. A potential drawback of employing the Kriging surrogates is the non-robustness of the solutions.

The proposed optimization strategy is very promising to be applied for design problems where the global dynamic behavior of complex structures is desired to be changed by local modifications.

In order to have an objective analysis of the effectiveness of the proposed SBO scheme on a real design problem, the CB and the reanalysis methods must be implemented directly in an FE software. The current implementation suffers from the fact that these computations are conducted at a higher level compared to the FE analysis.

7.5 Further Recommendations

As mentioned earlier, the efficiency of the optimization strategies is affected significantly by the number of the design variables. In order to reduce this number, reliable and computationally efficient screening methods are essential. These methods should be able to screen out the most important design variables by a minimal number of data. Further research on this subject will be very helpful and valuable in the structural optimization field.

The current version of the proposed SBO method can only be used for sizing optimization problems. This is due to the coupling procedure used in the CB method which requires compatibility of the interface nodes. Further research on the coupling methods for non-matching interfaces will make the optimization strategy suitable for shape optimization problems as well. However, reanalysis methods have to be reconsidered. Major geometrical modifications may lead to completely different mode shapes which cannot be represented accurately using the system matrices of the initial design.

The performance of different surrogate modeling approaches are application dependent and it is usually not known which method will be the most suitable for a certain analysis. It is even possible that different models identify distinct aspects of the investigated trend better than the other. Therefore, it is recommended to

develop strategies that consider multiple surrogate models simultaneously, evaluate them and generate a combined model whose predictions are much better than any of the individual meta-models. Hopefully this will eliminate concerns about *which surrogate modeling approach* to use.

A

Representation of Rigid Body Modes

Let $\mathbf{d}_{\mathrm{R}}^{(j)}$ be a vector of the jth rigid body displacement of a substructure. It can be written as a linear combination of the constraint modes such that

$$\mathbf{d}_{\mathrm{R}}^{(j)} = \begin{bmatrix} \boldsymbol{\Psi}_{\mathrm{ib}}^{0} \\ \mathbf{I}_{\mathrm{bb}} \end{bmatrix} \mathbf{y}^{(j)} \quad j = 1, 2, \ldots, 6 \tag{A.1}$$

where $\{\mathbf{y}^{(j)}\}^{\mathrm{T}} = \{y_1^{(j)}, y_2^{(j)}, \ldots, y_{N_{\mathrm{s}}}^{(j)}\}$ is a vector of coefficients.

If any modification in a substructure does not change the mesh of the Finite Element (FE) model, the rigid body displacements of the initial model also hold for the modified one. Accordingly,

$$\begin{bmatrix} \mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}} & \mathbf{K}_{\mathrm{ib}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}} \\ \mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}} & \mathbf{K}_{\mathrm{bb}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bb}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_{\mathrm{ib}}^{0} \\ \mathbf{I}_{\mathrm{bb}} \end{bmatrix} \mathbf{y}^{(j)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{A.2}$$

Equation (A.2) gives

$$\begin{aligned} (\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \mathbf{K}_{\mathrm{ib}})\mathbf{y}^{(j)} + (\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}})\mathbf{y}^{(j)} &= \mathbf{0} \\ \left[(\mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}})\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + (\mathbf{K}_{\mathrm{bb}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bb}})\right]\mathbf{y}^{(j)} &= \mathbf{0} \end{aligned} \tag{A.3}$$

for which the first term of the first equation cancels out due to $\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \mathbf{K}_{\mathrm{ib}} = \mathbf{0}$. Hence, the equalities

$$\begin{aligned} (\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}})\mathbf{y}^{(j)} &= \mathbf{0} \\ \left[(\mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}})\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + (\mathbf{K}_{\mathrm{bb}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bb}})\right]\mathbf{y}^{(j)} &= \mathbf{0} \end{aligned} \tag{A.4}$$

are satisfied for every modified component.

Now suppose that the constraint modes of a component are recomputed for the new design. If these modes are able to represent the rigid body displacements, the equality

$$\begin{bmatrix} \mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}} & \mathbf{K}_{\mathrm{ib}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}} \\ \mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}} & \mathbf{K}_{\mathrm{bb}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bb}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} \\ \mathbf{I}_{\mathrm{bb}} \end{bmatrix} \hat{\mathbf{y}}^{(j)} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{A.5}$$

has to hold where $\{\hat{\mathbf{y}}^{(j)}\}^{\mathrm{T}} = \{\hat{y}_1^{(j)}, \hat{y}_2^{(j)}, \ldots, \hat{y}_{N_\mathrm{s}}^{(j)}\}$ is a vector of coefficients for the jth rigid body mode. Equation (A.5) gives

$$\begin{aligned}
&\left[\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \mathbf{K}_{\mathrm{ib}}\right]\hat{\mathbf{y}}^{(j)} + \left[\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}}\right]\hat{\mathbf{y}}^{(j)} + \left[(\mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}})\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\right]\hat{\mathbf{y}}^{(j)} = \mathbf{0} \\
&\left[(\mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}})\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + (\mathbf{K}_{\mathrm{bb}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bb}})\right]\hat{\mathbf{y}}^{(j)} + \left[(\mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}})\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\right]\hat{\mathbf{y}}^{(j)} = \mathbf{0}.
\end{aligned} \tag{A.6}$$

Since the geometry has not changed, the rigid body displacements of the modified component can still be represented by the same combination of unit displacements on the interface. Hence, when $\hat{\mathbf{y}}^{(j)}$ is selected as equal to $\mathbf{y}^{(j)}$ and Equation (A.4) is taken into account, Equation (A.6) becomes

$$\begin{aligned}
&\left[(\mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}})\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\right]\mathbf{y}^{(j)} = \mathbf{0} \\
&\left[(\mathbf{K}_{\mathrm{bi}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{bi}})\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\right]\mathbf{y}^{(j)} = \mathbf{0}
\end{aligned} \tag{A.7}$$

If the equality

$$\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\mathbf{y}^{(j)} = \mathbf{0} \tag{A.8}$$

is proven to be fulfilled by an approximation method utilized for updating the constraint modes, the representation of rigid body modes is assured.

A.1 Binomial Series Expansion

In the Binomial Series (BS) approximation,

$$\begin{aligned}
\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} &\approx (\mathbf{I} - \mathbf{B} + \mathbf{B}^2 - \ldots)\boldsymbol{\Delta}\mathbf{r}_1 = \boldsymbol{\Delta}\mathbf{r}_1 + \boldsymbol{\Delta}\mathbf{r}_2 + \ldots \\
&\approx \left[-\mathbf{K}_{\mathrm{ii}}^{-1}(\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}})\right] + \mathbf{B}\left[\mathbf{K}_{\mathrm{ii}}^{-1}(\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}\boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}})\right] + \ldots
\end{aligned} \tag{A.9}$$

where $\mathbf{B} = \mathbf{K}_{\mathrm{ii}}^{-1}\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}$. When both sides of Equation (A.9) is multiplied by the coefficient vector $\mathbf{y}^{(j)}$ and Equation (A.4) is taken into account,

$$\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\mathbf{y}^{(j)} = \mathbf{0}.$$

Consequently, Equation (A.8) is satisfied for the binomial series approximation. That is, rigid body modes are always represented by BS if the FE mesh remains the same after the modifications.

A.2 Combined Approximations Approach

In the Combined Approximations (CA) approach, the column entries of each $\boldsymbol{\Delta}\mathbf{r}_k$, $k = 1, 2, \ldots, N_\mathrm{s}$ are conditioned (see Equation (2.33)). Therefore $\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}}\mathbf{y}^{(j)} = \mathbf{0}$ may not be satisfied when the number of the selected binomial series terms, k, is not sufficient.

