Model Reduction & Interface Modeling in Dynamic Substructuring

– Application to a multi-megawatt wind turbine –

MSc. Thesis

Author:	P.L.C. van der Valk
Supervisor Siemens:	Ir. S.N. Voormeeren
Supervisor TU Delft:	Prof. Dr. Ir. D.J. Rixen

January 14, 2010

Abstract

In modern day society concern is growing about the use of fossil fuels to meet our constantly rising energy demands, and the need for more sustainable energy is growing. Wind energy certainly has the potential to play a significant role in a sustainable future world energy supply and the wind power industry has grown to a globalized multi billion dollar industry. Manufacturers do not only compete with each other, but also with the traditional fossil energy sources. In order to come out on top, manufacturers are aiming at lowering the total turbine costs in order to lower the cost of renewable energy.

An important way of achieving this is by reducing the total weight of turbine, by optimizing the design of each individual component. This causes a chain reaction of benefits as less material is used, transport and installation is made easier, a smaller foundation can be used and so on. On the downside, these optimized turbine designs generally introduce more flexibility to the structure. As a result, components start to exhibit local dynamic behavior, which can lead to increased component loading and decreased reliability. However, the aero-elastic models commonly used in wind turbine engineering are often incapable of predicting these local dynamic effects and their interaction with the global dynamics, due to their relatively few degrees of freedom and geometric simplifications. Therefore, a need exists for more detailed structural dynamic analysis tools, without losing generality and versatility.

In this thesis the paradigm of *dynamic substructuring* is proposed to fill this need for detailed dynamic analysis tools in wind turbine engineering. Dynamic substructuring is a way to obtain the structural dynamic behavior of large and/or complex structures by dividing them into several smaller, simpler substructures (or components) of which the dynamic behavior is generally easier to determine. The dynamics of the total structure are then obtained by assembling the dynamic models of the components. A number of different techniques can be distinguished within the field of dynamic substructuring. In this thesis the emphasis is on the application and theory of *Component Mode Synthesis* techniques. The theoretical contributions are discussed first.

Firstly, a general framework for substructure assembly is presented. In addition to the classic "primal" or "dual" assembly of interface displacements, this framework allows to assemble interface forces in a similar "primal" or "dual" manner. Furthermore, the framework enables the direct assembly of interface displacements and interface forces. The latter is called "mixed" assembly. In other words, direct assembly of stiffness matrices with flexibility matrices.

Secondly, all common component model reduction techniques (Craig-Bampton, Rubin,

etc.) and the relatively new Dual Craig-Bampton method are discussed. The Mixed Craig-Bampton method is introduced in this work and is a true generalization of the Craig-Bampton and Dual Craig-Bampton methods. It is shown that the accuracy of the Mixed Craig-Bampton methods is always in line with the Craig-Bampton and Dual Craig-Bampton methods, thereby emphasizing its versatility.

Furthermore, a number of interface modeling strategies are discussed. Firstly, to enable assembly using only six degrees of freedom per interface, interface rigidification is discussed. A second option is to model the interface as fully flexible and retain all its degrees of freedom, which could result in incompatible substructure meshes. To overcome this issue several methods for assembly of non-conforming meshes are discussed. Finally, modeling of dynamic effects resulting from the interface itself (e.g. dynamic behavior of a bolt connection) is also presented in this thesis.

Finally, interface reduction techniques are presented. Reduction of interface displacements is already well known from literature. On the other hand, reduction of interface forces, which is also presented in this work, has not been found in literature. It is shown that both methods are able to significantly reduce the number of degrees of freedom of the (reduced) substructure models.

Using these methods and techniques, a dynamic substructuring analysis is performed using different reduced component models of the yaw system of a 2.3 Megawatt Siemens wind turbine. All the substructure components are modeled using the finite element method, but due to time limitations only one of the components is validated through measurements. By using the different component model reduction techniques, we were able to reduce the total number of degrees of freedom from almost 300.000 to approximately 750 for the entire yaw system, while maintaining an accurate model of the dynamic behavior for the frequency range of interest.

From these results one can conclude that the dynamic substructuring approach shows great potential for use in wind turbine engineering. Even though some models are significantly simplified and not all the models used here are validated, it is clear that the techniques presented in this thesis allow for creating compact and accurate descriptions of the dynamic behavior of wind turbines. Nonetheless several challenges, with respect to non-linear models, controller models and others, are still to be met in order to generalize the methodology for application in wind turbine engineering.

Acknowledgements

In January 2009 I was given the opportunity to do my internship and MSc. assignment at Siemens Wind Power. Now, a year later, the MSc. project is completed and will be presented in this thesis. A completed MSc. thesis will generally only give a good insight in the work performed, but not in the process to get there. One could compare the process of a MSc. assignment to a journey with an unknown destination; the only certain thing is the starting point. The ending point, the final goal, is generally not fully known. The road to get to the final MSc. thesis has a lot of intersections and dead-end side streets, it is therefore essential to have a number of "personal navigational satellites" which will guide you during the journey and make sure you will not get lost. Here, I would like to thank those who have guided me on the way or helped me in any way to achieve what is presented in this work.

First of all, I would like to thank Sven Voormeeren, who enabled me to cooperate in the *Hybrid Dynamic Substructuring in Wind Turbine Engineering* project at Siemens Wind Power. And who, apart from being a motivational, inspirational and professional supervisor which allowed me to perform on a higher level, also became a good friend.

Next to this I would also liked to thank prof. dr. ir. D.J. Rixen and the rest of the *Engineering Dynamics* staff whose enthusiasm and support have inspired me throughout my Master.

Furthermore I'd like to thank all my former colleagues from the The Hague office at Siemens Wind Power for the great times and support. But I'd also like to thank the former colleagues from Brande, especially in the Measurement, Structural and Load departments for their cooperation.

And last but not least, I would like to thank my family and friends, who where not involved directly in the work, but were still essential in the process. I would especially like to thank my parents, whose support and encouragement has been essential throughout my academic career.

Delft,
January 2010

Paul van der Valk

Contents

Nomenclature

Symbols

$\boldsymbol{M}$	mass matrix
$\boldsymbol{K}$	stiffness matrix
$\boldsymbol{C}$	damping matrix
$\boldsymbol{G}_f$	flexibility matrix
$\boldsymbol{G}_{res}$	residual flexibility matrix
$\boldsymbol{u}$	displacement DoF vector
$\boldsymbol{f}$	external force vector
$\boldsymbol{g}$	coupling force vector
$\boldsymbol{q}$	vector of generalized DoF
$\boldsymbol{\eta}$	modal DoF
$\boldsymbol{B}$	signed boolean matrix
$\boldsymbol{b}$	local boolean matrix
$\boldsymbol{L}$	boolean matrix
$\boldsymbol{\lambda}$	interface force intensities
$\boldsymbol{R}$	reduction matrix
$\boldsymbol{Z}$	dynamic stiffness matrix
$\boldsymbol{M}$	dynamic flexibility matrix
$\boldsymbol{\Phi}_f$	free interface vibration modes
$\boldsymbol{\Phi}_i$	fixed interface vibration modes
$\boldsymbol{\Phi}_r$	rigid body modes
$\boldsymbol{\Phi}_m$	mixed interface vibration modes
$\boldsymbol{\Psi}_C$	constraint modes
$\boldsymbol{\Psi}_a$	attachment modes
$\boldsymbol{\Psi}_{ar}$	residual attachment modes
$\boldsymbol{T}$	transformation matrix
$\boldsymbol{D}$	collocation matrix
$\boldsymbol{\Psi}_u$	interface displacement modes
$\boldsymbol{\Psi}_{int}$	interface modes
$\boldsymbol{\Psi}_\lambda$	interface force modes
$\boldsymbol{\Psi}_u$	interface displacement modes
ω	circular frequency

$\star_i$	part of matrix or vector acting on internal DoF
$\star_b$	part of matrix or vector acting on boundary (interface) DoF
$\star_d$	"dual" interface DoF
$\star_p$	"primal" interface DoF
$\tilde{\star}$	reduced (or associated to a reduced) matrix or vector
$\bar{\star}$	primal assembled matrix or vector
$\star^{(s)}$	variable associated to substructure number s

Abbreviations

DS	Dynamic substructuring
CMS	Component Mode Synthesis
DoF	Degree(s) of freedom
NDoF	Number of degrees of freedom
MAC	Modal Assurance Criterion
SUMAC	Substructure Modal Assurance Criterion
FRF	Frequency response function
FE	Finite element
FEM	Finite element method
LM FBS	Lagrange multiplier frequency based substructuring
rbm	Rigid body modes
CB	Craig-Bampton
DCB	Dual Craig-Bampton
MCB	Mixed Craig-Bampton
SWT	Siemens wind turbine
YR-TT	Yaw ring and towertop
YP	Yaw pad (array)
BP	Bedplate
YGB	Yaw gearbox (array)

CHAPTER 1

Introduction

1.1 Research context

At present there are few topics as heavily debated as "sustainability". On a daily basis the media are full of items on climate change, oil prices, CO_2 reductions, rising energy consumption and so on. Regardless of one's opinion on the subject, a fact of the matter is that more sustainable ways of power generation need to be found simply because the currently used resources will some day be exhausted [1].

One of the more promising ways of generating "green" electricity on a large scale is provided by wind energy. As a result, the wind turbine industry has undergone a huge transition: from a small group of (mainly Danish) enthusiasts in the early 1980's, the modern wind power industry now has grown to a globalized multi billion dollar industry.[2] However, to enable wind power to truly fulfill a significant role in a sustainable future energy supply, a number of technological challenges are still to be met. One of those challenges concerns the correct modeling and analysis of the structural dynamic behavior of the wind turbine.

Naturally a wind turbine, with its large and relatively slender structure and the complex excitations, exhibits all kinds of structural dynamic behavior. The dynamic loading and structural vibrations sometimes can cause problems, from cracking blades, breaking gearboxes to "singing" towers. These problems have not been limited to a single manufacturer, but simply seem inherent to the structure of a modern wind turbine.

To cope with these dynamic effects, wind turbine manufacturers, research institutes and universities have developed many different aero-elastic codes [2]. These advanced codes are perfectly suited to analyze the global dynamics of a wind turbine, taking into account aerodynamic loads and coupling, possibly wave loads (for offshore turbines), and hence are commonly used for certification purposes.

Driven by today's highly competitive wind turbine market, manufacturers are searching for ways to optimize their turbine designs and hence save costs. An important way of achieving this is by reducing the total weight of turbine, by optimizing the design of each individual component. This causes a chain reaction of benefits as less material is used, transport and installation is made easier, a smaller foundation can be used and so on. On the downside, these optimized turbine designs generally introduce more flexibility to

[1] Introduction from [1]

[2] From 2002 onwards, the wind power industry has seen an annual growth of no less than 25%.

the structure. As a result, components start to exhibit local dynamic behavior, which can lead to increased component loading and decreased reliability. In some cases the local dynamic effects can interact with the global dynamics of the turbine, or vice versa. Thorough understanding of these dynamics is a prerequisite to further increase the overall reliability of a wind turbine. However, the aero-elastic models commonly used in wind turbine engineering are often incapable of predicting these local dynamic effects and their interaction with the global dynamics, due to their relatively few degrees of freedom and geometric simplifications. Therefore, a need exists for more detailed structural dynamic analysis tools, without losing generality and versatility.

1.2 Research project and thesis assignment

In order to try to fill this need for detailed dynamic analysis tools, a PhD project headed by Sven Voormeeren was started in May 2008. This project, titled "Hybrid Dynamic Substructuring in Wind Turbine Engineering", is a collaboration between Delft University of Technology and Siemens Wind Power A/S and is aimed at advancing dynamic substructuring techniques and applying these to structural dynamic wind turbine problems. The work presented in this MSc. thesis is part of this PhD project.

Dynamic substructuring is a method to obtain the structural dynamic behavior of large and/or complex structures by dividing them into several smaller, simpler substructures (or components) of which the dynamic behavior is generally easier to determine. By using component model reduction techniques the number of degrees of freedom (DoF) of the substructure models are significantly reduced. An accurate and compact set of equations of motion describing the global dynamic behavior is obtained after assembly of these component models. These reduced models can then be used for efficient time simulation, control and/or optimization purposes. Before any dynamic substructuring analyses are possible, a general framework is to be set up and implemented. The thesis assignment is therefore:

"Set up and implement a general framework for the application of dynamic substructuring within wind turbine engineering"

The framework that is set up in this work is shown as the *Dynamic Substructuring Flowchart* (figure 1.1), this flowchart contains the steps that were identified to enable a successful DS analysis. A brief description of each step is given:

1. **Creation of component models**: Obtaining or creating the substructure models is the first step. These can either be full finite element models from commercial finite element (FE) software (ANSYS, NASTRAN, ABAQUS, etc.) or created in MatLab. Another option is to use the frequency response functions obtained from the measurements as experimental substructure models.

2. **Import in MatLab**: Since MatLab is chosen as the software platform, all component models need to be imported in MatLab. The finite element models are

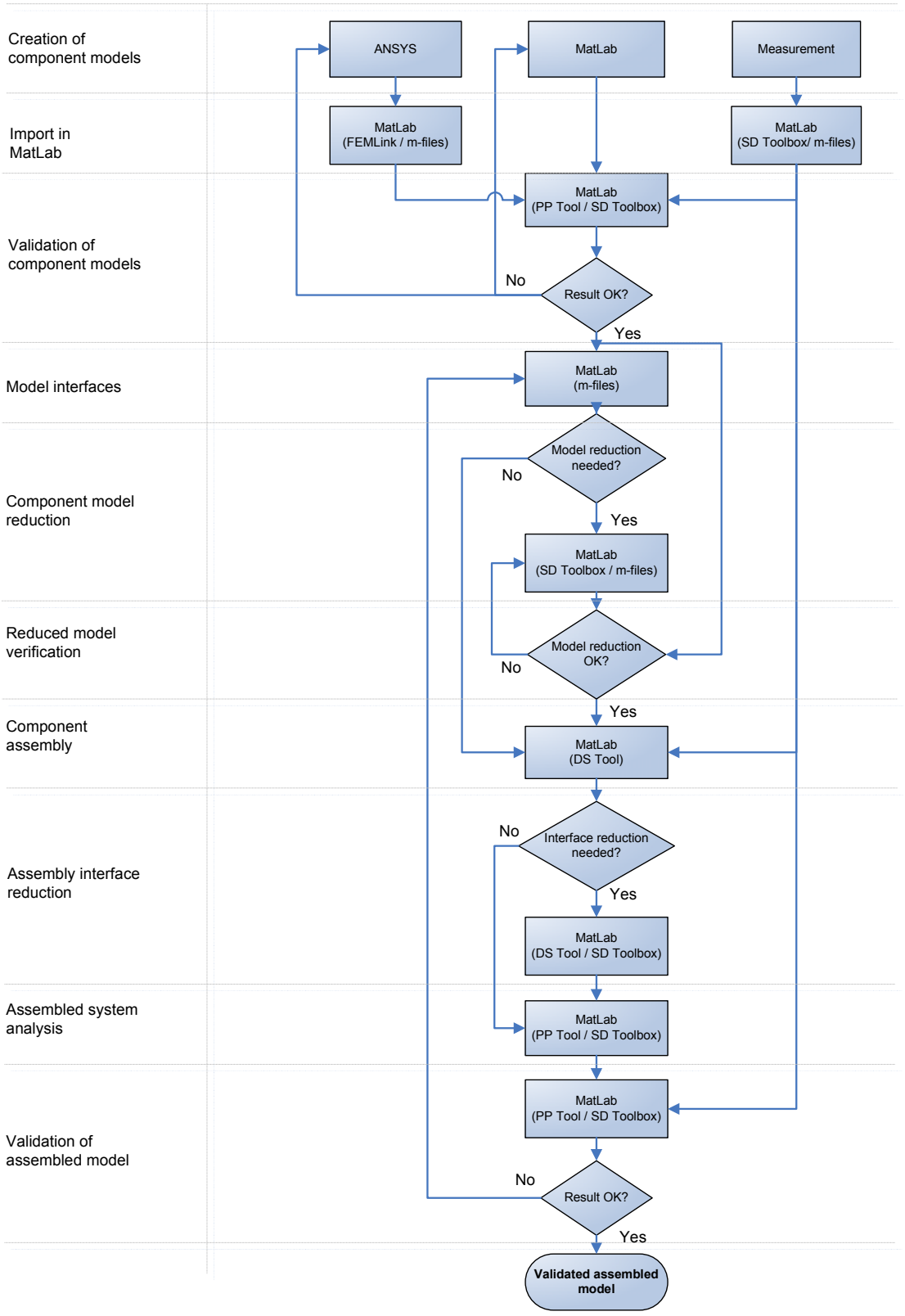


Figure 1.1: Dynamic Substructuring Flowchart

imported using FEMlink and the measurements using the SD Toolbox [3] and their data is sorted according to the data format given in appendix B.

3. **Validation of component models**: In order to create an accurate global model, accurate substructure models are required. Measurements on the components are therefore used to validate the created component models.

4. **Model interfaces**: Just as important as accurate substructure models are accurate interface models. In addition, the interfaces could also contain dynamic effects which need to be included in the global model. These need to be modeled before the assembly step.

5. **Component model reduction**: If needed, the component models can be reduced, using component model reduction methods. Reduction will in general be useful for models which are used for control purposes or when a high number of repetitive computation are performed (such as computing the response for several load cases).

6. **Verification of the reduced model**: After component model reduction, it is worthwhile to check whether the dynamic behavior of the component in still adequately described by the set of generalized degrees of freedom. This is performed by comparing the dynamic behavior of the reduced component with the dynamic behavior full (validated) model.

7. **Component assembly**: In order to obtain the global dynamic model, the (reduced or full) substructures need to be assembled.

8. **Interface reduction**: If the number of (generalized) degrees of freedom of an assembly of (reduced) substructures is too large, interface reduction can be applied. This involves a reduction of the retained interface degrees of freedom.

9. **Assembled system analysis**: In the previous steps a global (reduced) model is built from the (reduced) substructure models. This model can now be used to perform different types of analyses, such as modal analysis, harmonic analysis, time simulation, etc.

10. **Validation of assembled model**: Similar to component model validation, the global model needs to be validated. By performing this validation step, one gains insight in the accuracy of the global model.

The thesis focuses on the steps needed to obtain a reduced and assembled structural dynamic model from the component models (steps 2 to 8). The main subjects are therefore assembly techniques (in the time domain), component model reduction, interface modeling and interface reduction techniques.

1.3 Thesis outline

The work presented in this thesis, is divided into three parts. First of all, Part I treats the theory behind the dynamic substructuring approach and will discuss in detail different assembly, reduction and interface modeling techniques. In Part II the techniques from Part I are applied to the yaw system of a multi megawatt wind turbine. The conclusions and recommendations for further work are given in Part III. Each part consists of a number of chapters:

1.3.1 Part I: Introduction to dynamic substructuring and structural assembly

- **Chapter 2** starts with an introduction of dynamic substructuring. In the remainder of the chapter a general framework for structural assembly is presented.
- **Chapter 3** presents the concept of Component Mode Synthesis (CMS) for model reduction and treats a number of CMS methods: The classical Guyan and Craig-Bampton methods, the relatively new Dual Craig-Bampton and newly developed Mixed Craig-Bampton method.
- **Chapter 4** addresses several methods for interface modeling and assembly of substructures with non-conforming meshes.
- **Chapter 5** addresses the different approaches that can be taken after the assembly step. One can directly use the reduced and assembled substructures for analysis, but one can also create an even compacter system of equations by applying *interface reduction*. Secondly, the Substructure Modal Assurance Criterion (SUMAC) is presented and finally, a test structure is presented and used to demonstrate the reduction and assembly methods.

1.3.2 Part II: Application to a Multi-MW Wind Turbine Yaw System

- **Chapter 6** gives a description of the 2.3 MW Siemens Wind Turbine. Since only the yaw system of the wind turbine is considered in the current dynamic substructuring analysis, the subsystem boundaries are determined and the yaw system components and interfaces are identified and described.
- In **chapter 7** the modeling of the substructures is discussed and a detailed description of the yaw gearbox modeling strategy is shown. The bedplate model is validated by an experimental modal analysis and several measurements are performed on the gearbox in order to obtain a number of unknown parameters.

- In **Chapter 8** a number of dynamic substructuring analyses are performed using the component models of the yaw system. The component models are reduced using the different component model reduction methods described in chapter 3.

1.3.3 Part III: Conclusions and Recommendations

- The conclusions are presented in **Chapter 9**.
- **Chapter 9.2** gives a number of recommendations for future work, on both the theoretical and the practical level.

Part I

Theory of Dynamic Substructuring

CHAPTER 2

Introduction to dynamic substructuring and structural assembly

2.1 The why and how on dynamic substructuring

"Divide and conquer": an ancient tactic often successfully applied in a vast range of domains; from economics to warfare and politics. This knowledge must have triggered Schwarz [4] in 1890 to apply the strategy to a mathematical problem. He divided a complex domain in two simple parts (a circle and a rectangle) in order to find a solution for the associated differential equations of the combined domains. Since the analytical solutions where known in both sub domains, the solutions on the interfaces were used in an iterative way to converge to the solution on the complex domain. This idea of *domain decomposition* can be seen as the ancestor of dynamic substructuring, where the subdomains are in fact the components of the total structure.

It is thus a way to obtain the structural dynamic behavior of large and/or complex structures by dividing them into several smaller, simpler substructures (or components) of which the dynamic behavior is generally easier to determine. The dynamics of the total structure are then obtained by assembling the dynamic models of the components. The developments of these ideas came two decades after the development of the *finite element method* (FEM), which can be traced back to Hrennikoff [5] and Courant [6]. The first ideas for finite element model reduction and dynamic substructuring were published by Hurty [7,8] and Gladwell [9], these methods became known as *Component Mode Synthesis* (CMS). Following Hurty and Gladwell the classical CMS methods were introduced soon after: Craig and Bampton [10] in 1968, Mac Neal [11] in 1971 and Rubin [12] in 1975.

Dynamic substructuring (DS) appeared to be a very useful tool in structural dynamic analysis and quickly gained popularity in the structural engineering society. From the 1980's onward, experimental substructuring caught the attention of the experimental society. Due to much improved measurement hardware and experimental techniques, measurements could now be used in substructuring analyses [13–15].

The substructuring approach to dynamic analysis has several advantages:

- It allows the evaluation of structures that would otherwise be to large and/or complex to be simulated or measured as a whole.
- Experimentally obtained substructures (measurements) can be combined with numerical substructures (FE models), in order to compute the dynamic behavior of

the total structure.

- Local dynamic behavior and its influence on the global behavior can be determined more easily. This allows for local optimization of the design, but also for model simplification by eliminating local subsystem behavior which has no significant impact on the assembled system.
- It allows sharing and combining of substructures from different project groups.

Following the general introduction given above, a brief overview of dynamic substructuring will be given. In general we can identify two domains in which dynamic substructuring can be performed:

- The time domain, where the structural properties in terms of mass, stiffness and damping are used. Within the time domain we can identify two types of substructures:
  - The "physical" substructures, where one describes the substructure in terms of its discretized matrices from geometric distributions of mass, damping and stiffness (full FE models).
  - The "modal" substructures, that are described in terms of generalized (modal) degrees of freedom and their associated reduced matrices.
- The frequency domain, where frequency response functions (FRFs) of the components are assembled.

In addition one can distinguish two types of substructuring; numerical dynamic substructuring and experimental dynamic substructuring. Here "numerical" indicates that a number of numerical substructures (e.g. full FE models or superelements) are assembled. Hence, the analysis is usually done in the time domain or in the frequency domain (by using synthesized FRF's). Numerical DS is a well developed field and widely used in the structural dynamics community; CMS methods like the Guyan and Craig-Bampton methods are integrated in many FE software packages.

One could also encounter components which are difficult to be properly modeled (e.g. a car body and its interior), the component can then be measured and assembled with the remaining numerical substructure models. This approach is referred to as experimental dynamic substructuring and is usually taken if a measurement of the component is easier and/or more efficient than creating a model. This is usually done in the frequency domain using FRF coupling methods.

As already mentioned, structural dynamic analysis is in general performed in two domains; the time and frequency domain. In general we can state that any substructure in the time domain can be coupled to any other substructure in the time domain and any substructure in the frequency domain can be coupled to any other substructure in the frequency domain. This, and the relationships between the different types of substructure models, is visualized in 2.1.

Regardless of whether the substructures are modeled in the time or frequency domain,

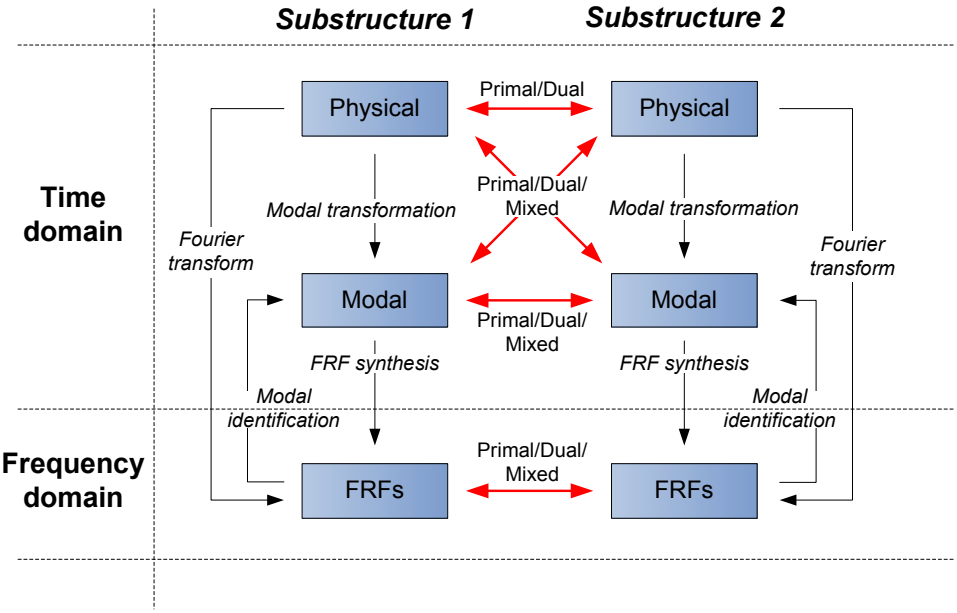


Figure 2.1: Substructures in the time and frequency domain and their possibilities for assembly

two conditions must be satisfied when assembling substructures:

1. Compatibility condition; interface displacements of the substructures must be compatible, i.e. the displacements of both sets of interface DoF must be the same.
2. Equilibrium condition; the forces connecting the substructures' interface degrees of freedom must be in equilibrium, e.g. opposite in direction and equal in magnitude.

Two common assembly methods in finite element modeling which satisfy both conditions are:

- Primal assembly: by choosing a unique set of degrees of freedom (DoF), one set of interface DoF is eliminated. Both substructures thus share the same set of interface DoF and compatibility and equilibrium are both *a priori* satisfied.
- Dual assembly: as stated earlier, the connection forces on both sides of the interface must be in equilibrium. One way of enforcing this is by choosing an unique set of interface forces, which will *a priori* satisfy the equilibrium condition. The compatibility condition is then explicitly added to the set of equations.

In general one can state that the difference between *primal* and *dual* assembly is that in primal assembly a unique set of DoF is found and thereby "merges" both interfaces to one unique interface. Dual assembly, on the other hand, retains all the substructure DoF and uses an additional set of coupling DoF to connect the substructures. In this section the assembly methods will be discussed in more detail for both the time and frequency domain. Other methods for substructure assembly also exist. One will be presented at the end of section 2.2.1 and is in fact a combination of primal and dual assembly.

2.2 Assembly in the time domain

As mentioned before, in the time domain two types of component models exist. A "physical" substructure is described by its mass, damping and stiffness distributions given by the $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{K}$ matrices respectively and the associated displacements ($\boldsymbol{u}$). The equations of motion for substructure s in the total structure are then given as:

$$\boldsymbol{M}^{(s)}\ddot{\boldsymbol{u}}^{(s)} + \boldsymbol{C}^{(s)}\dot{\boldsymbol{u}}^{(s)} + \boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{f}^{(s)} + \boldsymbol{g}^{(s)}. \tag{2.1}$$

Where $\boldsymbol{f}^{(s)}$ are the applied external forces and $\boldsymbol{g}^{(s)}$ are the connecting forces from the neighboring substructures.
A "modal" substructure is described by its generalized (or reduced) mass, damping and stiffness matrices given by the $\tilde{\boldsymbol{M}}$, $\tilde{\boldsymbol{C}}$ and $\tilde{\boldsymbol{K}}$ matrices respectively and the associated set of generalized DoF ($\boldsymbol{q}$). The reduction methods leading to such reduced systems will be discussed in detail in chapter 3.

$$\tilde{\boldsymbol{M}}^{(s)}\ddot{\boldsymbol{q}}^{(s)} + \tilde{\boldsymbol{C}}^{(s)}\dot{\boldsymbol{q}}^{(s)} + \tilde{\boldsymbol{K}}^{(s)}\boldsymbol{q}^{(s)} = \tilde{\boldsymbol{f}}^{(s)} + \tilde{\boldsymbol{g}}^{(s)}. \tag{2.2}$$

For simplicity of notation the superscript $^{(s)}$ will be omitted, and the notations denote substructure equations unless stated otherwise. Although included in (2.1) and (2.2), damping will be neglected in subsequent discussions. However, the techniques and methods described are also applicable to structures with light damping.

In general we can identify two types of interfaces when dealing with reduced substructures, those where the reduced set of DoF still contains the displacements of the interface DoF:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{u}_b \end{bmatrix}, \tag{2.3}$$

and those where all physical DoF are lost and the reduced set of DoF contains only modal amplitudes and interface forces:

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{g}_b \end{bmatrix}. \tag{2.4}$$

This brings us to three possible assembly cases:

- Interface displacements to interface displacements: $\boldsymbol{u}_b \leftrightarrow \boldsymbol{u}_b$
- Interface forces to interface forces: $\boldsymbol{g}_b \leftrightarrow \boldsymbol{g}_b$
- Interface displacements to interface forces: $\boldsymbol{u}_b \leftrightarrow \boldsymbol{g}_b$

The subscript b denotes coupling (boundary) degrees of freedom. In the following subsections each assembly case is discussed.

It is shown in this section that in general any type of substructure in the time domain can be assembled with any other type of substructure in the time domain. This allows for a true "LEGO" approach in dynamic substructuring. With a "LEGO approach" we mean the ability to use independently created substructures in the DS analysis and thereby enabling the best modeling approach for each substructures and the ability to use existing FE models in the analysis.

2.2.1 Assembly using interface displacements: $\boldsymbol{u}_b \leftrightarrow \boldsymbol{u}_b$

The first case is coupling two structures which both have the original set of interface DOF ($\boldsymbol{u}_b$). This could be either full FEM models (so called "physical" substructures) or reduced substructures of which the original set of interface DoF is retained in the set of generalized DoF (see chapter 3). As an example two full FEM models will be assembled

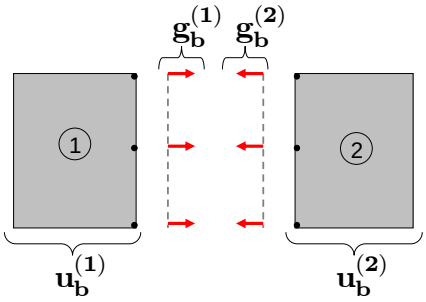


Figure 2.2: Assembly using interface displacements

as shown in figure 2.2. Since the interface DoF are in terms of interface displacements as indicated by the black dots (nodes), the compatibility condition can be satisfied by assembling the interface displacements ($\boldsymbol{u}_b^{(s)}$) on both sides of the interface. In order to be able to satisfy the equilibrium condition, an additional interface force field ($\boldsymbol{g}_b^{(s)}$) is introduced. Which results in the following set of equations (as stated before, the damping is neglected):

$$\begin{bmatrix} \boldsymbol{M}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^{(2)} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}^{(1)} \\ \ddot{\boldsymbol{u}}^{(2)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}^{(2)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{(1)} \\ \boldsymbol{f}^{(2)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}^{(1)} \\ \boldsymbol{g}^{(2)} \end{bmatrix} \tag{2.5}$$

The compatibility condition for assembly writes:

$$\boldsymbol{u}_b^{(1)} = \boldsymbol{u}_b^{(2)} \tag{2.6}$$

The compatibility condition can also be written in a matrix vector form:

$$\boldsymbol{B}\boldsymbol{u} = \boldsymbol{0} \tag{2.7}$$

The $\boldsymbol{B}$ matrix operates on the interface degrees of freedom and is a signed Boolean matrix if the substructure interfaces are conforming (hence for conforming meshes on the interface). Note that in practice the substructures often do not originate from a partitioning of a global mesh but are meshed independently. In that case the interface compatibility is usually enforced through nodal collocation (see section 4.4), so that the compatibility condition can still be written as in (2.7) but now the matrix $\boldsymbol{B}$ is no longer Boolean. The subsequent discussion is valid both when $\boldsymbol{B}$ is Boolean or not. If $\boldsymbol{B}$ is a signed Boolean matrix, the compatibility condition states that any pair of matching interface degrees of freedom $u^{(k)}$ and $u^{(l)}$ must have the same displacement, i.e. $u^{(k)} - u^{(l)} = 0$. More details on the formulation of the Boolean matrix $\boldsymbol{B}$ can be found in appendix A. The second condition, the equilibrium condition, writes:

$$\boldsymbol{g}_b^{(1)} + \boldsymbol{g}_b^{(2)} = \boldsymbol{0} \tag{2.8}$$

The equilibrium condition is expressed in matrix form as:

$$\boldsymbol{L}^T \boldsymbol{g} = \boldsymbol{0} \tag{2.9}$$

where the matrix $\boldsymbol{L}$ is the Boolean matrix localizing the interface DoF of the substructures in the global set of DoF. The expression states that when the connection forces are summed, their resultant must be equal to zero, i.e. $g^{(k)} + g^{(l)} = 0$. More details can be found in appendix A. It can be shown that $\boldsymbol{B}$ and $\boldsymbol{L}$ are in each others null space:

$$\begin{aligned} &\boldsymbol{BL} = \boldsymbol{0} \\ &\boldsymbol{L}^T\boldsymbol{B}^T = \boldsymbol{0} \end{aligned} \tag{2.10}$$

These two conditions enforce compatibility and equilibrium between the two substructures.

The full set of equations is now given by (2.5), (2.7) and (2.9). As described in the first part of this section, the two most common assembly methods will be discussed; primal and dual assembly.

Primal assembly using the interface displacements

In the formulation of (2.5), each substructure is separated in the equations of motion and has its own degrees of freedom. Since each system has a number of boundary DoF, some DoF have multiple entries in the assembled vector $\boldsymbol{u}$. From this set, an unique set of DoF $\bar{\boldsymbol{u}}$ can be found (2.11), that automatically satisfies the compatibility condition.

$$\boldsymbol{u} = \boldsymbol{L}\bar{\boldsymbol{u}} \tag{2.11}$$

Substituting this in the equilibrium condition of (2.9) gives:

$$\boldsymbol{Bu} = \boldsymbol{BL}\bar{\boldsymbol{u}} = \boldsymbol{0}$$

Due to the relation between $\boldsymbol{B}$ and $\boldsymbol{L}$ (2.10), the compatibility condition is satisfied *a priori* by the set of unique DoF. By pre-multiplying the equation of motion (2.5) with $\boldsymbol{L}^T$, the equilibrium condition allows to remove the interface forces $\boldsymbol{g}$ from the equations, since:

$$\boldsymbol{L}^T\boldsymbol{g} = \boldsymbol{0}$$

Thus, by substituting (2.11) into (2.5) and pre-multiplying with $\boldsymbol{L}^T$, the primal assembled system is obtained as:

$$\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{u}}} + \bar{\boldsymbol{K}}\bar{\boldsymbol{u}} = \bar{\boldsymbol{f}}\,, \tag{2.12}$$

where:

$$\begin{cases} \bar{\boldsymbol{M}} = \boldsymbol{L}^T\boldsymbol{M}\boldsymbol{L} \\ \bar{\boldsymbol{K}} = \boldsymbol{L}^T\boldsymbol{K}\boldsymbol{L} \\ \bar{\boldsymbol{f}} = \boldsymbol{L}^T\boldsymbol{f} \end{cases}$$

The matrices $\boldsymbol{M}$ and $\boldsymbol{K}$ are the uncoupled block diagonal mass and stiffness matrices. $\bar{\boldsymbol{M}}$ and $\bar{\boldsymbol{K}}$ are the assembled block diagonal mass and stiffness matrices and thus form the equations of motion of the assembled structure. As can be seen from (2.11) and (2.9), the primal assembly technique thus satisfies both conditions *a priori*. Note that this technique is also used to assemble individual finite elements in a larger finite element model.

Dual assembly using interface displacements

In dual assembly the full set of degrees of freedom $\boldsymbol{u}$ is retained, meaning the interface DoF are present as many times as there are substructures connected to the corresponding node. Interface forces are chosen in the form of:

$$\boldsymbol{g} = -\boldsymbol{B}^T\boldsymbol{\lambda}. \tag{2.13}$$

The interface forces are now described by the Lagrange multipliers , $\boldsymbol{\lambda}$, which are the interface force intensities . Due to the construction of $\boldsymbol{B}$ (see appendix A), the interface forces on both sides of the connection will always be opposite and equal, so equilibrium on the interface is *a priori* satisfied. This can be shown by substituting (2.13) into 2.9.

$$-\boldsymbol{L}^T\boldsymbol{B}^T\boldsymbol{\lambda} = \boldsymbol{0}$$

The system of equations (2.5), (2.9) and (2.7) now reduces to:

$$\begin{cases} \boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} + \boldsymbol{B}^T\boldsymbol{\lambda} = \boldsymbol{f} \\ \boldsymbol{B}\boldsymbol{u} = \boldsymbol{0} \end{cases} \tag{2.14}$$

In this equation $\boldsymbol{M}$ and $\boldsymbol{K}$ are block-diagonal matrices where each block represents the mass, damping or stiffness matrix of a substructures. In matrix form one can write:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \tag{2.15}$$

The main difference between primal and dual assembly of interface displacements is that in primal assembly the compatibility condition is satisfied *a priori*, whereas in dual assembly the equilibrium is satisfied *a priori*. Physically this can be interpreted as assembly by interface displacements or interface forces, respectively.