The rigid body displacements are guaranteed when $\boldsymbol{\epsilon}_{\mathrm{CA}}$ in Equation (2.38) is driven to zero, i.e.

$$\boldsymbol{\epsilon}_{\mathrm{CA}} = \left(\left[(\mathbf{I} + \mathbf{K}_{\mathrm{ii}}^{-1} \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}) \boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} \right] + \left[\mathbf{K}_{\mathrm{ii}}^{-1} (\boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}} \boldsymbol{\Psi}_{\mathrm{ib}}^{0} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ib}}) \right] \right) \rightarrow \mathbf{0}. \tag{A.10}$$

When both sides of Equation (A.10) are multiplied by the coefficient vector $\mathbf{y}^{(j)}$ and Equation (A.4) is taken into account, the equation becomes

$$\boldsymbol{\epsilon}_{\mathrm{CA}}\, \mathbf{y}^{(j)} = \left[(\mathbf{K}_{\mathrm{ii}} + \boldsymbol{\Delta}\mathbf{K}_{\mathrm{ii}}) \boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} \right] \mathbf{y}^{(j)} \rightarrow \mathbf{0}. \tag{A.11}$$

CA converges to the exact solution with an increasing number of the binomial series terms. An exact solution is obtained in cases where a newly created term becomes a linear combination of the previous terms [83]. Accordingly, $\boldsymbol{\Delta}\boldsymbol{\Psi}_{\mathrm{ib}} \mathbf{y}^{(j)} \rightarrow \mathbf{0}$ and Equation (A.8) is fulfilled.

B

The Levenberg-Marquardt Method

The unknown NN model parameters

$$\mathbf{z}^{\mathrm{T}} = \{w_{ji}^{\mathrm{HL}}, b_j^{\mathrm{HL}}, w_j^{\mathrm{OL}}, b^{\mathrm{OL}} \mid i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, S\} = \{z_1, z_2, \ldots, z_H\}$$

can be estimated using the least squares estimation

$$\min_{\mathbf{z}} f_{\mathrm{E}}(\mathbf{z}) = \min_{\mathbf{z}} \sum_{k=1}^{Q} \left[y^{(k)} - \hat{f}(\mathbf{x}^{(k)}, \mathbf{z}) \right]^2 = \min_{\mathbf{z}} \sum_{k=1}^{Q} \left[\epsilon^{(k)}(\mathbf{z}) \right]^2 \tag{B.1}$$

where $\mathbf{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(Q)}\}$ is a set of sample points, $\mathbf{y}^{\mathrm{T}} = \{y^{(1)}, y^{(2)}, \ldots, y^{(Q)}\}$ is a vector of the corresponding observations, Q is the total number of the data, $\hat{f}(\mathbf{x}, \mathbf{z})$ stands for the NN model and $H = nS + 2S + 1$ is the total number of the NN model parameters.

For the estimation of an optimum $\mathbf{z}$ by an iterative method, first an initial guess $\mathbf{z}_{(0)}$ is selected. Then, this guess is updated at each iteration m by

$$\mathbf{z}_{(m+1)} = \mathbf{z}_{(m)} + \mathbf{p}_{(m)}$$

where the vector $\mathbf{p}$ includes information about the search direction and a scalar learning rate indicating how far to go in the selected search direction.

In the Levenberg-Marquardt (LM) method, $\mathbf{p}_{(m)}$ is defined as

$$\mathbf{p}_{(m)} = -[\mathbf{J}^{\mathrm{T}}(\mathbf{z}_{(m)})\mathbf{J}(\mathbf{z}_{(m)}) + \mu_{(m)}\mathbf{I}]^{-1}\mathbf{J}^{\mathrm{T}}(\mathbf{z}_{(m)})\boldsymbol{\epsilon}(\mathbf{z}_{(m)}) \tag{B.2}$$

where $\mathbf{J}(\mathbf{z})$ is the Jacobian matrix with entries

$$J_{kh}(\mathbf{z}) = \frac{\partial \epsilon^{(k)}(\mathbf{z})}{\partial z_h}, \quad k = 1, 2, \ldots, Q \;\; h = 1, 2, \ldots, H.$$

(see Appendix C for details), $\boldsymbol{\epsilon}(\mathbf{z})^{\mathrm{T}} = \{\epsilon^{(1)}(\mathbf{z}), \epsilon^{(2)}(\mathbf{z}), \ldots, \epsilon^{(Q)}(\mathbf{z})\}$, $\mathbf{I}$ is an identity matrix and $\mu_{(m)}$ is a scalar learning rate for the mth iteration.

In Eq. (B.2), $\mathbf{H}(\mathbf{z}_{(m)}) = \mathbf{J}^{\mathrm{T}}(\mathbf{z}_{(m)})\mathbf{J}(\mathbf{z}_{(m)})$ is an approximation to the Hessian of $f_{\mathrm{E}}(\mathbf{z})$ and $\boldsymbol{\nabla} f_{\mathrm{E}}(\mathbf{z}_{(m)}) = \mathbf{J}^{\mathrm{T}}(\mathbf{z}_{(m)})\boldsymbol{\epsilon}(\mathbf{z}_{(m)})$ is the gradient of $f_{\mathrm{E}}(\mathbf{z})$. If this knowledge is used to rewrite Eq. (B.2),

$$\mathbf{p}_{(m)} = -[\mathbf{H}(\mathbf{z}_{(m)}) + \mu_{(m)}\mathbf{I}]^{-1}\boldsymbol{\nabla} f_{\mathrm{E}}(\mathbf{z}_{(m)}) \tag{B.3}$$

is obtained.

The learning rate μ has several effects on the LM method

1. For all $\mu > 0$, the matrix $\mathbf{H}(\mathbf{z}_{(m)})$ is positive definite and this ensures that $\mathbf{p}$ is a descent direction.

2. For large values of μ, $\mathbf{p}$ defines a short step in the steepest descent direction

   $$\mathbf{p}_{(m)} = -[\mu_{(m)}\mathbf{I}]^{-1}\boldsymbol{\nabla} f_{\mathrm{E}}(\mathbf{z}_{(m)})$$

   which is a good step if the current iterate m is far from an optimum.

3. If μ is very small, the LM method behaves like a Gauss-Newton method. That is,

   $$\mathbf{p}_{(m)} = -\mathbf{H}^{-1}(\mathbf{z}_{(m)})\boldsymbol{\nabla} f_{\mathrm{E}}(\mathbf{z}_{(m)})$$

   which is a good step in the neighborhood of the solution.

Consequently, the LM method begins with a small learning rate μ for instance $\mu_{(0)} = 0.01$. If a step produces a large value, i.e. $f_{\mathrm{E}}(\mathbf{z}_{(m+1)}) > f_{\mathrm{E}}(\mathbf{z}_{(m)})$, this indicates that the current iteration point is not in the neighborhood of the optimum. The learning rate μ is multiplied by a factor $\nu_1 > 1$ (e.g. $\nu_1 = 10$) and a step in the direction of steepest descent is taken. If μ produces a small value, then it is divided by a factor $\nu_2 > 1$ (e.g. $\nu_2 = 10$) and a step towards the Gauss-Newton direction is obtained. In that way, a faster convergence is obtained to the minimum. Hence, the speed of the Newton's method and the guaranteed convergence of the steepest descent are merged in a nice way.

The disadvantage of the algorithm is the storage requirement of the Jacobian. When the number of the NN parameters and the number of the data increase, the method becomes inefficient [100].

C

Chain Rule for Calculating the Partial Derivatives

The partial derivatives

$$\left\{ \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial w_{ji}^{\mathrm{HL}}}, \quad \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial b_{j}^{\mathrm{HL}}}, \quad \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial w_{j}^{\mathrm{OL}}}, \quad \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial b^{\mathrm{OL}}} \;\middle|\; i = 1, 2, \ldots, n \;\; j = 1, 2, \ldots, S \right\}$$

of the NN model $\hat{f}(\mathbf{x}^{(k)}, \mathbf{z})$ with respect to the model parameters

$$\mathbf{z}^{\mathrm{T}} = \{ w_{ji}^{\mathrm{HL}}, b_{j}^{\mathrm{HL}}, w_{j}^{\mathrm{OL}}, b^{\mathrm{OL}} \;\mid\; i = 1, 2, \ldots, n \quad j = 1, 2, \ldots, S \}$$

are required for solving the unconstrained optimization problem

$$\min_{\mathbf{z}} f_{\mathrm{E}}(\mathbf{z}) = \min_{\mathbf{z}} \sum_{k=1}^{Q} \left[y^{(k)} - \hat{f}(\mathbf{x}^{(k)}, \mathbf{z}) \right]^2 = \min_{\mathbf{z}} \sum_{k=1}^{Q} \left[\epsilon^{(k)}(\mathbf{z}) \right]^2 \tag{C.1}$$

with any gradient-based numerical algorithm.