A mix of both: Dirichlet to Neumann assembly

In addition to the previously described primal and dual assembly, a method which employs a mix of both also exists. Assume we want to couple two subsystems, denoted 1 and 2, both described by (2.1) (for simplicity we neglect the damping). The required Boolean matrices $\boldsymbol{B}$ and $\boldsymbol{L}$ are already determined in the previous section, thus the set of equations writes:

$$\begin{cases} \boldsymbol{M}^{(1)}\ddot{\boldsymbol{u}}^{(1)} + \boldsymbol{K}^{(1)}\boldsymbol{u}^{(1)} + \boldsymbol{B}^{(1)^T}\boldsymbol{\lambda} = \boldsymbol{f}^{(1)} \\ \boldsymbol{M}^{(2)}\ddot{\boldsymbol{u}}^{(2)} + \boldsymbol{K}^{(2)}\boldsymbol{u}^{(2)} + \boldsymbol{B}^{(2)^T}\boldsymbol{\lambda} = \boldsymbol{f}^{(2)} \\ \boldsymbol{B}^{(1)}\boldsymbol{u}^{(1)} + \boldsymbol{B}^{(2)}\boldsymbol{u}(2) = \boldsymbol{0} \end{cases} \tag{2.16}$$

Equilibrium is enforced through the $\boldsymbol{\lambda}$'s (2.13), as we can see, (2.16) is identical to the dual assembled system (2.14). The disadvantage of the dual system is that the boundary DoF are present multiple times in the DoF vector $\boldsymbol{u}$. We could reduce this set $\boldsymbol{u}$ to a set of unique DoF $\bar{\boldsymbol{u}}$, similar to what is done in primal assembly (2.17):

$$\begin{bmatrix} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}^{(1)} \\ \boldsymbol{L}^{(2)} \end{bmatrix} \bar{\boldsymbol{u}} \; \boldsymbol{L}\bar{\boldsymbol{u}} \tag{2.17}$$

Substituting (2.17) into (2.18), gives the final set of equations.

$$\begin{cases} \boldsymbol{M}^{(1)}\boldsymbol{L}^{(1)}\ddot{\bar{\boldsymbol{u}}} + \boldsymbol{K}^{(1)}\boldsymbol{L}^{(1)}\bar{\boldsymbol{u}} + \boldsymbol{B}^{(1)^T}\boldsymbol{\lambda} = \boldsymbol{f}^{(1)} \\ \boldsymbol{M}^{(2)}\boldsymbol{L}^{(2)}\ddot{\bar{\boldsymbol{u}}} + \boldsymbol{K}^{(2)}\boldsymbol{L}^{(2)}\bar{\boldsymbol{u}} + \boldsymbol{B}^{(2)^T}\boldsymbol{\lambda} = \boldsymbol{f}^{(2)} \end{cases} \tag{2.18}$$

By substituting (2.17) into (2.18), the last line of the equation would write:

$$\boldsymbol{B}^{(1)}\boldsymbol{L}^{(1)}\bar{\boldsymbol{u}} + \boldsymbol{B}^{(2)}\boldsymbol{L}^{(2)}\bar{\boldsymbol{u}} = \boldsymbol{B}\boldsymbol{L}\bar{\boldsymbol{u}} = \boldsymbol{0}$$

Hence, the compatibility condition is automatically satisfied, since $\boldsymbol{L}$ is in the nullspace of $\boldsymbol{B}$ (2.10), and the third line drops out of the set of equations. If we write the final equations in a matrix-vector formulation, we obtain:

$$\begin{bmatrix} \boldsymbol{M}^{(1)}\boldsymbol{L}^{(1)} & \boldsymbol{0} \\ \boldsymbol{M}^{(2)}\boldsymbol{L}^{(2)} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\bar{\boldsymbol{u}}} \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}^{(1)}\boldsymbol{L}^{(1)} & \boldsymbol{B}^{(1)^T} \\ \boldsymbol{K}^{(2)}\boldsymbol{L}^{(2)} & \boldsymbol{B}^{(2)^T} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{u}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{(1)} \\ \boldsymbol{f}^{(2)} \end{bmatrix} \tag{2.19}$$

The name of this type of assembly, "Dirichlet to Neumann", refers to the way the assembled system can be solved (using the Gauss-Seidel method) and not to the way the systems are assembled [16]. A disadvantage of this method is that the obtained matrices are no longer symmetric and diagonally sparse, despite the sub-matrices still being sparse.

2.2.2 Assembly using interface forces: $g_b \leftrightarrow g_b$

The second possible case when assembling substructures is that both sets of interface DoF consist of only interface forces ($\boldsymbol{g}_b$) as can be seen in figure 2.3. Here, the interface DoF are

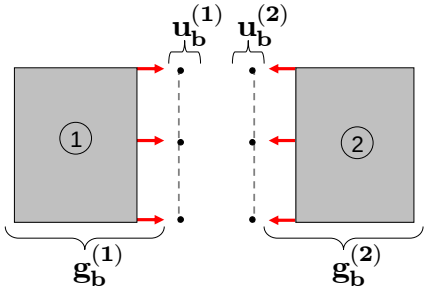


Figure 2.3: Assembly using interface forces

in terms of interface forces as indicated by the red arrows, the equilibrium condition can

be satisfied by assembling the interface forces ($\boldsymbol{g}_b^{(s)}$) on both sides of the interface. In order to be able to satisfy the compatibility condition, an additional interface displacement field ($\boldsymbol{u}_b^{(s)}$) is introduced.

This could be the case if we want to assemble two *dual* reduced systems (see section 3.6). By assuming no external forces and explicitly splitting the set of DoF according to (2.4), we obtain the following set of uncoupled equations:

$$\begin{bmatrix} \tilde{\boldsymbol{M}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{M}}^{(2)} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}^{(1)} \\ \ddot{\boldsymbol{g}}_b^{(1)} \\ \ddot{\boldsymbol{\eta}}^{(2)} \\ \ddot{\boldsymbol{g}}_b^{(2)} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{K}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}^{(2)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_\eta^{(1)} \\ \tilde{\boldsymbol{f}}_b^{(1)} \\ \boldsymbol{f}_\eta^{(2)} \\ \tilde{\boldsymbol{f}}_b^{(2)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_b^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{u}_b^{(2)} \end{bmatrix} \tag{2.20}$$

Since the interfaces of the substructures are now described in terms of *flexibility* instead of *stiffness*, the excitations are also in terms of interface *displacements* instead of interface *forces*. This means the part of the reduced stiffness matrix that acts on the interface DoF ($\tilde{\boldsymbol{K}}_{bb}$) is in fact a *flexibility* matrix. The excitation of the interface is therefore an interface displacement ($\boldsymbol{u}_b$), as can be seen in (2.20). The external forces $\boldsymbol{f}_\eta^{(s)}$ result from a projection of the ("internal" part of the) reduction basis ($\boldsymbol{R}_i$) on the applied (internal) forces $\boldsymbol{f}_i^{(s)}$. The external excitations ($\tilde{\boldsymbol{f}}_b^{(s)}$) are in fact the interface displacements due to the applied external load $\boldsymbol{f}_b^{(s)}$ on the interface, according to:

$$\tilde{\boldsymbol{f}}_b^{(s)} = -\tilde{\boldsymbol{K}}_{bb}\boldsymbol{f}_b^{(s)}$$

In order to assemble the two substructures, the conditions of compatibility and equilibrium have to be satisfied as before.

$$\boldsymbol{u}_b^{(1)} - \boldsymbol{u}_b^{(2)} = \boldsymbol{0}$$
$$\boldsymbol{g}_b^{(1)} + \boldsymbol{g}_b^{(2)} = \boldsymbol{0}$$

These two conditions once again enable two types of assembly; primal and dual assembly. In this case the interface forces will be assembled instead of the interface displacements.

Primal assembly using the interface forces

In the formulation of (2.20), each substructure is separated in the equations of motion and has its own degrees of freedom. Since each system has a number of interface forces, the interface forces have multiple entries in the assembled vector $\boldsymbol{q}$. Recalling from the dual assembly in section 2.2.1, the interface forces can be written according to:

$$\boldsymbol{g}^{(s)} = -\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}$$

Due to the equilibrium condition, the sets of interface forces can be written in terms of the (unique) set of interface force intensities ($\boldsymbol{\lambda}$). This allows for a unique set of DoF to be found, as:

$$\begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \begin{bmatrix} -\boldsymbol{B}^{(1)^T} \end{bmatrix} \\ \boldsymbol{0} & \boldsymbol{0} & \\ \boldsymbol{0} & \boldsymbol{I} & \begin{bmatrix} -\boldsymbol{B}^{(2)^T} \end{bmatrix} \\ \boldsymbol{0} & \boldsymbol{0} & \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \tilde{\boldsymbol{L}}\bar{\boldsymbol{q}} \tag{2.21}$$

$\tilde{\boldsymbol{L}}$ is similar to $\boldsymbol{L}$ given in section 2.2.1 and is used to reduce the uncoupled set of DoF to a set of unique (coupled) set of DoF. Since the equilibrium condition states that the connection forces on both sides of the interface have to be equal and *opposite*; $\boldsymbol{B}^{(1)}$ and $\boldsymbol{B}^{(2)}$ are signed differently. Substituting (2.21) into (2.20) gives.

$$\begin{bmatrix} \tilde{\boldsymbol{M}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{M}}^{(2)} \end{bmatrix} \tilde{\boldsymbol{L}} \begin{bmatrix} \ddot{\boldsymbol{\eta}}^{(1)} \\ \ddot{\boldsymbol{\eta}}^{(2)} \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{K}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}^{(2)} \end{bmatrix} \tilde{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{\eta}^{(1)} \\ \tilde{\boldsymbol{f}}_{b}^{(1)} \\ \boldsymbol{f}_{\eta}^{(2)} \\ \tilde{\boldsymbol{f}}_{b}^{(2)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(2)} \end{bmatrix}$$

Pre-multiplying this result with $\tilde{\boldsymbol{L}}^T$ gives on the right side of the equation:

$$\tilde{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \left[-\boldsymbol{B}^{(1)}\right] & & \left[-\boldsymbol{B}^{(2)}\right] & \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(2)} \end{bmatrix} = \boldsymbol{0}$$

By comparing this with (2.7), it is clear that this will always ensure compatibility.

$$\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{q}}} + \bar{\boldsymbol{K}}\bar{\boldsymbol{q}} = \bar{\boldsymbol{f}} \tag{2.22}$$

where:

$$\begin{cases} \bar{\boldsymbol{M}} = \tilde{\boldsymbol{L}}^T \tilde{\boldsymbol{M}} \tilde{\boldsymbol{L}} \\ \bar{\boldsymbol{K}} = \tilde{\boldsymbol{L}}^T \tilde{\boldsymbol{K}} \tilde{\boldsymbol{L}} \\ \bar{\boldsymbol{f}} = \tilde{\boldsymbol{L}}^T \boldsymbol{f}_R \\ \bar{\boldsymbol{q}} = \begin{bmatrix} \boldsymbol{\eta}^{(1)} & \boldsymbol{\eta}^{(2)} & \boldsymbol{\lambda} \end{bmatrix}^T \end{cases}$$

$\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{K}}$ denote the block diagonal mass and stiffness matrices containing the substructure mass and stiffness matrices. As is clear from (2.22), the set of DoF contains a unique field of interface forces ($\boldsymbol{\lambda}$), that act on both substructures. The result is thus similar to the dual assembly using interface displacements (section 2.2.1).

Dual assembly using the interface forces

Besides primal assembly using the interface forces, one can also assemble the substructures in a dual fashion. The set of equations given in (2.20) is again the starting point for the assembly method presented here. In the section describing dual assembly using interface displacements (section 2.2.1), all the DoF are retained and the assembly is done by introducing a unique set of interface forces (2.13). When performing a dual assembly using the interface forces, the full set of DoF is retained (and thus also the multiplicity in the interface forces) and a unique set of interface *displacements* is introduced. Due to the compatibility condition, the displacements on both sides of the interface can be written as a function of a unique set of interface displacements ($\boldsymbol{u}_b$).

$$\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{u}_{b}^{(2)} \end{bmatrix} = -\tilde{\boldsymbol{B}}^T \boldsymbol{u}_b. \tag{2.23}$$

Note that this is similar to the approach taken in dual assembly of the interface displacements (section 2.2.1), though in section 2.2.1 *equilibrium* was *a priori* satisfied, whereas here *compatibility* is *a priori* satisfied. This leads to a difference in construction of the Boolean matrices; when satisfying equilibrium *a priori* (as in dual assembly in section 2.2.1: $\boldsymbol{g} = -\boldsymbol{B}^T\boldsymbol{\lambda}$) the Boolean matrix $\boldsymbol{B}$ will be signed (since interface forces act in opposite directions), but when satisfying compatibility *a priori* (2.23) the Boolean matrix $\tilde{\boldsymbol{B}}$ will be unsigned (since both interfaces will have the same displacements). It can be derived that $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{L}}$ (2.21) are in each others nullspace:

$$\begin{aligned} \tilde{\boldsymbol{B}}\tilde{\boldsymbol{L}} &= \boldsymbol{0} \\ \tilde{\boldsymbol{L}}^T\tilde{\boldsymbol{B}}^T &= \boldsymbol{0} \end{aligned} \tag{2.24}$$

By substituting (2.23) into (2.20), the set of equations now writes:

$$\begin{bmatrix} \tilde{\boldsymbol{M}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{M}}^{(2)} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}^{(1)} \\ \ddot{\boldsymbol{g}}_b^{(1)} \\ \ddot{\boldsymbol{\eta}}^{(2)} \\ \ddot{\boldsymbol{g}}_b^{(2)} \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{K}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}^{(2)} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} + \tilde{\boldsymbol{B}}^T\boldsymbol{u}_b = \begin{bmatrix} \boldsymbol{f}_\eta^{(1)} \\ \tilde{\boldsymbol{f}}_b^{(1)} \\ \boldsymbol{f}_\eta^{(2)} \\ \tilde{\boldsymbol{f}}_b^{(2)} \end{bmatrix} \tag{2.25}$$

As the compatibility condition is now satisfied, we still need to satisfy the equilibrium condition:

$$\tilde{\boldsymbol{B}}\boldsymbol{q} = \tilde{\boldsymbol{B}} \begin{bmatrix} \boldsymbol{\eta}^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} = \boldsymbol{g}^{(1)} + \boldsymbol{g}^{(2)} = \boldsymbol{0} \tag{2.26}$$

Since $\tilde{\boldsymbol{B}}$ is a unsigned Boolean matrix, this matrix is used to enforce the equilibrium condition. The dynamic equation (2.25) and the equilibrium condition of (2.26) leads to the dual assembled set of substructures:

$$\begin{cases} \tilde{\boldsymbol{M}}\ddot{\boldsymbol{q}} + \tilde{\boldsymbol{K}}\boldsymbol{q} + \tilde{\boldsymbol{B}}^T\boldsymbol{u}_b = \boldsymbol{f}_R \\ \tilde{\boldsymbol{B}}\boldsymbol{q} = \boldsymbol{0} \end{cases}$$

In matrix form one can write:

$$\begin{bmatrix} \tilde{\boldsymbol{M}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \begin{bmatrix} \tilde{\boldsymbol{K}} & \tilde{\boldsymbol{B}}^T \\ \tilde{\boldsymbol{B}} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_R \\ \boldsymbol{0} \end{bmatrix} \tag{2.27}$$

Here $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{K}}$ are now block diagonal matrices containing the individual substructure matrices. As can be seen we have introduced an extra displacement field ($\boldsymbol{u}_b$) in order to assemble the neighboring interface forces. It is clear that this is not the most efficient way of assembling substructures using interface forces; each interface DoF now has three entries in the DoF vector ($\boldsymbol{q}$), whereas in the DoF vector resulting from the primal assembly each interface DoF is present only once.

2.2.3 Mixed interface assembly: $\boldsymbol{u}_b \leftrightarrow \boldsymbol{g}_b$

In addition to the above discussed coupling of interface displacements to neighboring interface displacements ($\boldsymbol{u}_b \leftrightarrow \boldsymbol{u}_b$) and the assembly of substructures using interface forces ($\boldsymbol{g}_b \leftrightarrow \boldsymbol{g}_b$), we could also encounter cases were an interface *displacement* will have to be assembled with an interface *force* ($\boldsymbol{u}_b \leftrightarrow \boldsymbol{g}_b$) as can be seen in figure 2.4. Since both interface DoF now represent different physical quantities, we cannot directly couple the substructures using primal or dual assembly. Here, the interface DoF of substructure

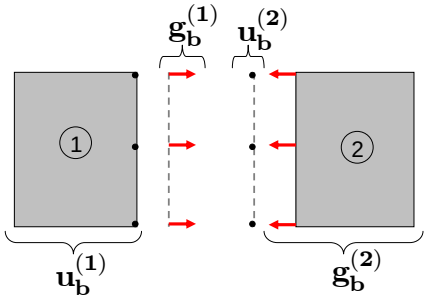


Figure 2.4: Mixed interface assembly

1 are in terms of interface displacements and the interface DoF of substructure 2 are interface forces. In order to be able to satisfy the equilibrium condition an additional interface force field is introduced for substructure 1. Furthermore, an additional interface displacement field is introduced for substructure 2, this enables one to satisfy the compatibility condition.

For the sake of illustration, suppose we want to couple a "physical" substructure to a "dual" reduced substructure (section 3.6). The set of equations would then write:

$$\begin{cases} \boldsymbol{M}^{(1)}\ddot{\boldsymbol{u}}^{(1)} + \boldsymbol{K}^{(1)}\boldsymbol{u}^{(1)} = \boldsymbol{f}^{(1)} + \boldsymbol{g}^{(1)} \\ \tilde{\boldsymbol{M}}^{(2)}\ddot{\boldsymbol{q}}^{(2)} + \tilde{\boldsymbol{K}}^{(2)}\boldsymbol{q}^{(2)} = \tilde{\boldsymbol{f}}^{(2)} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_b^{(2)} \end{bmatrix} \end{cases} \tag{2.28}$$

The matrices denoted by a tilde are matrices associated with the reduced substructure. Writing the equations in a block matrix form and explicitly splitting the set of substructure DoF

$$\boldsymbol{u}^{(1)} = \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b^{(1)} \end{bmatrix}, \quad \boldsymbol{q}^{(2)} = \begin{bmatrix} \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix}$$

gives:

$$\begin{bmatrix} \begin{bmatrix} \boldsymbol{M}_{ii}^{(1)} & \boldsymbol{M}_{ib}^{(1)} \\ \boldsymbol{M}_{bi}^{(1)} & \boldsymbol{M}_{bb}^{(1)} \end{bmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \begin{bmatrix} \tilde{\boldsymbol{M}}_{\eta\eta}^{(2)} & \tilde{\boldsymbol{M}}_{\eta g}^{(2)} \\ \tilde{\boldsymbol{M}}_{g\eta}^{(2)} & \tilde{\boldsymbol{M}}_{gg}^{(2)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i^{(1)} \\ \ddot{\boldsymbol{u}}_b^{(1)} \\ \ddot{\boldsymbol{\eta}}^{(2)} \\ \ddot{\boldsymbol{g}}_b^{(2)} \end{bmatrix} + \dots \\ \dots \begin{bmatrix} \begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} \\ \boldsymbol{K}_{bi}^{(1)} & \boldsymbol{K}_{bb}^{(1)} \end{bmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \begin{bmatrix} \tilde{\boldsymbol{K}}_{\eta\eta}^{(2)} & \tilde{\boldsymbol{K}}_{\eta g}^{(2)} \\ \tilde{\boldsymbol{K}}_{g\eta}^{(2)} & \tilde{\boldsymbol{K}}_{gg}^{(2)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(1)} \\ \boldsymbol{f}_\eta^{(2)} \\ \tilde{\boldsymbol{f}}_b^{(2)} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{u}_b^{(2)} \end{bmatrix} \end{bmatrix} \tag{2.29}$$

The external excitations $\tilde{\boldsymbol{f}}_b^{(2)}$ and $\boldsymbol{f}_\eta^{(2)}$ are similar to the excitations in (2.20). It can be seen that $\tilde{\boldsymbol{K}}_{gg}^{(2)}$ is a flexibility matrix, instead of a stiffness matrix, With the compatibility and equilibrium conditions:

$$\begin{aligned} \boldsymbol{u}_b^{(1)} - \boldsymbol{u}_b^{(2)} &= \boldsymbol{0} \\ \boldsymbol{g}_b^{(1)} + \boldsymbol{g}_b^{(2)} &= \boldsymbol{0} \end{aligned}$$