If the vector of the hidden layer outputs is represented by $\mathbf{d}^{\mathrm{HL}} = \{ d_1^{\mathrm{HL}}, d_2^{\mathrm{HL}}, \ldots, d_S^{\mathrm{HL}} \}$, the NN model is

$$\hat{f}(\mathbf{x}^{(k)}, \mathbf{z}) = \Psi^{(2)}(d^{\mathrm{OL}}) = d^{\mathrm{OL}} = \sum_{j=1}^{S} w_{j}^{\mathrm{OL}} \; \Psi_{j}^{(1)}(d_{j}^{\mathrm{HL}}) + b^{\mathrm{OL}}$$

where

$$d_{j}^{\mathrm{HL}} = \sum_{i=1}^{n} w_{ji}^{\mathrm{HL}} x_{i}^{(k)} + b_{j}^{\mathrm{HL}}$$

and the scalar term d^{OL} is the response of the model.

The partial derivatives with respect to the NN model parameters are calculated employing the chain rule as:

$$\begin{aligned}
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial w_{ji}^{\mathrm{HL}}} &= \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d_j^{\mathrm{HL}}} \frac{\partial d_j^{\mathrm{HL}}}{\partial w_{ji}^{\mathrm{HL}}} = \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d_j^{\mathrm{HL}}} x_i^{(k)} \\
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial b_j^{\mathrm{HL}}} &= \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d_j^{\mathrm{HL}}} \frac{\partial d_j^{\mathrm{HL}}}{\partial b_j^{\mathrm{HL}}} = \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d_j^{\mathrm{HL}}} \\
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial w_j^{\mathrm{OL}}} &= \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d^{\mathrm{OL}}} \frac{\partial d^{\mathrm{OL}}}{\partial w_j^{\mathrm{OL}}} = \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d^{\mathrm{OL}}} \; \Psi_j^{(1)}(d_j^{\mathrm{HL}}) \\
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial b^{\mathrm{OL}}} &= \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d^{\mathrm{OL}}} \frac{\partial d^{\mathrm{OL}}}{\partial b^{\mathrm{OL}}} = \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d^{\mathrm{OL}}}.
\end{aligned} \tag{C.2}$$

In Equation (C.2), the derivatives $\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d_j^{\mathrm{HL}}}$ and $\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d^{\mathrm{OL}}}$ still need to be calculated.

The derivative of $\hat{f}(\mathbf{x}^{(k)}, \mathbf{z})$ with respect to d^{OL} is

$$\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d^{\mathrm{OL}}} = \frac{\partial}{\partial d^{\mathrm{OL}}} \left(\Psi^{(2)}(d^{\mathrm{OL}}) \right) = \frac{\partial}{\partial d^{\mathrm{OL}}} \left(d^{\mathrm{OL}} \right) = 1.$$

The derivative of $\hat{f}(\mathbf{x}^{(k)}, \mathbf{z})$ with respect to d_j^{HL} require another application of the chain rule which is

$$\begin{aligned}
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d_j^{\mathrm{HL}}} &= \frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial d^{\mathrm{OL}}} \frac{\partial d^{\mathrm{OL}}}{\partial d_j^{\mathrm{HL}}} = \frac{\partial}{\partial d_j^{\mathrm{HL}}} \left[\sum_{j=1}^{S} w_j^{\mathrm{OL}} \; \Psi_j^{(1)}(d_j^{\mathrm{HL}}) + b^{\mathrm{OL}} \right] \\
&= w_j^{\mathrm{OL}} \; \frac{\partial \Psi_j^{(1)}(d_j^{\mathrm{HL}})}{\partial d_j^{\mathrm{HL}}}.
\end{aligned}$$

Finally the partial derivatives are found as:

$$\begin{aligned}
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial w_{ji}^{\mathrm{HL}}} &= w_j^{\mathrm{OL}} \; \frac{\partial \Psi_j^{(1)}(d_j^{\mathrm{HL}})}{\partial d_j^{\mathrm{HL}}} \; x_i^{(k)} \\
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial b_j^{\mathrm{HL}}} &= w_j^{\mathrm{OL}} \; \frac{\partial \Psi_j^{(1)}(d_j^{\mathrm{HL}})}{\partial d_j^{\mathrm{HL}}} \\
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial w_j^{\mathrm{OL}}} &= \Psi_j^{(1)}(d_j^{\mathrm{HL}}) \\
\frac{\partial \hat{f}(\mathbf{x}^{(k)}, \mathbf{z})}{\partial b^{\mathrm{OL}}} &= 1.
\end{aligned} \tag{C.3}$$

D

Most complying eigenvector and the MAC value:

Let us represent the kth eigenvector of the initial design by Φ_k^0. Assume that $\{\Phi_{k1}^{\mathrm{d}}, \Phi_{k2}^{\mathrm{d}}, \ldots, \Phi_{km}^{\mathrm{d}}\}$ are the m orthogonalized eigenvectors of the current design "d" which correspond to the kth eigenvalue ω_k and span the subspace $\mathbf{V}$.

The most complying eigenvector Φ_k^{d} in the subspace $\mathbf{V}$ with Φ_k^0 and their MAC value are calculated using the formulas:

$$\Phi_k^{\mathrm{d}} = \frac{\{\Phi_{k1}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{k1}^{\mathrm{d}}\|^2}\Phi_{k1}^{\mathrm{d}} + \ldots + \frac{\{\Phi_{km}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{km}^{\mathrm{d}}\|^2}\Phi_{km}^{\mathrm{d}}$$

$$\mathrm{MAC}(\Phi_k^0, \Phi_k^{\mathrm{d}}) = \mathrm{MAC}(\Phi_k^0, \Phi_{k1}^{\mathrm{d}}) + \ldots + \mathrm{MAC}(\Phi_k^0, \Phi_{km}^{\mathrm{d}}).$$

The derivation of the formulas is as follows:

The Degeneracy Theorem [48] states that any vector contained in the subspace $\mathbf{V}$ also belongs to the same eigenvector ω_k.

Let $\Phi_k^{\mathrm{d}} = c_1\Phi_{k1}^{\mathrm{d}} + c_2\Phi_{k2}^{\mathrm{d}} + \ldots + c_m\Phi_{km}^{\mathrm{d}}$ be a vector that lies in $\mathbf{V}$ where c_i, $i = 1, 2, \ldots, m$ are scalars. The distance between Φ_k^{d} and Φ_k^0 can be defined using the Euclidean distance as:

$$\begin{aligned}\|\Phi_k^{\mathrm{d}} - \Phi_k^0\|^2 &= \|c_1\Phi_{k1}^{\mathrm{d}} + c_2\Phi_{k2}^{\mathrm{d}} + \ldots + c_m\Phi_{km}^{\mathrm{d}} - \Phi_k^0\|^2 \\ &= c_1^2\|\Phi_{k1}^{\mathrm{d}}\|^2 + c_2^2\|\Phi_{k2}^{\mathrm{d}}\|^2 + \ldots + c_m^2\|\Phi_{km}^{\mathrm{d}}\|^2 \\ &\quad - 2c_1\{\Phi_{k1}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0 - \ldots - 2c_m\{\Phi_{km}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0 + \{\Phi_k^0\}^{\mathrm{T}}\Phi_k^0\end{aligned} \tag{D.1}$$

Minimizing the distance between Φ_k^{d} and Φ_k^0 with respect to the coefficients c_i, gives the vector Φ_k^{d}. That is,

$$\begin{array}{ccccc}\frac{\partial\|\Phi_k^{\mathrm{d}} - \Phi_k^0\|^2}{\partial c_1} & = & 2c_1\|\Phi_{k1}^{\mathrm{d}}\|^2 - 2\{\Phi_{k1}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0 & = & 0 \\ \vdots & & \vdots & & \vdots \\ \frac{\partial\|\Phi_k^{\mathrm{d}} - \Phi_k^0\|^2}{\partial c_m} & = & 2c_m\|\Phi_{km}^{\mathrm{d}}\|^2 - 2\{\Phi_{km}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0 & = & 0.\end{array} \tag{D.2}$$

Using Eq. (D.2); c_1, c_2,...,c_m are calculated as follows:

$$c_1 = \frac{\{\Phi_{k1}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{k1}^{\mathrm{d}}\|^2}, \quad \ldots, \quad c_m = \frac{\{\Phi_{km}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{km}^{\mathrm{d}}\|^2}.$$

Hence, in the subspace $\mathbf{V}$ the vector Φ_k^{d};

$$\Phi_k^{\mathrm{d}} = \frac{\{\Phi_{k1}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{k1}^{\mathrm{d}}\|^2}\Phi_{k1}^{\mathrm{d}} + \ldots + \frac{\{\Phi_{km}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{km}^{\mathrm{d}}\|^2}\Phi_{km}^{\mathrm{d}}$$

has a high correlation with the vector Φ_k^0 where the correlation is calculated using Eq.(2.39) as:

$$\begin{aligned}
\mathrm{MAC}(\Phi_k^0, \Phi_k^{\mathrm{d}}) &= \frac{(\{\Phi_k^0\}^{\mathrm{T}}\Phi_k^{\mathrm{d}})^2}{\|\Phi_k^0\|^2\|\Phi_k^{\mathrm{d}}\|^2} = \frac{\left(\{\Phi_k^0\}^{\mathrm{T}}\left\{\frac{\{\Phi_{k1}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{k1}^{\mathrm{d}}\|^2}\Phi_{k1}^{\mathrm{d}} + \ldots + \frac{\{\Phi_{km}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{km}^{\mathrm{d}}\|^2}\Phi_{km}^{\mathrm{d}}\right\}\right)^2}{\|\Phi_k^0\|^2\left\|\frac{\{\Phi_{k1}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{k1}^{\mathrm{d}}\|^2}\Phi_{k1}^{\mathrm{d}} + \ldots + \frac{\{\Phi_{km}^{\mathrm{d}}\}^{\mathrm{T}}\Phi_k^0}{\|\Phi_{km}^{\mathrm{d}}\|^2}\Phi_{km}^{\mathrm{d}}\right\|^2} \\
&= \frac{\left[\frac{\left(\{\Phi_k^0\}^{\mathrm{T}}\Phi_{k1}^{\mathrm{d}}\right)^2}{\|\Phi_{k1}^{\mathrm{d}}\|^2} + \ldots + \frac{\left(\{\Phi_k^0\}^{\mathrm{T}}\Phi_{km}^{\mathrm{d}}\right)^2}{\|\Phi_{km}^{\mathrm{d}}\|^2}\right]^2}{\|\Phi_k^0\|^2\left[\left(\frac{\{\Phi_k^0\}^{\mathrm{T}}\Phi_{k1}^{\mathrm{d}}}{\|\Phi_{k1}^{\mathrm{d}}\|^2}\right)^2\|\Phi_{k1}^{\mathrm{d}}\|^2 + \ldots + \left(\frac{\{\Phi_k^0\}^{\mathrm{T}}\Phi_{km}^{\mathrm{d}}}{\|\Phi_{km}^{\mathrm{d}}\|^2}\right)^2\|\Phi_{km}^{\mathrm{d}}\|^2\right]} \\
&= \frac{(\{\Phi_k^0\}^{\mathrm{T}}\Phi_{k1}^{\mathrm{d}})^2}{\|\Phi_k^0\|^2\|\Phi_{k1}^{\mathrm{d}}\|^2} + \ldots + \frac{(\{\Phi_k^0\}^{\mathrm{T}}\Phi_{km}^{\mathrm{d}})^2}{\|\Phi_k^0\|^2\|\Phi_{km}^{\mathrm{d}}\|^2} \\
&= \mathrm{MAC}(\Phi_k^0, \Phi_{k1}^{\mathrm{d}}) + \ldots + \mathrm{MAC}(\Phi_k^0, \Phi_{km}^{\mathrm{d}}).
\end{aligned} \tag{D.3}$$

Nomenclature

The general notation used in the thesis is as follows: The matrices are shown in bold face upper-case letters, e.g. $\mathbf{X}$, the vectors in bold face lower-case letters, e.g. $\mathbf{x}$, and, the components of matrices and vectors as well as scalars are in italic lower-case letters, e.g. x. The superscripts and the subscripts written in italic letters represent indices, e.g. x_i where i may take any defined value. When normal letters are used as superscripts or subscripts, it stands for an abbreviation of a name, e.g. x_{i} where "i" stands for "internal".

Roman symbols	
$\mathcal{A}(\mathbf{x})$	active set
$\mathbf{b}$	bias vector
$\mathbf{d}$	vector of physical degrees of freedom
$\mathcal{E}$	index set for the equality constraints
$\mathbf{f}_{\boldsymbol{\Delta}}(\cdot)$	residual force vector
$\hat{\mathbf{f}}_{\boldsymbol{\Delta}}(\cdot)$	a basis vector in the subspace that contains the approximations of the residual force vectors
$f_{\mathrm{E}}(\cdot)$	error function
$f_{\mathrm{P}}(\cdot)$	regularization function
$\mathcal{F}$	feasible set
$\mathbf{H}_t$	the basis of the subspace that spans the tth residual constraint mode
$\mathcal{I}$	index set for the inequality constraints
$\mathbf{I}$	identity matrix
$\mathbf{K}$	stiffness matrix
$\bar{\mathbf{K}}$	reduced stiffness matrix
$\boldsymbol{\Delta}\mathbf{K}$	residual stiffness matrix
$\mathcal{L}(\cdot,\cdot)$	Lagrange function
$L(\cdot,\cdot)$	likelihood function
$\mathbf{M}$	mass matrix
$\bar{\mathbf{M}}$	reduced mass matrix
$\boldsymbol{\Delta}\mathbf{M}$	residual mass matrix
N_{b}	the number of the binomial series terms
N_{i}	the number of the internal d.o.f.
N_{s}	the number of the constraint modes
N_{T}	truncated number of normal modes

n	number of design variables
$\mathbf{p}$	Kriging model parameter
$\mathbf{q}$	generalized degrees of freedom
Q	the number of data points
$\mathbf{R}$	reaction forces acting on the interface
$\mathbf{\Delta R}$	residual reaction forces acting on the interface
$\mathbf{R}_{\mathrm{D}}$	residual displacement matrix
$\tilde{\mathbf{R}}_{\mathrm{D}}$	the vectors that are going to be employed to enrich the initial fixed interface normal mode set
$\mathbf{r}_{\mathrm{D}}(\cdot)$	residual displacement vector
$\{\mathbf{\Delta r}_k\}_t$	kth binomial series term for the tth residual constraint mode
$\mathbf{R}_{\mathrm{L}}$	residual force matrix
$\hat{\mathbf{R}}_{\mathrm{L}}$	the basis of the subspace that contains the approximations of the residual force vectors
$\mathbf{T}$	transformation matrix
$\mathbf{W}$	weight matrix (in Chapter 4)
$\mathbf{w}$	weight vector
$\mathbf{x}$	design variables
$\mathbf{X}$	a matrix including the sample points
$\mathbf{y}$	unknown coefficients (in Chapter 2), a vector including the output values (in Chapter 4)
$\hat{\mathbf{y}}$	a vector including the predictions of $\mathbf{y}$
$\mathbf{z}$	NN model parameters