By taking a quick look at (2.29), it is clear we cannot satisfy the compatibility and the equilibrium condition straightforward, since the interface of substructure 1 is in terms of displacements and interface 2 in terms of forces. But from (2.29) it is clear that the interface displacements (in green) and forces (in red) for both interfaces are present in the set of equations. In order to be able to enable assembly, the interface displacements and the connection forces of both substructures need to be in the global DoF vector. This is achieved by bringing the vector containing the resulting coupling forces and displacements to the left of the equation and including them into the current set of DoF:

$$\begin{bmatrix} \begin{bmatrix} \boldsymbol{M}_{ii}^{(1)} & \boldsymbol{M}_{ib}^{(1)} \\ \boldsymbol{M}_{bi}^{(1)} & \boldsymbol{M}_{bb}^{(1)} \end{bmatrix} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \begin{bmatrix} \tilde{\boldsymbol{M}}_{\eta\eta}^{(2)} & \tilde{\boldsymbol{M}}_{\eta g}^{(2)} \\ \tilde{\boldsymbol{M}}_{g\eta}^{(2)} & \tilde{\boldsymbol{M}}_{gg}^{(2)} \end{bmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i^{(1)} \\ \ddot{\boldsymbol{u}}_b^{(1)} \\ \ddot{\boldsymbol{g}}_b^{(1)} \\ \ddot{\boldsymbol{\eta}}^{(2)} \\ \ddot{\boldsymbol{g}}_b^{(2)} \\ \ddot{\boldsymbol{u}}_b^{(2)} \end{bmatrix} + \dots \\ \dots \begin{bmatrix} \begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} \\ \boldsymbol{K}_{bi}^{(1)} & \boldsymbol{K}_{bb}^{(1)} \end{bmatrix} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \begin{bmatrix} \tilde{\boldsymbol{K}}_{\eta\eta}^{(2)} & \tilde{\boldsymbol{K}}_{\eta g}^{(2)} \\ \tilde{\boldsymbol{K}}_{g\eta}^{(2)} & \tilde{\boldsymbol{K}}_{gg}^{(2)} \end{bmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \\ \boldsymbol{u}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{f}_\eta^{(2)} \\ \tilde{\boldsymbol{f}}_b^{(2)} \\ \boldsymbol{0} \end{bmatrix} \end{bmatrix} \tag{2.30}$$

There is now a straightforward approach to integrate the two conditions into the set of equations, which is similar to the Dirichlet to Neumann assembly discussed previously. From (2.30) it can be seen that both the full set of connection forces and the full set of interface displacements is present in the DoF vector. Recalling from section 2.2.1, the interface forces can be written according to:

$$\boldsymbol{g}^{(s)} = -\boldsymbol{B}^{(s)}\boldsymbol{\lambda}$$

And recalling from section 2.2.2, the interface displacements can be written as:

$$\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_b^{(s)} \end{bmatrix} = -\tilde{\boldsymbol{B}}^{(s)^T}\boldsymbol{u}_b$$

A unique set of DoF can thus be found by incorporating these relations:

$$\begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{g}_b^{(2)} \\ \boldsymbol{u}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \hat{\boldsymbol{L}}\bar{\boldsymbol{q}} \tag{2.31}$$

It is clear that by choosing this unique set of DoF both the compatibility and the equilibrium conditions are always satisfied. By substituting (2.31) into (2.29), the assembled system of equations is obtained.

$$\begin{bmatrix} \begin{bmatrix} \boldsymbol{M}_{ii}^{(1)} & \boldsymbol{M}_{ib}^{(1)} \\ \boldsymbol{M}_{bi}^{(1)} & \boldsymbol{M}_{bb}^{(1)} \end{bmatrix} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \begin{bmatrix} \tilde{\boldsymbol{M}}_{\eta\eta}^{(2)} & \tilde{\boldsymbol{M}}_{\eta g}^{(2)} \\ \tilde{\boldsymbol{M}}_{g\eta}^{(2)} & \tilde{\boldsymbol{M}}_{gg}^{(2)} \end{bmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \hat{\boldsymbol{L}} \begin{bmatrix} \ddot{\boldsymbol{u}}_i^{(1)} \\ \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{\eta}}^{(2)} \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \cdots$$

$$\cdots \begin{bmatrix} \begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} \\ \boldsymbol{K}_{bi}^{(1)} & \boldsymbol{K}_{bb}^{(1)} \end{bmatrix} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \begin{bmatrix} \tilde{\boldsymbol{K}}_{\eta\eta}^{(2)} & \tilde{\boldsymbol{K}}_{\eta g}^{(2)} \\ \tilde{\boldsymbol{K}}_{g\eta}^{(2)} & \tilde{\boldsymbol{K}}_{gg}^{(2)} \end{bmatrix} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \hat{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \tilde{\boldsymbol{f}}_b^{(2)} \\ \boldsymbol{0} \end{bmatrix} \tag{2.32}$$

This assembly technique satisfies both compatibility and equilibrium *a priori* and requires no pre-multiplication by $\tilde{\boldsymbol{L}}^T$. One can thus compare this method to the Dirichlet to Neumann assembly (section 2.2.1), where the set of assembled DoF also contains both

a unique set of interface displacements and a unique set of interface forces. However, pre-multiplication by $\tilde{\boldsymbol{L}}^T$ is beneficial, since it reduces the size of the assembled mass and stiffness matrix and leads to symmetric matrices. In other words, pre- and post multiplication using $\tilde{\boldsymbol{L}}$ rearranges the rows and columns of the block diagonal mass and stiffness matrix to a more compact, square and *symmetric* form, namely:

$$\begin{aligned}
&\begin{bmatrix} \boldsymbol{M}_{ii}^{(1)} & \boldsymbol{M}_{ib}^{(1)} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{M}_{bi}^{(1)} & \boldsymbol{M}_{bb}^{(1)} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{M}}_{\eta\eta}^{(2)} & \tilde{\boldsymbol{M}}_{\eta g}^{(2)} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{M}}_{g\eta}^{(2)} & \tilde{\boldsymbol{M}}_{gg}^{(2)} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i^{(1)} \\ \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{\eta}}^{(2)} \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \cdots \\
&\cdots \begin{bmatrix} \boldsymbol{K}_{ii}^{(1)} & \boldsymbol{K}_{ib}^{(1)} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{bi}^{(1)} & \boldsymbol{K}_{bb}^{(1)} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{0} & \tilde{\boldsymbol{K}}_{\eta\eta}^{(2)} & -\tilde{\boldsymbol{K}}_{\eta g}^{(2)} \\ \boldsymbol{0} & \boldsymbol{I} & -\tilde{\boldsymbol{K}}_{g\eta}^{(2)} & \tilde{\boldsymbol{K}}_{gg}^{(2)} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i^{(1)} \\ \boldsymbol{u}_b \\ \boldsymbol{\eta}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i^{(1)} \\ \boldsymbol{f}_b^{(1)} \\ \boldsymbol{f}_\eta^{(2)} \\ \tilde{\boldsymbol{f}}_b^{(2)} \end{bmatrix}
\end{aligned} \tag{2.33}$$

2.3 Assembly in the frequency domain

When performing an experimental substructuring analysis, at least one of the substructures is obtained from measurements. Often this measured component is expressed in its frequency response functions (FRFs) for several inputs and outputs and no full system description (in terms of physical matrices $\boldsymbol{M}^{(s)}$, $\boldsymbol{C}^{(s)}$ and $\boldsymbol{K}^{(s)}$) is available. This is no problem however, since the FRFs contain all the information of mass, damping and stiffness.The equations of motion are now written in the frequency domain.

$$\boldsymbol{Z}(j\omega)\boldsymbol{u}(j\omega) = \boldsymbol{f}(j\omega) + \boldsymbol{g}(j\omega)$$

Where $\boldsymbol{Z}(j\omega)$ is the (block diagonal) dynamic stiffness matrix containing the dynamic stiffness matrices of all the substructures. The dynamic stiffness is obtained by taking the Fourier transform of the equations of motion in the time domain. This gives:

$$\boldsymbol{Z}(j\omega) = -\omega^2\boldsymbol{M} + j\omega\boldsymbol{C} + \boldsymbol{K} \tag{2.34}$$

In order to couple the substructures the compatibility and equilibrium conditions have to be satisfied, similar to coupling in the time domain. This will give the following set of equations:

$$\begin{cases} \boldsymbol{Z}(j\omega)\boldsymbol{u}(j\omega) = \boldsymbol{f}(j\omega) + \boldsymbol{g}(j\omega) \\ \boldsymbol{B}\boldsymbol{u}(j\omega) = \boldsymbol{0} \\ \boldsymbol{L}^T\boldsymbol{g}(j\omega) = \boldsymbol{0} \end{cases} \tag{2.35}$$

In measurements one usually excites the system with a force and measures the resulting response (displacement, velocity or accelerations). So, one directly measures the dynamic flexibility $\boldsymbol{Y}(j\omega)$ instead of stiffness, which is the inverse of the dynamic stiffness (2.36).

$$\boldsymbol{Y}(j\omega) \hat{=} \boldsymbol{Z}^{-1}(j\omega) \tag{2.36}$$

The size of $\boldsymbol{Y}(j\omega)$ depends on the number of inputs and outputs of the measurement. The possible methods of assembly for coupling substructures in the frequency domain are the same as assembly in the time domain. Although in this section only the basic primal and dual assembly are discussed, one can imagine that by using the techniques of section 2.2 it is also possible to assembly flexibility FRF's directly to dynamic stiffness FRF's. This avoids computing the inverse of one of the two FRF's and thereby also benefits the accuracy of the obtained assembly.

2.3.1 Primal assembly in the frequency domain

Analog to primal assembly in the time domain (section 2.2) a set of unique DoF ($\boldsymbol{q}$) is chosen:

$$\boldsymbol{u}(j\omega) = \boldsymbol{L}\boldsymbol{q}(j\omega) \tag{2.37}$$

By substituting (2.37) into (2.35) and since $\boldsymbol{L}^T$ and $\boldsymbol{B}^T$ are in each others nullspace, the compatibility equation is automatically satisfied. By pre-multiplying the set of equations by $\boldsymbol{L}^T$, the primal assembled system is obtained.

$$\bar{\boldsymbol{Z}}(j\omega)\boldsymbol{q}(j\omega) = \bar{\boldsymbol{f}}(j\omega) \tag{2.38}$$

where:

$$\begin{cases} \bar{\boldsymbol{Z}} = \boldsymbol{L}^T\boldsymbol{Z}\boldsymbol{L} \\ \bar{\boldsymbol{f}} = \boldsymbol{L}^T\boldsymbol{f} \end{cases}$$

The dynamic stiffness matrix of the assembled system is denoted by $\bar{\boldsymbol{Z}}$. The explicit frequency dependence has been omitted for clarity.

2.3.2 Dual assembly in the frequency domain

Analog to dual assembly in the time domain (section 2.2), the set of equations given in (2.35) is assembled.

$$\begin{cases} \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{B}^T\boldsymbol{\lambda} = \boldsymbol{f} \\ \boldsymbol{B}\boldsymbol{u} = \boldsymbol{0} \end{cases} \tag{2.39}$$

The Boolean matrices $\boldsymbol{B}$ and $\boldsymbol{L}$ are constructed identical to the ones used in section 2.2 (see appendix A). The equations shown in (2.39), can be rewritten from a dynamic equation in terms of stiffness to one in terms of flexibility.

$$\begin{cases} \boldsymbol{u} = \boldsymbol{Y}\boldsymbol{f} - \boldsymbol{Y}\boldsymbol{B}^T\boldsymbol{\lambda} \\ \boldsymbol{B}\boldsymbol{u} = \boldsymbol{0} \end{cases}$$

The first equation can be substituted in the compatibility condition (the second equation):

$$\boldsymbol{BYf} - \boldsymbol{BYB}^T\boldsymbol{\lambda} = \boldsymbol{0}$$

The first term ($\boldsymbol{BYf}$) describes the deformation of the (separate) interfaces due to the external forces, thereby resulting in a gap between the substructures. The second term ($\boldsymbol{BYB}^T\boldsymbol{\lambda}$) describes the interface deformations due to the resulting interface forces and "closes" the gap created by the external forces. The interface forces ($\boldsymbol{\lambda}$) can thus be written as:

$$\boldsymbol{\lambda} = \left(\boldsymbol{BYB}^T\right)^{-1}\boldsymbol{BYf}$$

By substituting this expression into the equations of motion (first equation), a direct and convenient expression for the dual assembled system is obtained.

$$\boldsymbol{u} = \boldsymbol{Yf} - \boldsymbol{YB}^T\left(\boldsymbol{BYB}^T\right)^{-1}\boldsymbol{BYf} \tag{2.40}$$

The obtained set of equations is known as Lagrange Multiplier Frequency Based Substructuring (LM FBS) [15].

Computing the inverse of $\boldsymbol{Y}$ to obtain the dynamic stiffness in (2.36) (and vice versa) can however yield errors. Since the response of undamped systems near resonance will be extremely large and will be almost zero at anti-resonances, this leads to a poor conditioning of the flexibility matrix and hence cause errors in the computation of the stiffness matrix. Due to the fact that the LM FBS method only requires the inverse of a part of the flexibility matrix, it is in general a faster method that is easier to implement. Frequency based substructuring is not limited to experimental substructuring only. Since flexibility FRF's ($\boldsymbol{Y}$) can be synthesized from the computed eigenmodes and the stiffness FRF's ($\boldsymbol{Z}$) can be computed directly, frequency based substructuring can also be applied to numerical problems [17]. One should however be aware of modal truncation when using synthesized FRFs; residual information of higher frequency modes should be taken into account in order to obtain statically correct FRF's. More details on frequency based substructuring and LM FBS can be found in [14, 18, 19]. The mixed assembly presented in section 2.2.3 can also be applied in the frequency domain and one is then able to directly assemble flexibility FRF's ($\boldsymbol{Y}$) with stiffness FRF's ($\boldsymbol{Z}$).

2.4 Summary

The chapter starts with a short history and a brief overview of dynamic substructuring. In section 2.2 and 2.3 the assembly techniques in DS are discussed and it is shown that any substructure can be assembled with any other substructure within its domain. Here it is irrelevant whether the substructure is described in terms of stiffness or flexibility. The possible assembly techniques and cases have been summarized in table 2.1. A ✓ states that the assembly method is possible for this assembly case, a ✕ states that it

is not possible. Here n denotes the sum of the substructure DoF and n_b is the sum of a unique set of interface DoF; note that the number of DoF only apply to unreduced assemblies.

	Assembly methods		
Assembly case	Primal	Dual	Dirichlet-Neumann
$\boldsymbol{u}_b \leftrightarrow \boldsymbol{u}_b$	✓ NDoF: $n - n_b$ Symmetric	✓ NDoF: $n + n_b$ Symmetric	✓ NDoF: n Non-symmetric
$\boldsymbol{g}_b \leftrightarrow \boldsymbol{g}_b$	✓ NDoF: $n + n_b$ Symmetric	✓ NDoF: $n + 3n_b$ Symmetric	
$\boldsymbol{u}_b \leftrightarrow \boldsymbol{g}_b$	×	×	✓ NDoF: n Non-symmetric

Table 2.1: Overview possible assembly methods

CHAPTER 3

Component model reduction techniques

3.1 Introduction

In the field of structural dynamics we often use finite element models which were built to asses deformations and stress concentrations in structures. These models are often too refined and have several hundreds of thousands (or even millions) of degrees of freedom. Finding a static solution for these models does not pose any problems, since there are many efficient solvers to handle this. Solving dynamic problems, such as computing vibration modes, harmonic and/or transient responses, requires solving many static-like problems and results in much longer computation times. Often the dynamic behaviour of a structure can be well described using a coarser mesh. Sometimes though, further coarsening of the mesh will result in shape-function violations and errors due to the geometry of the structure. In addition, a coarser mesh will lead to higher eigenfrequencies, since by reducing the number of DoF for the structure to deform in, the structure is "artificially stiffened". An finally, remeshing the structure could be very costly and difficult, especially for complex structures.

A more elegant approach would be to reduce the number of DoF without modifying the mesh. Such methods exist and are known as *component model reduction methods*. The basic idea behind reduction methods in structural dynamics is in fact modal superposition , where the nodal displacements are written in terms of normal modes and modal amplitudes.

$$\boldsymbol{u} = \sum_{j=1}^{n} \boldsymbol{\phi}_j \eta_j$$

When applying this idea in dynamic substructuring, one will have to make sure the connecting forces from neighboring substructures are represented well by the reduction basis. One therefore (usually) includes some sort of static modes in the reduction basis, which represent the static deformation caused by neighboring substructures. These type of reduction methods are known under the name *Component Mode Synthesis* (CMS).

The undamped equations of motions of substructure s, denoted by the superscript $^{(s)}$, are given in (3.1). Only in this introduction the superscript will be explicitly shown, for ease of notation it will be discarded in the rest of the chapter.

$$\boldsymbol{M}^{(s)}\ddot{\boldsymbol{u}}^{(s)} + \boldsymbol{K}^{(s)}\boldsymbol{u}^{(s)} = \boldsymbol{f}^{(s)} + \boldsymbol{g}^{(s)} \tag{3.1}$$

The substructure's mass matrix is denoted by $\boldsymbol{M}^{(s)}$ and the stiffness matrix by $\boldsymbol{K}^{(s)}$. The force vector is split into $\boldsymbol{f}^{(s)}$, which denote the externally applied forces and $\boldsymbol{g}^{(s)}$, the forces that result from the neighboring substructures. Now the set of original DoF ($\boldsymbol{u}^{(s)}$) is transformed into a set of generalized DoF ($\boldsymbol{q}^{(s)}$):

$$\boldsymbol{u}^{(s)} = \boldsymbol{R}^{(s)}\boldsymbol{q}^{(s)} \tag{3.2}$$

Here $\boldsymbol{R}^{(s)}$ represents the reduction basis of dimension $n^{(s)} \times r^{(s)}$. For an efficient reduction it is required that the reduced set of DoF is very small in comparison to the original set of DoF ($r^{(s)} << n^{(s)}$). Using this new set of DoF, the equations of motion (3.1) now write:

$$\boldsymbol{M}^{(s)}\boldsymbol{R}^{(s)}\ddot{\boldsymbol{q}}^{(s)} + \boldsymbol{K}^{(s)}\boldsymbol{R}^{(s)}\boldsymbol{q}^{(s)} = \boldsymbol{f}^{(s)} + \boldsymbol{g}^{(s)} + \boldsymbol{r}^{(s)} \tag{3.3}$$

An error ($\boldsymbol{r}^{(s)}$) will be made, since the new set of DoF does not span the full solution space; it is an approximation of the exact solution. Here it is chosen to allow only an error in the space not spanned by the reduction basis. This is achieved by pre-multiplying the equations by $\boldsymbol{R}^{(s)^T}$, hence:

$$\boldsymbol{R}^{(s)^T}\boldsymbol{r}^{(s)} = \boldsymbol{0} \tag{3.4}$$

Using this property, the projecting of equations 3.3 onto the reduction basis gives:

$$\tilde{\boldsymbol{M}}^{(s)}\ddot{\boldsymbol{q}}^{(s)} + \tilde{\boldsymbol{K}}^{(s)}\boldsymbol{q}^{(s)} = \tilde{\boldsymbol{f}}^{(s)} + \tilde{\boldsymbol{g}}^{(s)} \tag{3.5}$$

Where:

$$\begin{cases} \tilde{\boldsymbol{M}}^{(s)} = \boldsymbol{R}^{(s)^T}\boldsymbol{M}^{(s)}\boldsymbol{R}^{(s)} \\ \tilde{\boldsymbol{K}}^{(s)} = \boldsymbol{R}^{(s)^T}\boldsymbol{K}^{(s)}\boldsymbol{R}^{(s)} \\ \tilde{\boldsymbol{f}}^{(s)} = \boldsymbol{R}^{(s)^T}\boldsymbol{f}^{(s)} \\ \tilde{\boldsymbol{g}}^{(s)} = \boldsymbol{R}^{(s)^T}\boldsymbol{g}^{(s)} \end{cases} \tag{3.6}$$

All kinds of "modes" (in fact Ritz vectors) can be used to create a reduction basis, such as exact eigenmodes, approximate modes, static modes, interface modes, etc. In section 3.2 (structural) modes commonly used in CMS methods are described. These modes are used in the subsequent sections to describe a number of CMS methods.