Greek symbols

$\mathbf{\Psi}$	constraint mode matrix (in Chapter 2), correlation matrix (in Section 4.3)
$\Psi^{(1)}(\cdot)$	tangent sigmoid basis function
$\Psi^{(2)}(\cdot)$	linear basis function
$\mathbf{\Delta\Psi}$	residual constraint mode matrix
$\mathbf{\Phi}$	normal mode matrix
$\boldsymbol{\lambda}$	Lagrange multiplier vector (in Chapter 3)
λ	regression constant (in Chapter 4)
ω	natural frequency
$\mathbf{\Lambda}$	spectral matrix
$\boldsymbol{\epsilon}$	vector of errors
$\mathcal{E}$	index set for the equality constraints
$\boldsymbol{\theta}$	Kriging model parameter
μ	mean (in Section 4.3)
σ^2	variance (in Section 4.3)

Operators

$dim(\cdot)$	dimension
$\det(\cdot)$	determinant
$(\cdot)^{\mathrm{T}}$	transpose
$(\cdot)^{-1}$	inverse
$\|\| \cdot \|\|$	norm
$\| \cdot \|$	absolute value
$\nabla(\cdot)$	gradient
$\nabla_{\mathbf{x}}(\cdot)$	gradient with respect to $\mathbf{x}$
$\nabla_{\mathbf{xx}}(\cdot)$	Hessian matrix with respect to $\mathbf{x}$

Abbreviations

BS	Binomial Series
CA	Combined Approximations
CB	Craig-Bampton
CMS	Component Mode Synthesis
DOCE	Design of Computer Experiments
DOE	Design of Experiments
d.o.f.	degrees of freedom
ECB	Enriched Craig-Bampton
FE	Finite Element
FLOPs	FLoating-point OPerations
GAs	Genetic Algorithms
KKT	Karush-Kuhn-Tucker
LHS	Latin Hypercube Sampling
LM	Levenberg-Marquardt
MAC	Modal Assurance Criterion
MLHO	Multi-Level Hybrid Optimization
MSLO	Multi-Start Local Optimization
NLP	Non-Linear Programming
NNs	Neural Networks
SBO	Surrogate-Based Optimization
SQP	Sequential Quadratic Programming

General superscripts and subscripts

$(\cdot)^{c}$	cth component
$(\cdot)_{\mathrm{b}}$	boundary part
$(\cdot)_{\mathrm{i}}$	interior part
$(\cdot)^{0}$	initial quantity
$(\cdot)^{\mathrm{HL}}$	quantity corresponding to Hidden Layer
$(\cdot)^{\mathrm{OL}}$	quantity corresponding to Output Layer
$(\cdot)^{(i)}$	ith sample

Acknowledgments

Being a PhD. student far away from my home country was one of the most extraordinary experiences in my life. When I look behind, I am surprised to see the number of incidents that have been filling these four years. I should admit that, I had countless difficult and stressful periods. But at the end, the sun always shone through. *What did not kill me, made me definitely stronger.* I feel more mature and more confident to stand straight-up. I owe a great deal to the beautiful people around me for their support. I would like to take this opportunity to thank these people who were involved at some stage in my four year journey or have walked with me all the time through it.

The research presented in this thesis is part of *Artificial Intelligence for Industrial Applications* (AI4IA) project which is a network of Marie Curie host fellowships for "Early Stage Research Training". Many thanks to Slavko Velickov for writing the project proposal which opened the first door to my future career. I would also like to thank Armen Laziev for all his efforts in coordinating the project. Many experts in SKF Engineering and Research Center took an active role in AI4IA and among them I was adopted by Gerrit van Nijen. Dear Gerrit, our discussions were very educative and fruitful. I greatly enjoyed every single moment of the time that I have spent in SKF. I hope our paths will cross again in the future.

I would like to express my sincere gratitude to my promoter, Andre de Boer, for accepting me as a PhD student, offering valuable criticism and giving me an overall vision which I truly appreciate. Although he knew that I do not have an engineering background, he never lost his confidence and always encouraged me during my research. Dear Andre, thank you very much for trusting me with my decisions and giving me the freedom to develop my own ideas.

I am also deeply thankful to my assistant promoter, Marcel Ellenbroek, for his never ending enthusiasm in my research. His inspiring ideas and challenging questions are invaluable for this thesis. The fact that we have shared the same office for quite a long time, made him exceptionally accessible as a supervisor. He witnessed many of my frustrations as well as the triumphal moments about work. He was always very patient and enlightening. I admire his optimism and his easygoing approach to life.

In the last past two years of my PhD., N120 was frequently visited by me. Dear Geijs (Bert Geijselaers), thank you very much for making me feel welcome in your

office all the time. The breadth of your knowledge and your practical solutions to my problems still impresses me. I am also very thankful that you read my thesis and gave valuable advice which helped shape its structure and contents. I should also mention that Semih and I have beautiful memories of the times that we have spent with you and your family, your wife Karin and your daughters Irene and Liset.

I am very grateful to Daniel Rixen and Pascal Etman for their generosity in sharing their knowledge and experience with me. The discussions that I have had with Daniel Rixen about *reduction and reanalysis methods*, and with Pascal Etman and his former student Simon about *optimization methods* were very fruitful for my research.

Considering that a major part of my time (especially the final year) was spent at the university, without the friendly, positive and helpful people around me; coming at this stage would have been very difficult. I would like to thank all the members of the *Applied Mechanics* and *Production Technology* groups for the stimulating, challenging and enjoyable working environment. Special thanks goes to Debbie. She is the best secretary ever. Without her help, all the administrative deadlocks would be very difficult to break. I am also deeply thankful for all the support and encouragement that she has provided in my personal life. Dear Ruud (Spiering); the scientific discussions that we had were very enlightening. Thank you very much for your feedbacks concerning Chapter 2.

There are some colleagues with whom I have also shared time out of the office hours. Pawel: your friendship means so much to me. All the family gatherings with your wife, Iza, and kids, Dominika and Kinga, were great fun. Wouter (Quak): The guitar evenings that we have organized and the spectacular *ollie ballen* party were very entertaining. When you and Semih are not only playing music for yourself and able to finish a complete song, you two sound very good. Ronald&Antoinette: Thanks for the nice company in the office. I still have very nice memories from Poland and from the home gatherings. Timo&Annemiek: The climbing events and the barbecue organizations were great fun. I would also like to thank my former colleagues and friends; Igor&Ala, Marten&Krista and Ekke; for all the nice memories with them. Marieke (former office mate): Although it has been three years that you have graduated, we still see each other on a regular base. Thanks for all your efforts to keep this relationship alive. You are very special in my and Semih's life.

In the university, the population of the Turkish students are exponentially growing each year. Thanks to Turkish Association at Twente (TUSAT), I have had a chance to meet many Turkish people. The activities and the gatherings brought us closer. I would like to thank them for lessening the feeling of my homesickness with their presence and colorful personalities.

Rock Climbing occupied a significant time during my PhD. studies. The flatness of Netherlands makes this activity quite difficult during especially winter time. Thanks to the indoor climbing hall *Arqué*, I was able to relax my mind after busy working hours. I also had a chance to meet very friendly and wonderful people there, with some of whom I am still spending a lot of time. The first climbing buddies there were Vas, Hichem and Paul. Guys, we have shared so much time together including climbing hours, short climbing trips, barbecues, etc. We have witnessed each others

bitter-sweet moments. Although I am not even in the same country with some of you and we are barely able to see each other, your place will never be forgotten in my heart. Past two years, Jos and Tineke have constituted an important part of my life. The climbing sessions, home gatherings and sharing our daily problems bring us closer every passing day. Life here would be so dull without them. I am also very thankful for all the other climbers in Arqué for their friendships and entertaining chit-chats.

I also want to thank two very precious old friends who have never lost faith in me. Selime, you are like a sister to me. (Mus)TAFA, I don't think that we will ever loose each other. I am very lucky to have you guys in my life.

Dear mommy and daddy; your patience, understanding, encouragement and more importantly unconditional love through my life put me where I am right now. I am forever indebted to you.

I have saved my life's VIP to the end. Semih, whatever I will be writing here will not be enough to express my feelings towards you. You have been always with me through all the ups and downs of the past six years. Thank you for sharing the life with me and for being always honest with your thoughts about my acts and decisions even though you know that they may break my heart. I trust you more than anyone in this life. I love you...

Didem Akçay Perdahcıoğlu
Enschede, June 2010

Research Deliverables

Journal Papers

D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, H.J.M. Geijselaers, and A. de Boer. Updating the Craig-Bampton Reduction Basis for Efficient Structural Reanalysis. *International Journal for Numerical Methods in Engineering*, 2010. Accepted.

D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, P.J.M. van der Hoogt, and A. de Boer. An optimization method for dynamics of structures with repetitive component patterns. *Structural and Multidisciplinary Optimization*, 39(6):557-567, 2009.

J.W Wind, D. Akçay Perdahcıoğlu, and A. de Boer. Distributed multilevel optimization for complex structures. *Structural and Multidisciplinary Optimization*, 36(1):71-81, 2008.

Conference Papers

D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, and A. de Boer. On the coupling of reanalysis techniques with a surrogate-based design optimization method. *International Council of the Aeronautical Sciences, ICAS*, 2010.

D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, and A. de Boer. A hybrid design optimization method using enriched Craig-Bampton approach. In 16*th International Congress on Sound and Vibration, ICSV*, 2009.

D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, P.J.M. van der Hoogt, and A. de Boer. Design optimization of structures including repetitive patterns. In *International conference on Engineering Optimization, EngOpt*, 2008.

D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, P.J.M. van der Hoogt, and A. de Boer. Design Optimization utilizing dynamic substructuring and artificial intelligence techniques. In 26*th International Modal Analysis Conference, IMAC-XXV*, 2008.

D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, P.J.M. van der Hoogt, and A. de Boer. A hybrid design optimization strategy for solving dynamic problems of complex structures. In 2*nd International conference on Artificial intelligence for industrial applications, AI4IA*, 2008.

D. Akçay Perdahcıoğlu, P.J.M. van der Hoogt, and A. de Boer. Design optimization applied in structural dynamics. In 1*st International conference on Artificial intelligence for industrial applications, AI4IA*, 2007.

References

[1] *The Mathworks, Genetic Algorithm and Direct Search Toolbox^{TM}2.3, User's Guide.*

[2] *The Mathworks, Optimization Toolbox^{TM}4, User's Guide.*

[3] D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, and A. de Boer. A hybrid design optimization method using enriched craig-bampton approach. In *16th International Congress on Sound and Vibration, ICSV*, 2009.

[4] D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, H.J.M. Geijselaers, and A. de Boer. Updating the Craig-Bampton reduction basis for efficient structural reanalysis. *International Journal for Numerical Methods in Engineering*, 2010. Accepted with minor revision.

[5] D. Akçay Perdahcıoğlu, M.H.M. Ellenbroek, P.J.M. Hoogt van der, and A. de Boer. An optimization method for dynamics of structures with repetitive component patterns. *Structural and Multidisciplinary Optimization*, 39(6):557–567, 2009.

[6] D. Anderson and G. McNeill. Artificial neural networks technology. A state-of-the-art report, Data & Analysis Center for Software DACS, 1992.

[7] ANSYS®. *Release 11.0, Documentation for ANSYS, Elements Reference*, 2007.

[8] M. Armstrong. Problems with universal kriging. *Mathematieal Geology*, 16(1):101–108, 1984.

[9] C. Audet and J.E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):188–217, 2006.

[10] W. Banzhaf, P. Nordin, R.E. Keller, and F.D. Francone. *Genetic programming: an introduction on the automatic evolution of computer programs and its applications.* Morgan Kaufmann, 1997.

[11] J.F.M. Barthelemy and R.T. Haftka. Approximation concepts for optimum structural design a review. *Structural and Multidisciplinary Optimization*, 5(3):129–144, 1993.

[12] M.P. Bendsøe and O. Sigmund. *Topology optimization: theory, methods, and applications*. Springer, 2003.

[13] W.A. Benfield and R.F. Hruda. Vibration analysis of structures by component mode substitution. *AIAA*, 9(7):12551261, 1971.

[14] L. Berke and P. Hajela. Application of neural nets in structural mechanics. *Structural Optimization*, 4:90–98, 1992.

[15] D. Bienstock and G.L. Nemhauser, editors. *Integer programming and combinatorial optimization*. Springer, 2004.

[16] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] P.T. Boggs and J.W. Tolle. Sequential quadratic programming. *Acta Numerica*, pages 1–51, 1995.

[18] M.H.A Bonte. *Optimization Strategies for Metal Forming Processes*. PhD thesis, University of Twente, 2007.

[19] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. ICASE Report 98-47, NASA, Langley Research Center, 1998.

[20] A. Bouazzouni. Selecting a ritz basis for the reanalysis of the frequency response functions of modified structures. *Journal of Sound and Vibration*, 198(2):309–322, 1997.

[21] F. Bourquin. Analysis and comparison of several component mode synthesis methods on one-dimensional domains. *Numerische Mathematik*, 58:11–34, 1990.

[22] M.D. Buhmann. *Radial Basis Functions*. Cambridge University Press, 2003.

[23] C. Cerulli, F. van Keulen, and D.J. Rixen. Dynamic reanalysis and component mode synthesis to improve aircraft modeling for load calculation. In *48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007.

[24] S. Chatterjee and A.S. Hadi. *Regression Analysis by Example*. Wiley-Interscience, 4 edition, 2006.

[25] R. Choudhary and J. Michalek. Design optimization in computer aided architectural design. In *International Conference of the Association for Computer Aided Architectural Design Research In Asia*, 2005.

[26] C.A. Coello Coello. Theoretical and numerical constraint handling techniques used with evolutionary algorithms. a survey of the state of art. *Computer Methods in Applied Mechanics*, 191:1245–1287, 2002.

[27] A.R. Conn, N. Gould, and P.L. Toint. Improving the decomposition of partially separable functions in the context of large-scale optimization: A first approach. Technical report, IBM T.J. Watson Research Center, 1993.

[28] A.R. Conn, N. Gould, and P.L. Toint. *L. Dixon, D.F. Shanno and E. Spedicato (ed.) Algorithms for continuous optimization: State of the art*, chapter Large-scale nonlinear constrained optimization. Kluwer Academic Publishers, 1994.

[29] A.R. Conn, N.I.M Gould, and P.L. Toint. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM Journal of Numerical Analysis*, 28(2):545–572, 1991.

[30] A.R. Conn, N.I.M Gould, and P.L. Toint. A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Mathematics of Computation*, 66(217):261–288, 1997.

[31] A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-region methods*. MPS-SIAM series on optimization, 2000.

[32] A.R. Conn, K. Scheinberg, and L.N. Vicente. *Introduction to Derivative-Free Optimization*. MPS-SIAM series on optimization, 2008.

[33] R.D. Cook, D.S. Malkus, M.E. Plesha, and R.J. Witt. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, 4 edition, 2002.

[34] R.R. Craig Jr. A review of substructure coupling methods for dynamic analysis. Technical report, NASA CP-2001, 1976.

[35] R.R. Craig Jr. Coupling of substructures for dynamic analyses: An overview. In *Structures, Structural Dynamics and Material Conference. 41st AIAA/ASME/ASCE/AHS/ASC*, 2000.

[36] R.R. Craig Jr. and M.C.C. Bampton. Coupling of substructures for dynamic analysis. *AIAA*, 6(7):1313–1319, 1968.

[37] S. Donders, R. Hadjit, L. Hermans, M. Brughmans, and W. Desmet. A wave-based substructuring approach for fast modification predictions and industrial vehicle optimization. *International Journal of Vehicle Design*, 43(1-2):100–115, 2007.

[38] C.T. Cornelius (Editor). *Intelligent Systems: Technology and Applications*, volume 6. CRC Press, 2002.

[39] M. Ehrgott and X. Gandibleux. *Multiple criteria optimization: state of the art annotated bibliographic surveys*. Kluwer Academic Publishers, 2002.