3.2 Commonly used modes in CMS

As mentioned in section 3.1, a reduction basis can be build from any sort of "modes". Generally a basis is build from a set of vibration modes, which contain information of the substructure's dynamic behavior, and a set of static modes, which represent the static behavior of the substructure to a unit interface force or displacement [20]. The most important types of vibrational and static modes will be presented in this section, since they form the ingredients for the CMS methods described later.

3.2.1 Free interface vibration modes

Free vibration modes are the vibration shapes of the substructure if the interface DoF are unconstrained. They are obtained by solving the free vibration eigenvalue problem :

$$\left(\boldsymbol{K} - \omega_r^2 \boldsymbol{M}\right) \boldsymbol{\phi}_{f,r} = \boldsymbol{0} \tag{3.7}$$

Here, $\boldsymbol{\phi}_{f,r}$ is the r^{th} free vibration mode with its associated eigenfrequency ω_r^2 . The free vibration modes thus contain information of the substructure dynamics and are the "true" eigenmodes of the system. Damping is neglected in the discussion, as already mentioned in the introduction. A single free interface vibration mode is denoted by $\boldsymbol{\phi}_f$ and a set of free interface vibration modes is denoted by $\boldsymbol{\Phi}_f$.

3.2.2 Rigid body modes

A special type of free vibration modes are rigid body modes; if the substructure is not fully constrained a number of rigid body modes exists. Rigid body modes are modes in which the substructure shows displacements without deformations; it displaces as a rigid body.

$$\boldsymbol{K}\boldsymbol{\Phi}_r = \boldsymbol{0} \tag{3.8}$$

In addition to the global rigid body displacements (displacements of the total structure), a structure could for instance also contain mechanisms (i.e. hinges etc.) which will allow for parts of the structure to displace without introducing any deformations, these displacements are in fact also rigid body modes and (3.8) also holds. For systems having no mechanisms, the geometry of the substructure can be used to obtain the rigid body modes without solving (3.8). A single rigid body mode is denoted by $\boldsymbol{\phi}_r$ and a set of rigid body modes is denoted by $\boldsymbol{\Phi}_r$. Rigid body modes can be computed from an eigenvalue analysis as in (3.7), but can also be computed in an other way which requires less computation effort.

Firstly, the translational rigid body modes are obtained by giving the nodes a unit displacement in the x, y and z directions, respectively.

$$\boldsymbol{\phi}_{r(x,j)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \boldsymbol{\phi}_{r(y,j)} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \boldsymbol{\phi}_{r(z,j)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad j = 1 \ldots n$$

Here n is the number of nodes in the FE model and the subscript (j) implies the j^{th} node.

For the rotational rigid body modes a node is chosen randomly (e.g. the black node in figure 3.1) and a position vector $\boldsymbol{d}$ from this "reference-node" to each of the other nodes is determined. The displacement of a node due to a rotation of the body is now computed by taking the cross product between the rotation vector $\boldsymbol{e}_\theta$ ($\theta = \theta_x, \theta_y, \theta_z$) and

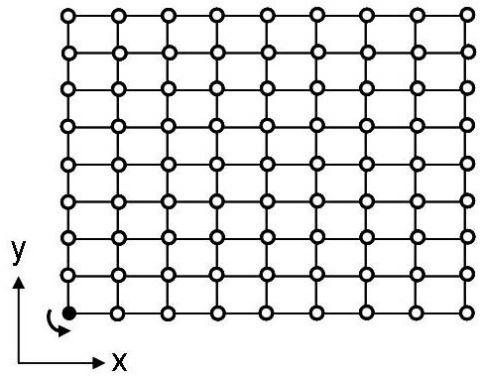


Figure 3.1: Geometry FE model

the vector $\boldsymbol{d}$, where θ is a rotation around the x, y or z axis. If the nodes have only three DoF per node (for example when using solid elements), the rigid body displacement of one node is according to (3.9). If the nodes have six DoF per node (for example when using shell elements), the rigid body displacement is according to (3.10).

$$\boldsymbol{\phi}^{geo}_{r(\theta),j} = \boldsymbol{e}_\theta \times \boldsymbol{d}_j, \quad j = 1 \dots n$$

$$\boldsymbol{\Phi}^{geo}_{r,j} = \begin{bmatrix} \boldsymbol{\phi}^{geo}_{r(x),j} & \boldsymbol{\phi}^{geo}_{r(y),j} & \boldsymbol{\phi}^{geo}_{r(z),j} & \boldsymbol{\phi}^{geo}_{r(\theta_x),j} & \boldsymbol{\phi}^{geo}_{r(\theta_y),j} & \boldsymbol{\phi}^{geo}_{r(\theta_z),j} \end{bmatrix}, \quad j = 1 \dots n \tag{3.9}$$

$$\boldsymbol{\Phi}^{geo}_{r(j)} = \begin{bmatrix} \boldsymbol{\phi}^{geo}_{r(x),j} & \boldsymbol{\phi}^{geo}_{r(y),j} & \boldsymbol{\phi}^{geo}_{r(z),j} & \boldsymbol{\phi}^{geo}_{r(\theta_x),j} & \boldsymbol{\phi}^{geo}_{r(\theta_y),j} & \boldsymbol{\phi}^{geo}_{r(\theta_z),j} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad j = 1 \dots n \tag{3.10}$$

The superscript *geo* denotes that the rigid body modes are found from the geometry of the structural model. By applying (3.9) or (3.10) to all nodes, the rigid body displacement vectors are obtained. By construction the translational rigid body displacement vectors are already mass orthogonal, i.e:

$$\boldsymbol{\phi}^{geo^T}_{r(k)} \boldsymbol{M} \boldsymbol{\phi}^{geo}_{r(l)} = 0$$
for $\mathrm{k} = \mathrm{x}, \mathrm{y}, \mathrm{z}$ and $\mathrm{l} = \mathrm{x}, \mathrm{y}, \mathrm{z}$
if $\mathrm{k} \neq \mathrm{l}$

However since the point of rotation is chosen randomly, it is very likely (since the chosen node will probably not be at the center of mass) that the rotational rigid body displacement vectors are usually a combination of a translational rigid body mode and a rotational rigid body mode. In that case the rigid body displacement vectors are not mass and stiffness orthogonal (note that the displacement vectors are not yet mass normalized).

$$\boldsymbol{\phi}^{geo^T}_{r(k} \boldsymbol{M} \boldsymbol{\phi}^{geo}_{r(l)} = 0$$
for $\mathrm{k} = \mathrm{x}, \mathrm{y}, \mathrm{z}, \theta_\mathrm{x}, \theta_\mathrm{y}, \theta_\mathrm{z}$ and $\mathrm{l} = \theta_\mathrm{x}, \theta_\mathrm{y}, \theta_\mathrm{z}$
if $\mathrm{k} \neq \mathrm{l}$

In order to orthogonalize the rigid body displacement vectors, the translational rigid body modes (which are already mass-orthogonal by construction) have to be projected out of the space spanned by the rotational rigid body displacement vectors. This is described in [21], the projection is gives:

$$\boldsymbol{\Phi}_{r(\theta)} = \boldsymbol{P}^{geo}_{r(t)}\boldsymbol{\Phi}^{geo}_{r(\theta)} = \left(\boldsymbol{I} - \boldsymbol{\Phi}^{geo}_{r(t)}\left(\boldsymbol{\Phi}^{geo^T}_{r(t)}\boldsymbol{M}\boldsymbol{\Phi}^{geo}_{r(t)}\right)^{-1}\boldsymbol{\Phi}^{geo^T}_{r(t)}\boldsymbol{M}\right)\boldsymbol{\Phi}^{geo}_{r(\theta)} \tag{3.11}$$

The subscript (t) denotes the translational rigid body modes and the subscript (θ) the rotational ones. After the projection the rigid body modes are mass and stiffness orthogonal. Alternatively, one could also choose to solve the eigenproblem on the subspace of the rigid body modes in order to orthogonalize the rigid body modes with respect to the mass matrix. A last step is to mass normalize the rigid body modes, after this last step the ortho-normalized rigid body modes are obtained.

$$\boldsymbol{\phi}_{r(k)} = \frac{\boldsymbol{\phi}_{r(k)}}{\boldsymbol{\phi}^T_{r(k)}\boldsymbol{M}\boldsymbol{\phi}_{r(k)}}, \quad k = 1 \dots n_r$$

Since in this approach the rbm are identified with respect to a chosen "reference-node", rigid body modes due to mechanisms within the model cannot be detected. One has to make sure the structure has no mechanisms which can move freely, otherwise the set of rigid body modes obtained will not be complete.

3.2.3 Fixed interface vibration modes

A second approach to include vibrational information in a reduction basis would be to use fixed interface vibration modes. These can be computed by constraining the boundary DoF. First, the system is partitioned into boundary DoF ($\boldsymbol{u}_b$) and internal DoF ($\boldsymbol{u}_i$) .

$$\begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bi} \\ \boldsymbol{M}_{ib} & \boldsymbol{M}_{ii} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{u}}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.12}$$

The next step is to constrain the boundary DoF ($\boldsymbol{u}_b = \boldsymbol{0}$), resulting in the following equation.

$$\boldsymbol{M}_{ii}\ddot{\boldsymbol{u}}_i + \boldsymbol{K}_{ii}\boldsymbol{u}_i = \boldsymbol{0} \tag{3.13}$$

This can be solved as an eigenvalue problem.

$$\left(\boldsymbol{K}_{ii} - \omega^2_{i,j}\boldsymbol{M}_{ii}\right)\boldsymbol{\phi}_{i,j} = \boldsymbol{0} \tag{3.14}$$

The result is the set of eigenmodes and eigenfrequencies of the substructure constrained at its boundary DoF. A single fixed interface vibration mode will be denoted by $\boldsymbol{\phi}_i$, a set of fixed interface vibration modes by $\boldsymbol{\Phi}_i$.

3.2.4 Constraint modes

A constraint mode is the static deformation shape due to a unit displacement applied to one of the boundary DoF, while the remaining boundary DoF are restrained and no forces are applied at the internal DoF. The set of constraint modes thus contains the substructure's static response to applied interface displacements. The computation of the constraint modes starts with splitting the DoF into boundary DoF ($\boldsymbol{u}_b$) and internal DoF ($\boldsymbol{u}_i$), giving the equation of motion (3.12). The second equation in (3.12) then writes:

$$\boldsymbol{M}_{ii}\ddot{\boldsymbol{u}}_i + \boldsymbol{M}_{ib}\ddot{\boldsymbol{u}}_b + \boldsymbol{K}_{ii}\boldsymbol{u}_i + \boldsymbol{K}_{ib}\boldsymbol{u}_b = \boldsymbol{0} \tag{3.15}$$

By neglecting the inertia forces in (3.15), the remaining "static" part can be condensed on the boundary DoF $\boldsymbol{u}_b$:

$$\boldsymbol{u}_{i,stat} = -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\boldsymbol{u}_b \tag{3.16}$$

$-\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}$ forms the static condensation matrix , whose columns contain so-called static modes. These static modes represent the static response of the internal DoF ($\boldsymbol{u}_i$) for unit displacements of the boundary DoF ($\boldsymbol{u}_b$). The original set of degrees of freedom $\boldsymbol{u}$ are reduced to a set of boundary DoF ($\boldsymbol{u}_b$), as:

$$\begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix} = \boldsymbol{\Psi}_C\boldsymbol{u}_b = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{bi} \end{bmatrix}\boldsymbol{u}_b \tag{3.17}$$

The constraint modes are denoted by $\boldsymbol{\Psi}_C$. Using these constraint modes, interface compatibility between substructures can easily be enforced. Since the original set of interface DoF ($\boldsymbol{u}_b$) are retained, primal assembly can be used to couple the substructure to other reduced substructures or full FE models, as described in section 2.2.1.

3.2.5 Attachment modes

Attachment modes are defined as the displacement vector due to a unit force applied at one of the interface DoF ($\boldsymbol{u}_b$). Attachment modes are therefor columns of the associated flexibility matrix ($\boldsymbol{G}$).

$$\begin{aligned} &\boldsymbol{K}^{+}\boldsymbol{f} = \boldsymbol{G}\boldsymbol{f} = \boldsymbol{u} \\ &\boldsymbol{f} = \begin{bmatrix} 0 & 0 & \dots & 1 \end{bmatrix}^T \end{aligned} \tag{3.18}$$

Here, $\boldsymbol{K}^{+}$ is the pseudo-inverse (or generalized inverse) of the stiffness matrix. Computing the attachment modes will be straightforward if the structure is fully constrained. One only has to solve the following set of equations:

$$\boldsymbol{\Psi}_a = \boldsymbol{K}^{-1}\boldsymbol{f} \tag{3.19}$$

Here $\boldsymbol{f}$ is a force vector containing unit forces at the boundary DoF and $\boldsymbol{\Psi}_a$ are the obtained *attachment modes*.

However, a difficulty encountered while computing the attachment modes of an unconstrained structure, is the presence of rigid body modes. For an unconstrained structure, a solution for (3.18) is not possible, unless the force vector $\boldsymbol{f}$ is self-equilibrating. One could compare the (static) system in (3.18) as a structure without any mass free floating in space. By applying a non-equilibrating force, the structure undergoes an infinite acceleration and undetermined deformations, due to the absence of equilibrating forces. In this case, three steps have to be taken in order to find the attachment modes, which are now referred to as *flexibility attachment modes* [20]:

1. Compute a generalized inverse of $\boldsymbol{K}$.
2. Determine a self-equilibrating input force vector $\boldsymbol{f}_{eq}$.
3. Mass-orthogonalize the set of obtained attachment modes with the set of rigid body modes.

These steps will be described in this section.

The first step is to compute a generalized inverse of $\boldsymbol{K}$. In the case that structure is "free floating" (in other words, the structure has rigid body modes), the generalized inverse of the stiffness matrix ($\boldsymbol{K}^+$) is not unique. One option for computing ($\boldsymbol{K}^+$) is to constrain the structure at a set of some DoF ($\boldsymbol{u}_0$), also referred to as *isostatic constraints* or *temporary links* (these will be discussed later on in this section), and compute the inverse of the constrained stiffness matrix.

$$\begin{bmatrix} \boldsymbol{K}_{00} & \boldsymbol{K}_{0i} & \boldsymbol{K}_{0b} \\ \boldsymbol{K}_{i0} & \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{b0} & \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_C \\ \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} \tag{3.20}$$

Here the constrained DoF are set to zero and $\boldsymbol{f}_C$ are the associated constraint forces. By constraining the system, the lower part is no longer singular. This allows for computing the so called *constrained flexibility matrix* $\boldsymbol{G}_c$, which is a generalized inverse of $\boldsymbol{K}$:

$$\boldsymbol{K}^+ = \boldsymbol{G}_c = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_{ii} & \boldsymbol{G}_{ib} \\ \boldsymbol{0} & \boldsymbol{G}_{bi} & \boldsymbol{G}_{bb} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} \end{bmatrix}^{-1} \\ \boldsymbol{0} & \end{bmatrix} \tag{3.21}$$

The second step in obtaining the attachment modes is to determine an equilibrated force vector $\boldsymbol{f}_{eq}$, where the original force vector $\boldsymbol{f}$ is added to an equilibrating inertia force $\boldsymbol{M}\ddot{\boldsymbol{\Phi}}_r$, resulting from rigid body accelerations.

$$\boldsymbol{f}_{eq} = \boldsymbol{f} - \boldsymbol{M}\ddot{\boldsymbol{\Phi}}_r \tag{3.22}$$

The difficulty here is to obtain these inertia forces. Starting by separating the displacement vector ($\boldsymbol{u}$) into a rigid body displacement vector ($\boldsymbol{u}_{rb}$) and a flexible body displacement vector ($\boldsymbol{u}_f$) and recalling the principle of modal superposition [22], one can write:

$$\boldsymbol{u} = \boldsymbol{u}_r + \boldsymbol{u}_f = \boldsymbol{\Phi}_r\boldsymbol{\eta}_r + \boldsymbol{\Phi}_f\boldsymbol{\eta}_f \tag{3.23}$$

The rigid body modes are included in $\boldsymbol{\Phi}_r$, with $\boldsymbol{\eta}_r$ as their modal amplitudes and *all* the flexible body modes are included in $\boldsymbol{\Phi}_f$ (i.e. no modal truncation is applied) with $\boldsymbol{\eta}_f$ as their modal amplitudes. Substitution of (3.23) into (3.1) the modal formulation of the equations of motion is obtained:

$$\boldsymbol{M\Phi}_r\ddot{\boldsymbol{\eta}}_r + \boldsymbol{M\Phi}_f\ddot{\boldsymbol{\eta}}_f + \boldsymbol{K\Phi}_r\boldsymbol{\eta}_r + \boldsymbol{K\Phi}_f\boldsymbol{\eta}_f = \boldsymbol{f} \tag{3.24}$$

The rigid body modes do not introduce any deformations to the system ($\boldsymbol{K\Phi}_r = \boldsymbol{0}$, see section 3.2.2) and the rigid body inertia forces are interpreted as a set of external forces. Pre-multiplication by $\boldsymbol{\Phi}_r^T$ gives:

$$\boldsymbol{\Phi}_r^T\left(\boldsymbol{M\Phi}_f\ddot{\boldsymbol{\eta}}_f + \boldsymbol{K\Phi}_f\boldsymbol{\eta}_f = \boldsymbol{f} - \boldsymbol{M\Phi}_r\ddot{\boldsymbol{\eta}}_r\right)$$

By applying the mode orthogonality relationship to the remaining equation, one finds:

$$\boldsymbol{\Phi}_r^T\boldsymbol{f} = \boldsymbol{\Phi}_r^T\boldsymbol{M\Phi}_r\ddot{\boldsymbol{\eta}}_r$$

In the subsequent steps, the expression for the equilibrated input force $\boldsymbol{f}_{eq}$ is obtained.

$$\left(\boldsymbol{\Phi}_r^T\boldsymbol{M\Phi}_r\right)^{-1}\boldsymbol{\Phi}_r^T\boldsymbol{f} = \ddot{\boldsymbol{\eta}}_r$$

Substituting the obtained expression for $\ddot{\boldsymbol{\eta}}_r$ into (3.22) gives :

$$\boldsymbol{f}_{eq} = \left(\boldsymbol{I} - \boldsymbol{M\Phi}_r\left(\boldsymbol{\Phi}_r^T\boldsymbol{M\Phi}_r\right)^{-1}\boldsymbol{\Phi}_r^T\right)\boldsymbol{f} = \boldsymbol{P}\boldsymbol{f} \tag{3.25}$$

The obtained matrix $\boldsymbol{P}$ is also referred to as the *inertia-relief projection matrix* [20]. Any force vector that is pre-multiplied by $\boldsymbol{P}$ will become self-equilibriated. $\boldsymbol{P}$ projects the input vector $\boldsymbol{f}$ on a space outside of the rigid body modes (which is similar to (3.11)), such that $\boldsymbol{f}_{eq}$ will not excite any rigid body modes. By substituting $\boldsymbol{f}_{eq}$ for $\boldsymbol{f}$ into (3.18), a solution to the problem can now be found.

$$\boldsymbol{K}^+\boldsymbol{f}_{eq} = \boldsymbol{G}_c\boldsymbol{f}_{eq} = \tilde{\boldsymbol{\Psi}}_a \tag{3.26}$$

The final step is to orthogonalize $\tilde{\boldsymbol{\Psi}}_a$ with respect to the rigid body modes to find the attachment modes $\boldsymbol{\Psi}_a$. This is done by pre-multiplying $\tilde{\boldsymbol{\Psi}}_a$ by $\boldsymbol{P}^T$.

$$\boldsymbol{\Psi}_a = \boldsymbol{P}^T\tilde{\boldsymbol{\Psi}}_a = \boldsymbol{P}^T\boldsymbol{K}^+\boldsymbol{P}\boldsymbol{f} \tag{3.27}$$

By replacing the generalized inverse $\boldsymbol{K}^+$, by the constrained flexibility matrix $\boldsymbol{G}_c$ (3.21), the *elastic flexibility matrix* is found.

$$\boldsymbol{G}_f = \boldsymbol{P}^T\boldsymbol{G}_c\boldsymbol{P} \tag{3.28}$$

The elastic flexibility matrix has the following properties.

$$\begin{aligned}&\boldsymbol{G}_f^T = \boldsymbol{G}_f\\ &\boldsymbol{G}_f^T\boldsymbol{K}\boldsymbol{G}_f = \boldsymbol{G}_f\\ &\boldsymbol{U}_{rb}^T\boldsymbol{M}\boldsymbol{G}_f = \boldsymbol{0}\end{aligned} \tag{3.29}$$

The flexibility attachment modes are therefore the columns of the elastic flexibility matrix associated with the boundary DoF.

$$\boldsymbol{\Psi}_a = \boldsymbol{G}_f \boldsymbol{f} \tag{3.30}$$

One can extend this approach to computing quasi-static attachment modes around a certain frequency ω_p, as is presented in [23]. Whereas the attachment modes exactly represent the structures deformation resulting from static forces, the quasi static attachment modes exactly represent the structures deformation resulting from forces at a frequency ω_p.

Obtaining the isostatic constraints

In order to obtain the constrained flexibility matrix ($\boldsymbol{G}_c$) described above, a set of isostatic constraints has to be chosen (3.21). This set of constraints is used to (mathematically) constrain the substructure, such that:

$$\boldsymbol{K}_{iso} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_C \end{bmatrix}$$

Here $\boldsymbol{K}_C$ is the constrained part of the stiffness matrix $\boldsymbol{K}$. Since $\boldsymbol{K}_C$ is non-singular, it is also invertible. Applying loads to the obtained isostatic constrained stiffness matrix will result in constraint forces ($\boldsymbol{f}_C$) which will ensure the structure is in equilibrium. One can imagine that the constraint forces should be as small as possible in order to get the best possible estimate of the static deformation of the structure due to a unit interface force.

$$\begin{bmatrix} \boldsymbol{f}_C \\ \boldsymbol{f}_i \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{static} \end{bmatrix} \tag{3.31}$$

Not every set of constraints will lead to the best possible solution for $\boldsymbol{u}_{static}$. In order compute an "optimal" constrained flexibility matrix, the DoF that have the highest rigid body displacements need to be constrained. This can be seen as an "optimal" set of constraints, in the sense that they result in minimal constraint forces.

In figure 3.2 the algorithm for computing these isostatic constraints is given and will be discussed stepwise here.

1. We start with the rigid body modes determined from the geometry $\boldsymbol{\Phi}_r^{geo}$, as discussed earlier in this section.
2. The second step is to check whether the structure is constrained at certain DoF. If this is the case the constraints matrix $\boldsymbol{S}_{phy}$ is build up and is an all-zero matrix with the same number of columns as substructure DoF and the number of rows is equal to the number of constraints applied ($n_{phy} \times n$). At the indices which are associated with the constrained DoF, a 1 is placed, resulting in for example:

$$\boldsymbol{S}_{phy} = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix}$$

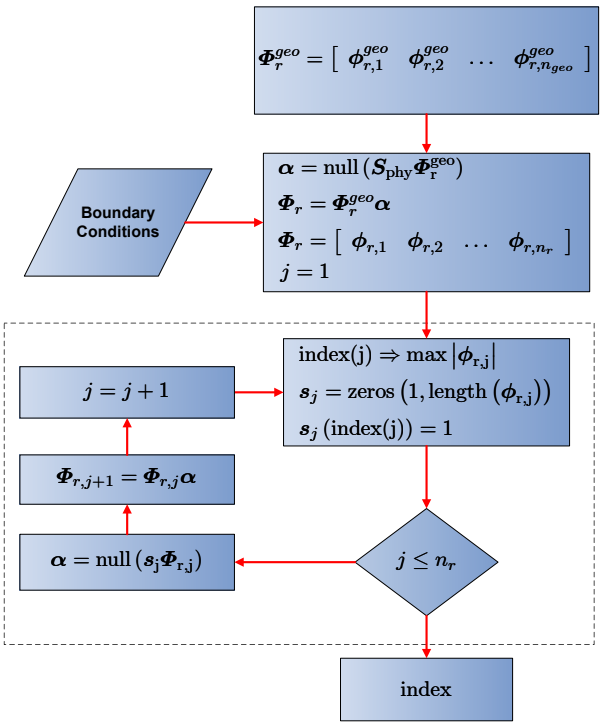


Figure 3.2: Algorithm for obtaining isostatic constraints

Now, $\boldsymbol{\alpha}$ can be computed by taking the nullspace of the constrained lines of the rigid body modes, and this $\boldsymbol{\alpha}$ gives an answer to the question of what combination of "old" rigid body modes will generate the"new" set of rigid body modes. One could interpret this as computing what (combinations of) rigid motions are still possible after applying the physical constraints. The actual set of rigid body modes ($\boldsymbol{\Phi}_r$) is obtained by multiplying the geometric rigid body modes ($\boldsymbol{\Phi}_r^{geo}$) with $\boldsymbol{\alpha}$.

$$\boldsymbol{\Phi}_r = \boldsymbol{\Phi}_r^{geo}\boldsymbol{\alpha}$$

This new set of rigid body modes, are the actual (physical) rigid body modes of the structure at hand. If the structure is not fully constrained and some rigid body modes remain, we will need to determine the set of isostatic constraints.

3. The set of rigid body modes computed in the first two steps is now used in the rest of the algorithm. At the start of the algorithm j is equal to one. In the first block within the computation loop, the DoF that has the largest relative (translational) displacement due to the j^{th} rigid body mode is found and subsequently constrained. Rotations will not be considered, since this is a different physical quantity, one cannot directly compare a rotation to a translation. The first step in the loop is to find the j^{th} isostatic constraint, this could for instance give:

$$\boldsymbol{s}_1 = [\ 1 \quad 0 \quad \dots \quad 0\ ]$$

4. As long as $j < n_r$, the constraint vector $\boldsymbol{s}_j$ is used to determine a new $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \text{null}\,(\boldsymbol{s}_{\text{j}}\boldsymbol{\Phi}_{\text{r,j}})$$

By adding a constraint to the structure, the set of rigid body modes (rbm) will alter. Firstly, one rigid body mode is now constrained, secondly the point of rotation for the rotational rigid body modes is shifted as is already explained in step 1.

5. Since the new set of rbm will be within the space of the original set of rbm, we can write the new set of rbm as a linear combination of the old set of rbm. Note that $\boldsymbol{\alpha}$ can be a vector or a matrix, depending on the number of constraints applied to the structure.

$$\boldsymbol{\Phi}_{r,j+1} = \boldsymbol{\Phi}_{r,j}\boldsymbol{\alpha}$$

6. As described, the new set of rigid body modes is computed. Using this new set of rigid body modes as input for the first step in the loop, creates the next isostatic constraint. This is repeated until the structure is statically determined. Using the set of isostatic constraints, the stiffness matrix can be split according to (3.20).

3.2.6 Residual Flexibility Attachment modes

A special type of attachment modes are the residual flexibility attachment modes. Due to their construction (see section 3.2.5), attachment modes are mass-orthogonal to the rigid body modes.

$$\boldsymbol{\Psi}_a^T \boldsymbol{M} \boldsymbol{U}_{rb} = \boldsymbol{0}$$

However the attachment modes will not be mass and stiffness orthogonal to the free interface vibration modes. The attachment modes are obtained by taking the columns of the elastic flexibility matrix ($\boldsymbol{G}_f$) associated to the boundary DoF ($\boldsymbol{u}_b$), but the same flexibility matrix can also be build from a spectral expansion of the normal eigenmodes [21]:

$$\boldsymbol{G} = \sum_{r=1}^{n-m} \frac{\boldsymbol{\theta}_r \boldsymbol{\theta}_r^T}{\omega_r^2} \tag{3.32}$$

Or in other words, although they are constructed differently, $\boldsymbol{G}$ and $\boldsymbol{G}_f$ are physically the same and span the same subspace. The idea behind reduction techniques is to reduce the number of DoF, so usually only $k << n$ (normal) vibration modes are included as a basis for the reduction. The reduction basis could be augmented using attachment modes, but since these attachment modes contain some of the same flexibility information in the vibration modes, it is beneficial to correct the attachment modes in order to create a $\boldsymbol{M}$ and $\boldsymbol{K}$ orthogonal basis.

$$\boldsymbol{G}_{res} = \boldsymbol{G} - \sum_{r=1}^{k-m} \frac{\boldsymbol{\theta}_r \boldsymbol{\theta}_r^T}{\omega_r^2} = \sum_{s=k+1}^{n} \frac{\boldsymbol{\theta}_s \boldsymbol{\theta}_s^T}{\omega_s^2} \tag{3.33}$$

The now obtained matrix $\boldsymbol{G}_{res}$ is the *residual flexibility matrix* and is equal to a spectral expansion of the $d = n - k$ discarded normal modes. In addition to the properties of the flexibility matrix (3.29), the residual flexibility matrix has the following properties:

$$\begin{aligned} \boldsymbol{\Phi}_f^T \boldsymbol{M} \boldsymbol{G}_{res} &= \boldsymbol{0} \\ \boldsymbol{\Phi}_f^T \boldsymbol{K} \boldsymbol{G}_{res} &= \boldsymbol{0} \end{aligned} \tag{3.34}$$

From the residual flexibility matrix the *residual flexibility attachment modes* can be found by simply picking the columns associated to the boundary DoF $\boldsymbol{u}_b$.

$$\boldsymbol{G}_{res} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix} = \boldsymbol{\Psi}_{ar} \tag{3.35}$$

Since the residual flexibility attachment modes are columns of the residual flexibility matrix, they are also mass and stiffness orthogonal to the normal modes.

The ingredients described in this section will be used throughout this chapter to describe a number of CMS methods.

3.3 Guyan reduction

One of the oldest reduction methods is Guyan reduction [20, 21, 24]. The dynamic equations that govern the system are:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \tag{3.36}$$

In order to reduce the system, the degrees of freedom $\boldsymbol{u}$ are split into internal DoF ($\boldsymbol{u}_i$) and interface (or boundary) DoF ($\boldsymbol{u}_b$). By assuming no external forces are applied to the internal DoF $\boldsymbol{u}_i$ and neglecting the substructures internal inertia forces, the internal DoF can be condensed on the boundary DoF using the constraint modes described in section 3.2.4.

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{\Psi}_C \boldsymbol{u}_b \tag{3.37}$$

The reduction basis is thus formed by the set of constraint modes. Substitution of (3.37) into (3.36), will lead to the reduced dynamic equations, which are now only a function of the boundary DoF.

$$\tilde{\boldsymbol{M}}\ddot{\boldsymbol{u}}_b + \tilde{\boldsymbol{K}}\boldsymbol{u}_b = \tilde{\boldsymbol{f}}_b \tag{3.38}$$

Where:

$$\begin{cases} \tilde{\boldsymbol{K}} = \boldsymbol{\Psi}_C^T \boldsymbol{K} \boldsymbol{\Psi}_C = \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \\ \tilde{\boldsymbol{M}} = \boldsymbol{\Psi}_C^T \boldsymbol{M} \boldsymbol{\Psi}_C = \boldsymbol{M}_{bb} - \boldsymbol{M}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ib} + \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \\ \tilde{\boldsymbol{f}}_b = \boldsymbol{\Psi}_C^T \boldsymbol{f}_b \end{cases}$$

(3.39)

Since in the derivation of the condensed stiffness the inertia forces are neglected, the exact solution is found if this technique is applied to static problems. If it is applied to dynamic problems, an approximate solution is found. This is due to the fact that the internal inertia forces of the substructure are statically condensed on the interface and only contribute in a quasi-static manner. This approximation is valid as long as the highest eigenfrequency (ω) one wants to compute for the entire structure is much lower than the lowest eigenfrequency ($\omega^{(s)}$) of the (reduced) substructure when $\boldsymbol{q}_b$ is clamped. Due to the easy assembly of the reduced matrices in the finite element method, the Guyan reduced components are classified in the so called *superelements*.

Suppose we want to evaluate the total (reduced) structure around a certain frequency (ω_p). Instead of neglecting the inertia forces in the reduction step for substructure s, we could incorporate the inertia forces around the frequency ω_p. Eq. (3.36) then becomes:

$$-\omega^2 \boldsymbol{M}\boldsymbol{u} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{Z}(\omega)\boldsymbol{u} = \boldsymbol{f}(\omega) \tag{3.40}$$

Here $\boldsymbol{Z}(\omega)$ is the dynamic stiffness . This dynamic stiffness matrix is now used to condensate the internal degrees of freedom onto the boundary DoF.

$$\boldsymbol{u}_i = -\boldsymbol{Z}_{ii}^{-1}(\omega_p)\boldsymbol{Z}_{ib}(\omega_p)\boldsymbol{u}_b = \hat{\boldsymbol{S}}(\omega_p)\boldsymbol{u}_b \tag{3.41}$$

Using the static condensation of (3.41) in (3.37) and (3.39), will lead to the reduced set of equations. By taking into account the dynamic stiffness at ω_p in the condensation process, the exact behavior of the substructure will no longer be found for a static problem ($\omega = 0$), but for a harmonic problem at $\omega = \omega_p$ [25].

3.4 The Craig-Bampton method

The Craig-Bampton method [10, 21] can be seen as an expansion on Guyan's reduction. Instead of reducing with only constraint modes, the Craig-Bampton method also includes internal vibrational information and thereby creates a more complete basis for the reduction. This vibrational information is the set of fixed-vibration modes; the substructure is fixed at its boundary DoF and an analysis is done in order to obtain the eigenmodes (see section 3.2.3). Again, splitting $\boldsymbol{u}$ into the boundary DoF $\boldsymbol{u}_b$ and internal DoF $\boldsymbol{u}_i$, the system of equations writes (3.42).

$$\begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bi} \\ \boldsymbol{M}_{ib} & \boldsymbol{M}_{ii} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{u}}_i \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{g}_b \\ \boldsymbol{0} \end{bmatrix} \tag{3.42}$$

Here $\boldsymbol{g}_b$ are the reaction forces with the neighboring substructures. The constraint modes are computed as described in section 3.2.4. The internal DoF ($\boldsymbol{u}_i$) are now described in terms of constraint modes ($\boldsymbol{\Psi}_C$) and fixed-interface vibration modes ($\boldsymbol{\Phi}_i$).

$$\boldsymbol{u}_i = \boldsymbol{\Psi}_{C,i}\boldsymbol{u}_b + \boldsymbol{\Phi}_i\boldsymbol{\eta}_i$$

The denotation i denotes the part of the vector or matrix associated to the internal DoF, whereas b denotes the part of the vector or matrix associated to the boundary DoF. The reduction basis in matrix form will then be:

$$\begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{\Psi}_{C,i}\boldsymbol{u}_b + \boldsymbol{\Phi}_i\boldsymbol{\eta}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{\Psi}_{C,i} & \boldsymbol{\Phi}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{\eta}_i \end{bmatrix} = \boldsymbol{R}_{CB} \begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{\eta}_i \end{bmatrix} \tag{3.43}$$

Using the reduction matrix $\boldsymbol{R}_{CB}$ to reduce the original set of equations (3.42), we obtain:

$$\begin{aligned} \tilde{\boldsymbol{K}} &= \boldsymbol{R}_{CB}^T \boldsymbol{K} \boldsymbol{R}_{CB} \\ \tilde{\boldsymbol{M}} &= \boldsymbol{R}_{CB}^T \boldsymbol{M} \boldsymbol{R}_{CB} \end{aligned} \tag{3.44}$$

Here $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{K}}$ are the reduced mass and stiffness matrices, which are given in more detail below.

$$\begin{aligned} \tilde{\boldsymbol{K}} &= \begin{bmatrix} \tilde{\boldsymbol{K}}_{bb} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Omega}_i^2 \end{bmatrix} \\ \tilde{\boldsymbol{K}}_{bb} &= \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \\ \tilde{\boldsymbol{M}} &= \begin{bmatrix} \tilde{\boldsymbol{M}}_{bb} & \tilde{\boldsymbol{M}}_{b\zeta} \\ \tilde{\boldsymbol{M}}_{\zeta b} & \boldsymbol{I} \end{bmatrix} \\ \tilde{\boldsymbol{M}}_{bb} &= \boldsymbol{M}_{bb} - \boldsymbol{M}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ib} + \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib} \\ \tilde{\boldsymbol{M}}_{\zeta b} &= \boldsymbol{\Phi}_m^T \left(\boldsymbol{M}_{ib} - \boldsymbol{M}_{ii}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\right) = \tilde{\boldsymbol{M}}_{b\zeta}^T \end{aligned} \tag{3.45}$$

One of the advantages of the Craig-Bampton method is the straightforward reduction; both the constraint modes and the fixed interface modes can easily be computed. Secondly, in the reduced system the original interface DoF ($\boldsymbol{u}_b$) are retained, thereby allowing for easy assembly of the reduced substructures as superelements in regular FE packages. One can easily add, remove or replace substructures, without having to redo an analysis on the full model. A disadvantage of Craig-Bampton reduction is that if the interface of the substructure is changed (for example due to connecting more components), the entire reduction basis changes and has to be recomputed.

Assembly of reduced substructures is discussed in section 2.2.

3.5 The Rubin and Mac Neal methods

In the method developed by Craig and Bampton, fixed interface modes are employed. Another possibility would be to use free interface vibration modes and approximate the nodal displacements as:

$$\boldsymbol{u} = \boldsymbol{u}_{stat} + \sum_{j=1}^{n-m} \boldsymbol{\phi}_{f,j}\eta_{f,j}$$

A direct result of this choice is that rigid body modes have to be included in the static part of $\boldsymbol{u}$ if the substructure is not statically constrained. In order to include the flexible static information, the basis is augmented by residual flexibility attachment modes (see section 3.2.6), that represent the static response due to a unit force at one of the boundary DoF. These ingredients for a reduction basis are proposed by both Mac Neal [11] and Rubin [12] (and also Rixen later [26]). The reduction basis then writes:

$$\boldsymbol{u} = \boldsymbol{\Psi}_{ar}\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r + \boldsymbol{\Phi}_f\boldsymbol{\eta}_f \tag{3.46}$$

This reduction basis leads to a "dual" system, with $\boldsymbol{\eta}$ to describe the modal amplitudes and $\boldsymbol{g}_b$ to describe the interface forces. In order to facilitate easy assembly in finite element software, the reduction basis is transformed in order to obtain interface *displacements* instead of interface *forces*. By pre-multiplying (3.46) by $\boldsymbol{b}^T$, which is a Boolean matrix acting on the interface DoF ($\boldsymbol{u}_b$), we obtain the following expression for $\boldsymbol{u}_b$.

$$\boldsymbol{u}_b = \boldsymbol{b}\boldsymbol{u} = \boldsymbol{b}\left(\boldsymbol{\Psi}_{ar}\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r + \boldsymbol{\Phi}_f\boldsymbol{\eta}_f\right) = \boldsymbol{\Psi}_{ar,b}\boldsymbol{g}_b + \boldsymbol{\Phi}_{r,b}\boldsymbol{\eta}_r + \boldsymbol{\Phi}_{f,b}\boldsymbol{\eta}_f$$

From this equation the interface forces ($\boldsymbol{g}_b$) can be written as:

$$\boldsymbol{g}_b = \boldsymbol{\Psi}_{ar,b}^{-1}\left(\boldsymbol{u}_b - \boldsymbol{\Phi}_{r,b}\boldsymbol{\eta}_r - \boldsymbol{\Phi}_{f,b}\boldsymbol{\eta}_f\right) \tag{3.47}$$

Both Rubin's method [12,27] and the method proposed by Mac Neal [11,27] employ this idea for constructing their reduction bases. The difference between the two methods is in the construction of the reduced mass matrix. Whereas Mac Neal reduces the mass and stiffness matrix using only free vibration modes and substitutes the connection forces ($\boldsymbol{g}_b$) for (3.47) to reduce the substructure, Rubin extends on (3.47) and builds a reduction basis to reduce the mass and stiffness matrix consistently. Both methods will be discussed in detail in the following subsections.