[40] D. Emmrich, P. Haeussler, and A. Albers. Automated structural optimization of flexible components using msc.adams/flex and msc.nastran sol200. In *1st European MSC.ADAMS User Conference*, 2002.

[41] L. Erikson, E. Johansson, N. Kettaneh-Wold, C. Wikström, and S. Wold. *Design of Experiments: Principles and Applications*. Umetrics Academy, 2000.

[42] P. Etman. *Optimization of Multibody Systems using Approximation Concepts*. PhD thesis, Technische Universiteit Eindhoven, 1997.

[43] C. Farhat and M. Geradin. On a componenet mode synthesis method and its applications to incompatible substructures. *Computers & Structures*, 51(5):459–473, 1992.

[44] C. Fillon. *New Strategies for Efficient and Practical Genetic Programming*. PhD thesis, Università degli Studi di Trieste, 2008.

[45] F.D. Foresee and M.T. Hagan. Gauss newton approximation to bayesian learning. In *IEEE Trans. on Neural Networks, ICNN97*, pages 1930–1935, 1997.

[46] A. Forrester, A. Sobester, and A. Keane. *Engineering Design via Surrogate Modeling. A Practical Guide*. John Wiley & Sons Ltd., 2008.

[47] A.J. Forrester and A.J. Keane. Recent advances in surrogate-based optimization. *Aerospace Sciences*, 45:50–79, 2009.

[48] M. Géradin and D. Rixen. *Mechanical Vibrations. Theory and Application to Structural Dynamics*. Wiley, 1994.

[49] P.E. Gill, W. Murray, D.B. Ponceleon, and M.A. Saunders. Solving reduced kkt systems in barrier methods for linear and quadratic programming. Technical Report SOL 91-7, Department of Mathematics, University of California at San Diego, 1991.

[50] P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright. Procedures for optimization problems with a mixture of bounds and general constraints. *ACM Transactions on Mathematical Software*, 10(3):282–298, 1984.

[51] A.A. Giunta, J.M. Dudley, R. Narducci, B. Grossman, R.T. Haftka, W.H. Mason, and L.T. Watson. Noisy aerodynamic response and smooth approximations in hsct design. In *5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis & Optimization*, page 11171128, 1994.

[52] A.A. Giunta, S.F. Wojtkiewicz Jr., and M.S. Eldred. Overview of modern design of experiments methods for computational simulations. In *41st AIAA Aerospace Sciences Meeting and Exhibition*, 2003.

[53] T. Goel, R.T. Haftka, W. Shyy, and N.V. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33:199–216, 2007.

[54] R.L. Goldman. Vibration analysis by dynamic partitioning. *AIAA*, 7:1152–1154, 1969.

[55] P. Goovaerts. Ordinary cokriging revisited. *Mathematical Geology*, 30(1):21–42, 1998.

[56] N.I.M Gould, D. Orban, and P.L. Toint. Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, 14:299–361, 2005.

[57] M. J. Griffin. *Handbook of human vibration*. Elsevier Academic Press, 1990. Reprint 2004.

[58] I. Griva, S.G. Nash, and A. Sofer. *Linear and Nonlinear Optimization*. SIAM, 2 edition, 2009.

[59] A.A. Groenwold and L.F.P. Etman. On the conditional acceptance of iterates in sao algorithms based on convex separable approximations. *Structural and Multidisciplinary Optimization*, 2010.

[60] C. Grosan, A. Abraham, and H. Ishibuchi. *Hybrid Evolutionary Algorithms*, volume 75 of *Studies in Computational Intelligence*. Springer-Verlag, 2007.

[61] R.T. Haftka. First and second order constraint approximations in structural optimization. *Computational Mechanics*, 3:89–104, 1988.

[62] R.T. Haftka and H.M. Adelman. Recent developments in structural sensitivity analysis. *Structural Optimization*, 1:137–151, 1989.

[63] R.T. Haftka and Z. Gürdal. *Elements of Structural Optimization*, volume 11. Kluwer Academics Publishers, 3 edition, 1992.

[64] M.T. Hagan, H.B. Demuth, and M. Beale. *Neural Network Design*. An International Thomson Publishing Company, ITP, 1996.

[65] S.P. Han. A globally convergent method for nonlinear programming. *Journal of Optimization Theory and Applications*, 22(3):297–309, 1977.

[66] M.H. Hassoun. *Fundamentals of Artificial Neural Networks*. MIT Press, 1995.

[67] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer-Verlag, New York, 2001.

[68] J.H. Heo and K.F. Ehman. A method for substructural sensitivity synthesis. *Journal of Vibration and Acoustics*, 113:201–208, 1991.

[69] A.E. Hoerl and R.W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. Technometrics is currently published by American Statistical Association.

[70] C.M.E. Holden and A.J. Keane. Visualization methodologies in aircraft design. In *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.

[71] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[72] G. Hou, V. Maroju, and R.J. Yang. Component mode synthesis based design optimization method for local structural modification. *Structural Optimization*, 10:128–136, 1995.

[73] W.C. Hurty. Dynamic analysis of structural systems using component modes. *AIAA*, 3(4):678–685, 1965.

[74] J.H. Jacobs, L.F.P. Etman, F. van Keulen, and J.E. Rooda. Framework for sequential approximate optimization. *Structural and Multidisciplinary Optimization*, 27:384–400, 2004.

[75] D. Johnson. *Advanced Structural Mechanics*. Thomas Telford Ltd., 2 edition, 2000.

[76] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Global Optimization*, 13(4):455–492, 1998.

[77] V.R. Joseph, Y. Hung, and A. Sudjianto. Blind kriging: A new method for developing metamodels. *ASME Journal of Mechanical Design*, 130(3), 2008.

[78] F. Jurecka. *Robust Design Optimization Based on Metamodeling Techniques*. PhD thesis, Lehrstuhl für Statik der Technischen Universität München, 2007.

[79] C.W. Kim, S.N. Jung, and J.H. Choi. Automotive structure vibration with component mode synthesis on a multi-level. *International Journal of Automotive Technology*, 9(1):119–122, 2008.

[80] U. Kirsch. Efficient sensitivity analysis for structural optimization. *Computer methods in applied mechanics and engineering*, 117:143–156, 1994.

[81] U. Kirsch. *Design Oriented Analysis of Structures, A Unified Approach*, volume 95 of *Solid Mechanics and Its Applications*. Springer, 2002.

[82] U. Kirsch. Approximate vibration reanalysis of structures. *AIAA*, 41(3):504–511, 2003.

[83] U. Kirsch. *Reanalysis of Structures. A Unified Approach for Linear, Nonlinear, Static and Dynamic Systems*, volume 151 of *Solid Mechanics and Its Applications*. Springer, 2008.

[84] U. Kirsch, M. Bogomolini, and F. van Keulen. Efficient finite difference design sensitivities. *AIAA*, 43(2):399–405, 2005.

[85] U. Kirsch and M. Bogomolni. Procedures for approximate eigenproblem reanalysis of structures. *International Journal for Numerical Methods in Engineering*, 60:1969–1986, 2004.

[86] U. Kirsch, M. Kocvara, and J. Zowe. Accurate reanalysis of structures by a preconditioned conjugate gradient method. *International Journal for Numerical Methods in Engineering*, 55(2):233–251, 2002.

[87] U. Kirsch and P.Y. Papalambros. Exact and accurate solutions in the approximate reanalysis of structures. *AIAA*, 39(11):2198–2205, 2001.

[88] N.K Kittusamya and B. Buchholz. Whole-body vibration and postural stress among operators of construction equipment: A literature review. *Journal of Safety Research*, 35:255–261, 2004.

[89] L. Krog, A. Tucker, and G. Rollema. Application of topology, sizing and shape optimization methods to optimal design of aircraft components. Copyright© Altair Engineering Ltd., 2002.

[90] R.M. Lewis and V. Torczon. Pattern search methods for linearly constrained minimization. *SIAM Journal on Optimization*, 10(3):917–941, 2000.