3.5.1 Mac Neal's method

As a reminder, the set of equations of motion is:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} + \boldsymbol{g}$$

In the reduction technique proposed by Mac Neal the displacement field is expressed as a linear combination of the free interface modes.

$$\boldsymbol{u} = \boldsymbol{\Phi}\boldsymbol{\eta} \tag{3.48}$$

where:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_f \end{bmatrix}$$

Now (3.48) is substituted in the set of equations of motion and the obtained set of equations is pre-multiplied by (3.48) transposed. From (3.47) we know the interface forces can also be written in terms of the generalized DoF:

$$\boldsymbol{\Phi}^T\boldsymbol{M}\boldsymbol{\Phi}\ddot{\boldsymbol{\eta}} + \boldsymbol{\Phi}^T\boldsymbol{K}\boldsymbol{\Phi}\boldsymbol{\eta} = \boldsymbol{\Phi}^T\boldsymbol{f} + \boldsymbol{\Phi}^T\boldsymbol{b}^T\boldsymbol{\Psi}_{ar,b}^{-1}\left(\boldsymbol{u}_b - \boldsymbol{\Phi}_{r,b}\boldsymbol{\eta}_r - \boldsymbol{\Phi}_{f,b}\boldsymbol{\eta}_f\right) \tag{3.49}$$

Due to the orthogonality relationships and the fact that the normal modes are mass-normalized, the reduced mass matrix will be (partially) identity. Rewriting (3.49) into a matrix vector form leads to the following system of equations.

$$\begin{aligned}&\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_{rb} \\ \ddot{\boldsymbol{\eta}}_f \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \cdots \\ &\cdots \begin{bmatrix} \boldsymbol{\Phi}_{r,b}^T\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{r,b} & \boldsymbol{0} & -\boldsymbol{\Phi}_{r,b}^T\boldsymbol{\Psi}_{ar,b}^{-1} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{f,b}^T\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{f,b} + \boldsymbol{\Omega}_f^2 & -\boldsymbol{\Phi}_{f,b}^T\boldsymbol{\Psi}_{ar,b}^{-1} \\ -\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{r,b} & -\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{f,b} & \boldsymbol{\Psi}_{ar,b}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r^T\boldsymbol{f} \\ \boldsymbol{\Phi}_f^T\boldsymbol{f} \\ \boldsymbol{g}_b \end{bmatrix}\end{aligned} \tag{3.50}$$

where:

$$\boldsymbol{\Omega}_f^2 = \boldsymbol{\Phi}^T\boldsymbol{K}\boldsymbol{\Phi} = \operatorname{diag}\left(\omega_1^2, \omega_2^2, \ldots, \omega_{\mathrm{k}}^2\right)$$

Although the substructure is reduced using free vibration modes and residual flexibility modes, the reduction process still leads to a so called *superelement* since the physical boundary DoF ($\boldsymbol{u}_b$) are kept within the set of generalized DoF. These boundary DoF can be easily assembled with other superelements or full finite element models.

The inconsistency of Mac Neal's method is in the fact that the mass and stiffness matrices are not reduced with the same basis. This inconsistency will result in an uncoupled reduced mass matrix (sparse) and a coupled reduced stiffness matrix (full). When assembling the reduced substructures, the third equation of (3.50), will enforce an exact compatibility between the substructures. This will lead to a stiffening effect on the interface which could lead to interface locking and as a result significant errors if the substructures have displacement fields that can hardly define a compatible interface.

3.5.2 Rubin's method

Rubin [12] starts by transforming the reduction basis proposed in (3.46) from interface forces ($\boldsymbol{g}_b$) to interface displacements ($\boldsymbol{u}_b$). By substituting (3.47) into (3.46), the final reduction basis is obtained.

$$\boldsymbol{u}_i = \boldsymbol{\Psi}_{ar,i}\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{u}_b + \left(\boldsymbol{\Phi}_{r,i} - \boldsymbol{\Psi}_{ar,i}\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{r,b}\right)\boldsymbol{\eta}_r + \left(\boldsymbol{\Phi}_{f,i} - \boldsymbol{\Psi}_{ar,i}\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{f,b}\right)\boldsymbol{\eta}_f \tag{3.51}$$

or, in matrix formulation.

$$\begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix} = \boldsymbol{R}_R \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{u}_b \end{bmatrix} = \cdots$$
$$\cdots \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{\Phi}_{r,i} - \boldsymbol{\Psi}_{ar,i}\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{r,b} & \boldsymbol{\Phi}_{f,i} - \boldsymbol{\Psi}_{ar,i}\boldsymbol{\Psi}_{ar,b}^{-1}\boldsymbol{\Phi}_{f,b} & \boldsymbol{\Psi}_{ar,i}\boldsymbol{\Psi}_{ar,b}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{u}_b \end{bmatrix} \tag{3.52}$$

The reduction matrix $\boldsymbol{R}_R$ is used to reduce the full set of equations of motion of a substructure (3.1). In this method both matrices are reduced consistently in terms of displacements.

$$\begin{aligned} \tilde{\boldsymbol{K}} &= \boldsymbol{R}_R^T \boldsymbol{K} \boldsymbol{R}_R \\ \boldsymbol{M}_R &= \boldsymbol{R}_R^T \boldsymbol{M} \boldsymbol{R}_R \end{aligned} \tag{3.53}$$

It can be proven that the reduced stiffness matrix, obtained through Rubin and Mac Neal are equal. The difference is in the reduced mass matrix; the Mac Neal reduced system results in a complete inertia uncoupling between the boundary DoF ($\boldsymbol{u}$) and the modal amplitudes ($\boldsymbol{\eta}_r$ and $\boldsymbol{\eta}_f$) and thus the internal DoF, whereas Rubin's method leads to a full inertia coupling. Similar to Mac Neal's method, the reduction gives a *superelement* which can easily be assembled in finite element methods. An advantage that the Rubin and Mac Neal methods have over the Craig-Bampton method is that the reduction basis only partially changes if one alters the set of interface DoF. The free interface modes and the associated parts of the reduced matrices will not change, and one only has to recompute the parts of the reduction basis and matrices that are associated to the interface DoF.

Reduction methods using free-vibration modes are usually more accurate in the lower frequency domain, in comparison to methods using fixed-interface vibration modes. Rigid body modes have a significant influence on the lower frequency responses, by including these explicitly in the reduction basis, one can imagine this will improve the accuracy in the lower frequency domain. It is clear from both methods, the sparseness of the stiffness matrices is lost, the reduced mass matrix resulting from Rubin's method will also be full. Although the method proposed by Mac Neal gives a (very) sparse mass matrix, this is due to the fact that the reduction method is not fully consistent, and will in general lead to less accurate results than the method of Rubin.

3.6 The Dual Craig-Bampton method

The Dual Craig Bampton method uses the free interface vibration modes (section 3.2.1) and residual flexibility modes (section 3.2.6) of the system to form a reduction basis [28]. These modes are also proposed by Rubin and Mac Neal as a basis for reduction (section 3.5), but there is a difference between the methods. Where Rubin and Mac Neal transform the interface forces back to interface displacements to enable primal assembly

of reduced structures, the Dual Craig Bampton method maintains the interface forces as part of the new set of generalized DoF. In this section the method will be presented and we will show that assembly of the reduced substructures is quite similar to "normal" superelement assembly and is described in section 2.2.2 and 2.2.3.

The original set of DoF can be written as (3.54). Here the DoF ($\boldsymbol{u}$) can be represented in terms of the free vibration modes of the substructure and a static solution.

$$\boldsymbol{u} = \boldsymbol{\Psi}_{ar}\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r + \boldsymbol{\Phi}_f\boldsymbol{\eta}_f \tag{3.54}$$

The full set of DoF is thus written in terms of residual attachment modes ($\boldsymbol{\Psi}_{ar}$), rigid body modes ($\boldsymbol{\Phi}_r$) and free interface vibration modes ($\boldsymbol{\Phi}_f$). This reduction basis is the same as the reduction basis introduced in (3.46). Rewriting the reduction basis into a matrix-vector form gives the reduction basis $\boldsymbol{R}_{DCB}$:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} = \boldsymbol{R}_{DCB} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_f & \boldsymbol{\Psi}_{ar} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{\eta}_f \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Psi}_{ar} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{g}_b \end{bmatrix} \tag{3.55}$$

In the new set of DoF given here the interface forces $\boldsymbol{g}_b$ are added to the displacements $\boldsymbol{u}$. For simplicity the rigid body and normal modes are combined in $\boldsymbol{\Phi}$. The equation of motion can thus be written as:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{b}^T \\ -\boldsymbol{b} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{u}_b \end{bmatrix} \tag{3.56}$$

The second equation in (3.56) may be thought of as redundant. Nonetheless it is added for symmetry and as can be seen in section 2.2.2 it is also used to enforce compatibility during assembly. By projecting these equations onto the reduction basis $\boldsymbol{R}_{DCB}$, the reduced system is obtained.

$$\begin{aligned} \boldsymbol{R}_{DCB}^T \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{R}_{DCB} \begin{bmatrix} \ddot{\boldsymbol{\eta}} \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \boldsymbol{R}_{DCB}^T \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{b}^T \\ -\boldsymbol{b} & \boldsymbol{0} \end{bmatrix} \boldsymbol{R}_{DCB} \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{g}_b \end{bmatrix} &= \cdots \\ \cdots \boldsymbol{R}_{DCB}^T \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} + \boldsymbol{R}_{DCB}^T \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{u}_b \end{bmatrix} & \end{aligned}$$

Here $\boldsymbol{b}$ is a local Boolean matrix acting on the boundary DoF. By using the properties of the residual flexibility matrix (3.34), the projection onto the reduction basis gives the following reduced matrices:

$$\tilde{\boldsymbol{M}} = \boldsymbol{R}_{DCB}^T \begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \boldsymbol{R}_{DCB} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{res} \end{bmatrix}$$

$$\tilde{\boldsymbol{K}} = \boldsymbol{R}_{DCB}^T \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{b}^T \\ -\boldsymbol{b} & \boldsymbol{0} \end{bmatrix} \boldsymbol{R}_{DCB} = \begin{bmatrix} \boldsymbol{\Omega}^2 & -\begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_f \end{bmatrix}^T \boldsymbol{b}^T \\ -\boldsymbol{b} \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{\Phi}_f \end{bmatrix} & -\boldsymbol{F}_{res} \end{bmatrix} \tag{3.57}$$

$$\begin{aligned} \boldsymbol{M}_{res} &= \boldsymbol{\Psi}_{ar}^T \boldsymbol{M} \boldsymbol{\Psi}_{ar} \\ \boldsymbol{F}_{res} &= \boldsymbol{\Psi}_{ar}^T \boldsymbol{K} \boldsymbol{\Psi}_{ar} - \boldsymbol{\Psi}_{ar}^T \boldsymbol{b}^T - \boldsymbol{b}\boldsymbol{\Psi}_{ar} = \boldsymbol{b}\boldsymbol{\Psi}_{ar} \end{aligned}$$

$\boldsymbol{\Omega}^2$ is a square matrix filled with zeros and on the diagonal the rigid body and free interface eigenfrequencies. Assembly of the reduced substructures now involves "coupling" of the interface forces, thereby resulting in a true dual system (as described in section 2.2.2 and 2.2.3).

One of the big advantages of reducing a substructure using Dual Craig-Bampton is that the reduction basis only partially changes if the interface is altered. One of the ingredients of the reduction basis are the free interface modes, which do not change if the interface is altered and only the set of residual flexibility modes has to be recomputed (i.e. columns of the residual flexibility matrix have to be replaced, removed and/or added). As a result one only has to update the $\boldsymbol{F}_{res}$ and $\boldsymbol{M}_{res}$ parts of the reduced matrices, which is quite cheap since $\boldsymbol{K}^+$ has already been computed. As already mentioned in section 3.5, this is also partially true for the Rubin and Mac Neal methods. The difference however, is that to obtain the final reduction basis for these methods one has to transform the description in interface forces back to a description in interface displacement, which requires an extra computation step as shown in (3.47).

The second line in (3.57), is in fact the compatibility equation and writes:

$$\boldsymbol{b}\left(\boldsymbol{\Phi\eta} + \boldsymbol{\Psi}_{ar}\boldsymbol{g}_b\right) = \boldsymbol{u}_b - \boldsymbol{M}_{res}\ddot{\boldsymbol{g}}_b \tag{3.58}$$

If one neglects the last term in this equation (e.g. $\boldsymbol{M}_{res} = \boldsymbol{0}$), it gives an exact compatibility condition (as in the Mac Neal method, section 3.5.1) $\boldsymbol{bu} = \boldsymbol{u}_b$. But due to the last term in (3.58), one actually allows a small error on the compatibility equation: $\boldsymbol{bu} = \boldsymbol{u}_b + \epsilon$. Residual matrices $\boldsymbol{M}_{res}$ and $\boldsymbol{F}_{res}$ (3.57) are actually the modal mass and flexibility on the interface associated with the eigenmodes that were discarded in the reduction step. The compatibility equation (3.58) will thus allow an incompatibility between the substructures in the space spanned by the discarded normal modes. Note that if one would retain *all* the eigenmodes within the reduction basis $\boldsymbol{R}_{DCB}$, both $\boldsymbol{M}_{res} = \boldsymbol{0}$ and $\boldsymbol{F}_{res} = \boldsymbol{0}$ and the perfect compatibility can then be satisfied. The incompatibility introduced by the reduction will allow for motion which is physically not possible (e.g. relative sliding of the interfaces). If the reduction basis is too poor, the incompatibility can introduce spurious modes and eigenfrequencies in the frequency range of interest. There are basically two ways to get these spurious modes to higher frequencies; enrich the reduction basis with more normal modes and/or enrich the reduction basis with higher order residual modes as described in [29].

3.7 The Mixed Craig-Bampton method

In the previous sections, the Craig-Bampton and Dual Craig-Bampton methods have been discussed. A question that naturally arises is: when should one use the Craig-Bampton method with fixed interface modes and when is its dual counterpart with free interface modes to be preferred? Unfortunately, a definitive answer to this question is hard to find.

Consider now the situation where two components are assembled: component 1 is a very stiff and/or heavy structure whereas substructure 2 is very flexible and/or light. One can imagine that after assembly the stiff structure will behave as if its interface is still quasi-free, while the motion of the flexible structure will be largely dictated through its interface with the stiff structure. Ideally, one would therefore want to reduce substructure 1 using the Dual Craig-Bampton method and component 2 using the regular Craig-Bampton method. However, this can still lead to difficulties where one of the substructures is for example very stiff in one direction (or at one location) while it is relatively flexible in another direction (or at another location). In this case one would want to employ a mix of both methods per substructure, that is, fix some interface DoF in the model reduction while others are left free. In this section a "Mixed Craig-Bampton" method will be introduced which is aiming to combine the best of both worlds. Compared to other mixed boundary CMS methods [30–32] this method is unique in the fact that both the reduction and assembly are performed in a mixed sense.

For the sake of illustration let us consider the assembly of two substructures; the subsequent discussion is equally valid for the assembly of an arbitrary number of components. Applying the Mixed Craig-Bampton method to then comprises the following steps:

1. For both components define the interfaces (choose DoF);
2. Using some criterion, choose free / fixed interface condition for each corresponding pair of interface DoF;
3. Calculate the reduction bases for both components and compute the reduced matrices;
4. Assemble the reduced components in the correct manner.

The definition of interfaces and selection of interface DoF is no different than for other reduction methods. The subsequent steps however are non-standard and will be discussed in detail in the following sections. It should be noted that the reduction of substructures using the Mixed Craig-Bampton method cannot be performed separately, since the reduction basis of one substructure is dependent on the properties of its neighboring components, or at least in terms of global properties.

3.7.1 Selection of Free or Fixed Modes

In order to select fixed or free modes for the reduction basis of the substructures, some criterion must be established. To this end, an *a priori* estimate is needed of the components' behavior. Ideally, one would want to know the response of every interface DoF of both substructures to a unit load, compare these and for both substructures select free or fixed modes. This is however computationally inefficient, hence some approximation is needed. Such an approximation can be made by estimating the substructure behavior by looking only at the value on the diagonal of the stiffness matrix corresponding to the interface DoF. Then, three cases can be distinguished and the following selection scheme is proposed:

- In the first case, subsystem 1 is much stiffer than subsystem 2 i.e:

  $$\frac{K_{ii}^{(1)}}{K_{ii}^{(2)}} > 10^c$$

  where c is some constant that can be chosen to suit the problem at hand. In this case, subsystem 1 will feel some (connection) forces through its interface but will not be influenced very much by the presence of its neighboring substructure. Hence it behaves nearly as if it were free. The motion of substructure 2 will however be largely dictated through its interface with system 1. The natural choice in this case would thus be to let the interface DoF of component 1 free in the reduction, while the corresponding interface DoF of substructure 2 should be fixed. We will denote DoF that remain free by "dual" DoF, while DoF that are fixed are called "primal" DoF.

- In the second case, the stiffness at the interface DoF of both subsystems is of approximately the same order of magnitude, i.e.

  $$10^{-c} \leq \frac{K_{ii}^{(1)}}{K_{ii}^{(2)}} \leq 10^c$$

  In this case, both interface DoF can be reduced with either fixed or free interface modes. The choice for fixed or free modes can be made per set of interface DoF, although a consistent choice for the complete assembly leads to a simpler assembly procedure as will be discussed in section 3.7.3. Furthermore, it is recommended to choose fixed modes for these DoF as this limits the total number of DoF of the assembled system.

- In the third case, subsystem b is much stiffer then a:

  $$\frac{K_{ii}^{(1)}}{K_{ii}^{(2)}} < 10^{-c}$$

  Using the same reasoning as before, the natural choice is to reduce subsystem 2 with free interface modes and system 1 with fixed interface modes.

Using the above selection scheme, the substructure DoF vector $\boldsymbol{u}^{(s)}$ can be partitioned into internal DoF $\boldsymbol{u}_i^{(s)}$, "dual" boundary DoF $\boldsymbol{u}_d^{(s)}$ and "primal" boundary DoF $\boldsymbol{u}_i^{(s)}$, as:

$$\boldsymbol{u}^{(s)} = \begin{bmatrix} \boldsymbol{u}_i^{(s)} & \boldsymbol{u}_d^{(s)} & \boldsymbol{u}_p^{(s)} \end{bmatrix}^T$$

The above division of DoF can now be used to find the reduction basis for this substructure, how this is done will be discussed in the next section.