[91] J.P. Loeppky, J. Sacks, and W.J. Welch. Choosing the size of a computer experiment: A practical guide. *Technometrics*, 51(4):366–376, 2009.

[92] R.H. Mac Neal. A hybrid method of component mode synthesis. *Computers and Structures*, 1(4):581–601, 1971.

[93] D.J.C. Mackay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

[94] D. Markovic, K.C. Park, and A. Ibrahimbegovic. Reduction of substructural interface degrees of freedom in flexibility-based component mode synthesis. *International Journal for Numerical Methods in Engineering*, 70:163–180, 2007.

[95] J.D. Martin and T.W. Simpson. Use of kriging models to approximate deterministic computer models. *AIAA*, 43(4):853–863, 2005.

[96] G. Masson, B. Ait Brik, S. Cogan, and N. Bouhaddi. Component mode synthesis (cms) based on an enriched ritz approach for efficient structural optimization. *Journal of Sound and Vibration*, 296:845–860, 2006.

[97] K. Maute, M. Nikbay, and C. Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *AIAA*, 39(11):2051–2061, 2001.

[98] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239245, 1979.

[99] L. Meirovitch. *Fundamentals of Vibrations*. Mechanical Engineering. McGraw-Hill International, 2001.

[100] M.B. Menhaj and M.T. Hagan. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.

[101] M.D. Morris and T.J. Mitchell. Exploratory designs for computational experiments. Technical report, Oak Ridge National Laboratory, 1992.

[102] A. Nagamatsu, P. Sok-Chu, T.I. Ishii, and S. Honda. Vibration analysis and structural optimization of a press machine. *Finite Elements in Analysis and Design*, 14:297–310, 1993.

[103] D. Nguyen and B. Widrow. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Proceedings of the International Joint Conference on Neural Networks*, pages 21–26, 1990.

[104] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[105] I. Nowak. *Relaxation and decomposition methods for mixed integer nonlinear programming*, volume 152. Birkhäuser Verlag, 2005.

[106] A.B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452, 1992.

[107] P.Y. Papalambros and D.J. Wilde. *Principles of optimal design: modeling and computation*. Cambridge University Press, 2 edition, 2000.

[108] K. Park and M.P. Castanier. Component mode synthesis for substructures for non-matching interfaces. In *Noise and Vibration Conference and Exhibition*, 2007.

[109] E.P. Petrov, R. Vitali, and R.T. Haftka. Optimization of mistuned bladed discs using gradient-based response surface approximations. In *AIAA-2000-1522*, 2000.

[110] B.T. Polyak. Newton's method and its use in optimization. *European Journal of Operational Research*, 181:10861096, 2007.

[111] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanatan, and P.K. Tucker. Surrogate-based analysis and optimization. *Aerospace Sciences*, 41:1–28, 2005.

[112] J.W.S. Rayleigh. *The Theory of Sound*, volume 1. Dover, 2 edition, 1945. Unabridged second revised edition with an historical introduction by Robert B. Lindsay.

[113] J.N. Reddy. *An Introduction to the Finite Element Method*. Engineering Mechanics. McGraw-Hill International, 2 edition, 1993.

[114] J. Renegar. *A mathematical view of interior-point methods in convex optimization*. MPS-SIAM series on optimization, 2001.

[115] D. Rixen, C. Farhat, and M. Géradin. A two-step, two-field hybrid method for the static and dynamic analysis of substructure problems with conforming and non-conforming interfaces. *Computer Methods in Applied Mechanics and Engineering*, 154:229–264, 1998.

[116] D.J. Rixen. A dual craig-bampton method for dynamic substructuring. *Journal of Computational and Applied Mathematics*, 168(1-2):383–391, 2002.

[117] S. Rubin. Improved component mode representation for structural dynamic analysis. *AIAA*, 13(8):995–1006, 1975.

[118] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435, 1989.

[119] K. Saitou, K. Izui, S. Nishiwaki, and P. Papalambros. A survey of structural optimization in mechanical product development. *Journal of Computing and Information Science in Engineering*, 5(3):214–226, 2005.

[120] A.A. Shabana. *Theory of Vibration: an introduction*. Mechanical Engineering. Springer, 2 edition, 1995.

[121] A.J. Simpson, A.J. Booker, D. Ghosh, A.A. Giunta, P.N. Koch, and R.J. Yang. Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Journal of Structural and Multidisciplinary Optimization*, 27:302–313, 2004.

[122] T.M. Simpson, D.K.J. Lin, and W. Chen. Sampling strategies for computer experiments: Design and analysis. *Reliability and Safety*, 2(3):209–240, 2001.

[123] T.W. Simpson, V. Toropov, V. Balabanov, and F.A.C. Viana. Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come - or not. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2008.

[124] H. Sippel, E.L. Stanton, and J.G. Crose. The usage of numerical optimization in the development process. In *International FEM Congress, Baden-Baden, Germany*, 1998.

[125] S.N. Sivanandam, S. Sumathi, and S.N. Deepa. *Introduction to Neural Networks using MATLAB 6.0*. TATA MCGRAW HILL PUBLIHSERS, 7 edition, 2008.

[126] M.J. Smith. *An Evaluation of Component Mode Synthesis for Modal Analysis of Finite Element Models*. PhD thesis, The University of British Columbia, 1993.

[127] A.J. Smola and B. Scholkopf. A tutorial on support vector regression. *Statistics and Computing*, 14:199222, 2004.

[128] J. Sobieszczanski-Sobieski, S. Kodiyalam, and R.Y. Yang. Optimization of a car body under constraints of noise, vibration and harshness (nvh), and crash. *Journal of Structural and Multidisciplinary Optimization*, 22:295–306, 2001.

[129] M. Sugiyama and H. Ogawa. Optimal design of regularization term and regularization parameter by subspace information criterion. *Neural Networks*, 15(3):349–361, 2002.

[130] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359–373, 1987.

[131] V.V. Toropov, A.A. Filatov, and A.A. Polynkin. Multi-parameter structural optimization using fem and multi-point explicit approximations. *Structural Optimization*, 6:7–14, 1993.

[132] S. Tosserams, L.F.P. Etman, and J.E. Rooda. Multi-modality in augmented lagrangian coordination for distributed optimal design. *Journal of Structural and Multidisciplinary Optimization*, 2009.

[133] A.H. van den Boogaard. *Thermally Enhanced Forming of Aluminium Sheet: Modelling and Experiments*. PhD thesis, Universiteit Twente, 2002.

[134] F. Van Keulen, R.T. Haftka, and N.H. Kim. Review of options for structural design sensitivity analysis. part 1: Linear systems. *Computer methods in applied mechanics and engineering*, 194:3213–3243, 2005.

[135] F. Van Keulen and V.V. Toropov. New developments in structural optimization using adaptive mesh refinement and multi-point approximations. *Engineering Optimization*, 29:217–234, 1997.

[136] F.A.C Viana, R.T. Haftka, and V. Steffen Jr. Multiple surrogates: how cross-validation errors can help us to obtain the best predictor. *Journal of Structural and Multidisciplinary Optimization*, 39:439–457, 2009.

[137] W.J. Welch, R.J. Buck, J. Sacks, H.P. Wynn, T.J. Mitchell, and M.D. Morris. Screening, predicting and computer experiments. *Technometrics*, 34(1):15–25, 1992.

[138] J. Wind. Silent component fast, a global local optimization method for dynamic problems. Master's thesis, University of Twente, 2005.

[139] J. Wind, D. Akçay Perdahcıoğlu, and A. de Boer. Distributed multilevel optimization for complex structures. *Structural and Multidisciplinary Optimization*, 36(1):71–81, 2008.

[140] T.T. Wong, W.S. Luk, and P.A. Heng. Sampling with hammersley and halton points. *Graphics, Gpu, and Game tools*, 2(2):9–24, 1997.

[141] L.E. Zerpa, N.V. Queipo, S. Pintos, and J.L. Salager. An optimization methodology of alkaline surfactant polymer flooding processes using field scale numerical simulation and multiple surrogates. *Journal of Petroleum Science and Engineering*, 47:197–208, 2005.