It should be remarked that the above proposed selection method only works properly if the component interface coincides with the material interface. If this is not the case, the values on the diagonal of the stiffness matrix not truly reflect the "overall" stiffness of the system (imagine for instance a rubber bushing with a metal core). In such situations one should use some other criterion for selecting fixed/free modes or resort to "engineering judgement".

3.7.2 Reduction of Subsystems

Given the partitioning of substructure DoF introduced above, the partitioned equations of motion of a substructure become (damping is neglected and the substructure denotation $^{(s)}$ is omitted for clarity):

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{id} & \boldsymbol{M}_{ip} \\ \boldsymbol{M}_{di} & \boldsymbol{M}_{dd} & \boldsymbol{M}_{dp} \\ \boldsymbol{M}_{pi} & \boldsymbol{M}_{pd} & \boldsymbol{M}_{pp} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i \\ \ddot{\boldsymbol{u}}_d \\ \ddot{\boldsymbol{u}}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{id} & \boldsymbol{K}_{ip} \\ \boldsymbol{K}_{di} & \boldsymbol{K}_{dd} & \boldsymbol{K}_{dp} \\ \boldsymbol{K}_{pi} & \boldsymbol{K}_{pd} & \boldsymbol{K}_{pp} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_d \\ \boldsymbol{u}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_d \\ \boldsymbol{f}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_d \\ \boldsymbol{g}_p \end{bmatrix}$$

It should now be realized that the DoF in $\boldsymbol{u}_d$ will be reduced and assembly will be performed using interface forces. Hence, the interface forces $\boldsymbol{g}_d$ are now included in the DoF vector and an additional equation is added to ensure symmetry of the equations:

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{id} & \boldsymbol{M}_{ip} & \boldsymbol{0} \\ \boldsymbol{M}_{di} & \boldsymbol{M}_{dd} & \boldsymbol{M}_{dp} & \boldsymbol{0} \\ \boldsymbol{M}_{pi} & \boldsymbol{M}_{pd} & \boldsymbol{M}_{pp} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_i \\ \ddot{\boldsymbol{u}}_d \\ \ddot{\boldsymbol{u}}_p \\ \ddot{\boldsymbol{g}}_d \end{bmatrix} + \cdots$$

$$\cdots \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{id} & \boldsymbol{K}_{ip} & \boldsymbol{0} \\ \boldsymbol{K}_{di} & \boldsymbol{K}_{dd} & \boldsymbol{K}_{dp} & -\boldsymbol{I} \\ \boldsymbol{K}_{pi} & \boldsymbol{K}_{pd} & \boldsymbol{K}_{pp} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_d \\ \boldsymbol{u}_p \\ \boldsymbol{g}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{f}_d \\ \boldsymbol{f}_p \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{g}_p \\ -\boldsymbol{u}_d \end{bmatrix}$$

Let us now introduce the DoF set $\boldsymbol{u}_m$, the set of internal plus "dual" boundary DoF, to denote the DoF that will be replaced by generalized DoF in the reduction:

$$\boldsymbol{u}_m = \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_d \end{bmatrix} \quad \text{and} \quad \boldsymbol{b}_m = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}.$$

Here $\boldsymbol{b}_m$ is the associated localization matrix. Using this DoF set, the above partitioned equation of motion can be expressed more compactly as:

$$\begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{mp} & \boldsymbol{0} \\ \boldsymbol{M}_{pm} & \boldsymbol{M}_{pp} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}}_m \\ \ddot{\boldsymbol{u}}_p \\ \ddot{\boldsymbol{g}}_d \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{mp} & -\boldsymbol{b}_m^T \\ \boldsymbol{K}_{pm} & \boldsymbol{K}_{pp} & \boldsymbol{0} \\ -\boldsymbol{b}_m & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_m \\ \boldsymbol{f}_p \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_p \\ -\boldsymbol{u}_d \end{bmatrix} \tag{3.59}$$

As a consequence of the partitioning of DoF, the Mixed Craig-Bampton reduction base in general consists of three ingredients:

- Static constraint modes associated to the interface DoF $\boldsymbol{u}_p$ that are retained (the "primal" DoF);
- Residual flexibility modes associated to the interface DoF $\boldsymbol{u}_d$ that will be left free (the "dual" DoF);
- Fixed/free vibration modes of the structure. The interface DoF that will be reduced in a primal manner will be fixed while the dual interface DoF will be free and participate in the eigenmode computation.

The reduction now consists in approximating the internal and dual DoF by a combination of (a truncated set of) fixed/free vibration modes $\boldsymbol{\Phi}_m$, the constraint modes $\boldsymbol{\Psi}_C$ and the residual flexibility modes $\boldsymbol{\Psi}_{ar}$:

$$\boldsymbol{u}_m \approx \boldsymbol{\Phi}_m \boldsymbol{\eta}_m + \boldsymbol{\Psi}_{ar} \boldsymbol{g}_d + \boldsymbol{\Psi}_C \boldsymbol{u}_p \tag{3.60}$$

The ingredients of the reduction basis can be computed as follows. Firstly, the vibration modes result from solving the fixed/free eigenproblem with the DoF in $\boldsymbol{u}_m$ free and the $\boldsymbol{u}_p$ fixed, so:

$$\left(\boldsymbol{K}_{mm} - \omega^2 \boldsymbol{M}_{mm}\right) \boldsymbol{\Phi}_m = \boldsymbol{0}$$

Note that in case the fixed DoF in $\boldsymbol{u}_p$ do not fully constrain the system, $\boldsymbol{\Phi}_m$ also contains the remaining rigid body modes. Secondly, the constraint modes can be computed by condensing the stiffness matrix to the "primal" DoF, as:

$$\boldsymbol{\Psi}_{C,m} = -\boldsymbol{K}_{mm}^{+} \boldsymbol{K}_{mp}$$

Note that in case a set of primal interface DoF is chosen that constrains the rigid body modes of the substructure, the pseudo-inverse $^+$ becomes a normal inverse. Finally, the residual flexibility modes can be found by:

$$\boldsymbol{\Psi}_{ar} = \boldsymbol{K}_{mm}^{+} \boldsymbol{b}_m^T - \sum_{i=n_r+1}^{k} \frac{\boldsymbol{\Phi}_{m,i} \boldsymbol{\Phi}_{m,i}^T \boldsymbol{b}_m^T}{\omega_{m,i}^2} \tag{3.61}$$

Only the contribution of the flexible modes is taken into account; the possible n_r rigid body modes in $\boldsymbol{\Phi}_m$ do not contribute to the stiffness. Again it should be noted that the pseudo-inverse becomes a normal inverse in case there are no rigid body modes in $\boldsymbol{K}_{mm}$. Note that in the above expression k is the number of retained modes and usually is much smaller than the number of DoF in $\boldsymbol{u}_m$. Next, the Mixed Craig-Bampton reduction matrix can be put in matrix form as:

$$\begin{bmatrix} \boldsymbol{u}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_p \end{bmatrix} = \boldsymbol{R}_{MCB} \begin{bmatrix} \boldsymbol{\eta}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_m & \boldsymbol{\Psi}_C & \boldsymbol{\Psi}_{ar} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_p \end{bmatrix}$$

Using this reduction matrix, the reduced stiffness and mass matrices can be computed. For the stiffness matrix we find:

$$\tilde{\boldsymbol{K}} = \boldsymbol{R}_{MCB}^T \boldsymbol{K} \boldsymbol{R}_{MCB} = \begin{bmatrix} \boldsymbol{\Omega}_m^2 & \boldsymbol{0} & -\boldsymbol{\Phi}_{m,m}^T \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}_{pp} & -\boldsymbol{\Psi}_{C,m}^T \\ -\boldsymbol{\Phi}_{m,m} & -\boldsymbol{\Psi}_{C,m} & -\boldsymbol{F}_{res} \end{bmatrix} \tag{3.62}$$

where

$$\begin{aligned} \tilde{\boldsymbol{K}}_{pp} &= \boldsymbol{K}_{pp} - \boldsymbol{K}_{pm} \boldsymbol{K}_{mm}^{+} \boldsymbol{K}_{mp} \\ \boldsymbol{F}_{res} &= \boldsymbol{b}_m \boldsymbol{\Psi}_{ar} \\ \boldsymbol{\Phi}_{m,m} &= \boldsymbol{b}_m \boldsymbol{\Phi}_m \\ \boldsymbol{\Psi}_{C,m} &= \boldsymbol{b}_m \boldsymbol{\Psi}_C \end{aligned}$$

and $\boldsymbol{\Omega}_m^2$ is a diagonal matrix containing the squares of the fixed/free eigenfrequencies of the system. For the mass matrix we find:

$$\tilde{\boldsymbol{M}} = \boldsymbol{R}_{MCB}^T \boldsymbol{M} \boldsymbol{R}_{MCB} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{M}_{\phi m} & \boldsymbol{0} \\ \boldsymbol{M}_{m\phi} & \tilde{\boldsymbol{M}}_{pp} & \boldsymbol{M}_{m\psi} \\ \boldsymbol{0} & \boldsymbol{M}_{\psi m} & \boldsymbol{M}_{res} \end{bmatrix} \tag{3.63}$$

where

$$\begin{aligned} \boldsymbol{M}_{res} &= \boldsymbol{\Psi}_{ar}^T \boldsymbol{M}_{mm} \boldsymbol{\Psi}_{ar} \\ \boldsymbol{M}_{\phi m} &= \boldsymbol{M}_{m\phi}^T = \boldsymbol{\Phi}_m^T \left(\boldsymbol{M}_{mp} - \boldsymbol{M}_{mm} \boldsymbol{K}_{mm}^+ \boldsymbol{K}_{mp} \right) \\ \boldsymbol{M}_{\psi m} &= \boldsymbol{M}_{m\psi}^T = \boldsymbol{\Psi}_{ar}^T \left(\boldsymbol{M}_{mp} - \boldsymbol{M}_{mm} \boldsymbol{K}_{mm}^+ \boldsymbol{K}_{mp} \right) \\ \tilde{\boldsymbol{M}}_{pp} &= \boldsymbol{M}_{pp} + \boldsymbol{K}_{pm} \boldsymbol{K}_{mm}^+ \boldsymbol{M}_{mm} \boldsymbol{K}_{mm}^+ \boldsymbol{K}_{mp} - \boldsymbol{M}_{pm} \boldsymbol{K}_{mm}^+ \boldsymbol{K}_{mp} - \boldsymbol{K}_{pm} \boldsymbol{K}_{mm}^+ \boldsymbol{M}_{mp} \end{aligned}$$

The reduced equations of motion of a substructure thus become:

$$\begin{aligned} &\begin{bmatrix} \boldsymbol{I} & \boldsymbol{M}_{\phi m} & \boldsymbol{0} \\ \boldsymbol{M}_{m\phi} & \tilde{\boldsymbol{M}}_{pp} & \boldsymbol{M}_{m\psi} \\ \boldsymbol{0} & \boldsymbol{M}_{\psi m} & \boldsymbol{M}_{res} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_m \\ \ddot{\boldsymbol{u}}_p \\ \ddot{\boldsymbol{g}}_d \end{bmatrix} + \cdots \\ &\cdots \begin{bmatrix} \boldsymbol{\Omega}_m^2 & \boldsymbol{0} & -\boldsymbol{\Phi}_{m,m}^T \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}_{pp} & -\boldsymbol{\Psi}_{C,m}^T \\ -\boldsymbol{\Phi}_{m,m} & -\boldsymbol{\Psi}_{C,m} & -\boldsymbol{F}_{res} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_m \\ \boldsymbol{u}_p \\ \boldsymbol{g}_d \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_m^T \boldsymbol{f}_m \\ \boldsymbol{\Psi}_C^T \boldsymbol{f}_m + \boldsymbol{f}_p \\ \boldsymbol{\Psi}_{ar}^T \boldsymbol{f}_m \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_p \\ -\boldsymbol{u}_d \end{bmatrix} \end{aligned} \tag{3.64}$$

From this equation one can clearly see that the Mixed Craig-Bampton method is a true generalization of the original Craig-Bampton and Dual Craig-Bampton methods; if there are no "dual" DoF (i.e. $\boldsymbol{u}_d$ is empty) the reduced matrices are exactly equal to those found with the Craig-Bampton method whereas in the absence of "primal" DoF (i.e. $\boldsymbol{u}_p$ is empty) the matrices become those of the Dual Craig-Bampton method. Practically, one can easily implement this by taking a high value for the parameter c in the selection scheme presented in section 3.7.1 and choosing either fixed or free DoF.

3.7.3 Assembly of Mixed Craig-Bampton Reduced Subsystems

The main challenge with the idea outlined above is that at one side of the interface a physical DoF is still present, while at the other side of the interface is described in terms of (free) mode shapes and interface forces. These are two different physical entities and cannot be assembled directly. To overcome this, other mixed methods [30, 31] transform the "dual" DoF back to interface displacements and assemble the reduced matrices as superelements, thereby implicitly enforcing exact compatibility. Here we take the "fully mixed" approach ; in addition to reducing the substructures using mixed interface modes, the assembly is also performed using mixed DoF (so forces and displacements). It is believed that this is the most natural way to assemble the systems and adds to the strength of mixed boundary reduction methods, since no exact compatibility is required between substructures. Instead, slightly incompatible interface displacements are allowed which makes sense when assembling components that have different reduction bases.

For the sake of illustration let us consider the assembly of two substructures; the subsequent discussion is equally valid for the assembly of an arbitrary number of components. In block diagonal form we can write (not showing the mass matrices for compactness):

$$\cdots + \left[\begin{array}{cc} \left[\begin{array}{ccc} \boldsymbol{\Omega}_m^2 & \boldsymbol{0} & -\boldsymbol{\Phi}_{m,m}^T \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}_{pp} & -\boldsymbol{\Psi}_{C,m}^T \\ -\boldsymbol{\Phi}_{m,m} & -\boldsymbol{\Psi}_{C,m} & -\boldsymbol{F}_{res} \end{array} \right]^{(2)} & \\ & \left[\begin{array}{ccc} \boldsymbol{\Omega}_m^2 & \boldsymbol{0} & -\boldsymbol{\Phi}_{m,m}^T \\ \boldsymbol{0} & \tilde{\boldsymbol{K}}_{pp} & -\boldsymbol{\Psi}_{C,m}^T \\ -\boldsymbol{\Phi}_{m,m} & -\boldsymbol{\Psi}_{C,m} & -\boldsymbol{F}_{res} \end{array} \right]^{(1)} \end{array} \right] \left[\begin{array}{c} \boldsymbol{\eta}_m^{(1)} \\ \boldsymbol{u}_p^{(1)} \\ \boldsymbol{g}_d^{(1)} \\ \boldsymbol{\eta}_m^{(2)} \\ \boldsymbol{u}_p^{(2)} \\ \boldsymbol{g}_d^{(2)} \end{array} \right] \cdots$$

$$\cdots = \left[\begin{array}{c} \boldsymbol{0} \\ \boldsymbol{g}_p^{(1)} \\ -\boldsymbol{u}_d^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{g}_p^{(2)} \\ -\boldsymbol{u}_d^{(2)} \end{array} \right]$$

The DoF sets $\boldsymbol{u}_p^{(s)}$ containing the retained interface DoF (the "primal" boundary DoF) can now be split in a part that is connected to another "primal" DoF and a part that should be connected to a "dual" boundary DoF, i.e. an interface force:

$$\boldsymbol{u}_p = \left[\begin{array}{c} \boldsymbol{u}_{pp} \\ \boldsymbol{u}_{pd} \end{array} \right]$$

The same holds for the set of interface forces of a subsystem; part of these DoF connects to other interface forces while another part connects to displacement DoF:

$$\boldsymbol{g}_d = \left[\begin{array}{c} \boldsymbol{g}_{dd} \\ \boldsymbol{g}_{dp} \end{array} \right]$$

Hence, the total DoF set of a reduced system can be written as:

$$\boldsymbol{q}^{(s)} = \left[\begin{array}{ccccc} \boldsymbol{\eta}_m^{(s)} & \boldsymbol{u}_{pp}^{(s)} & \boldsymbol{u}_{pd}^{(s)} & \boldsymbol{g}_{dd}^{(s)} & \boldsymbol{g}_{dp}^{(s)} \end{array} \right]^T$$

In block diagonal form we can thus write the for the assembly of two substructures:

$$\cdots + \begin{bmatrix} \begin{bmatrix} & & \\ & \tilde{\boldsymbol{K}}^{(1)} & \\ & & \end{bmatrix} & \\ & \begin{bmatrix} & & \\ & \tilde{\boldsymbol{K}}^{(2)} & \\ & & \end{bmatrix} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_m^{(1)} \\ \boldsymbol{u}_{pp}^{(1)} \\ \boldsymbol{u}_{pd}^{(1)} \\ \boldsymbol{g}_{dd}^{(1)} \\ \boldsymbol{g}_{dp}^{(1)} \\ \boldsymbol{\eta}_m^{(2)} \\ \boldsymbol{u}_{pp}^{(2)} \\ \boldsymbol{u}_{pd}^{(2)} \\ \boldsymbol{g}_{dd}^{(2)} \\ \boldsymbol{g}_{dp}^{(2)} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \\ \tilde{\boldsymbol{f}}^{(1)} \\ \end{bmatrix} \\ \begin{bmatrix} \\ \tilde{\boldsymbol{f}}^{(2)} \\ \end{bmatrix} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_{pp}^{(1)} \\ \boldsymbol{g}_{pd}^{(1)} \\ -\boldsymbol{u}_{dd}^{(1)} \\ -\boldsymbol{u}_{dp}^{(1)} \\ \boldsymbol{0} \\ \boldsymbol{g}_{pp}^{(2)} \\ \boldsymbol{g}_{pd}^{(2)} \\ -\boldsymbol{u}_{dd}^{(2)} \\ -\boldsymbol{u}_{dp}^{(2)} \end{bmatrix}$$

where the mass matrix is not shown for compactness. It can thus be seen that in order to assemble the two systems, the following DoF should be connected:

$$\begin{aligned} \boldsymbol{u}_{pp}^{(1)} &\leftrightarrow \boldsymbol{u}_{pp}^{(2)} \\ \boldsymbol{g}_{dd}^{(1)} &\leftrightarrow \boldsymbol{g}_{dd}^{(2)} \\ \boldsymbol{u}_{pd}^{(1)} &\leftrightarrow \boldsymbol{g}_{dp}^{(2)} \\ \boldsymbol{g}_{dp}^{(1)} &\leftrightarrow \boldsymbol{u}_{pd}^{(2)} \end{aligned}$$

The assembly of the $\boldsymbol{u}_{pp}^{(s)}$ and $\boldsymbol{u}_{pp}^{(s)}$ can be done in a straightforward manner, using the techniques treated in sections 2.2.1 and 2.2.2, respectively. The mixed assembly (forces to displacements and vice versa) is somewhat more challenging, but can be accomplished using the method described in section 2.2.3. The result is an assembled system where each substructure is reduced using the most appropriate ingredients given its neighboring substructures.

Finally, note that in case the interface DoF for which it holds that $10^{-c} \leq K_{ii}^{(1)}/K_{ii}^{(2)} \leq 10^{c}$ (i.e. interface DoF pairs with (approximately) equal magnitude of stiffness), either a fixed or free boundary at both interfaces can be chosen. When a consistent choice is made, this can lead to either of two simplifications of the assembly procedure:

- When all these interface DoF are fixed in the reduction base, there will be no DoF in the set $\boldsymbol{g}_{dd}$;
- When all these interface DoF are left free in the reduction base, there will be no DoF in the set $\boldsymbol{u}_{pp}$.

3.8 Summary

In this chapter a number of reduction methods has been presented, generally known as "Component Mode Synthesis" methods. Firstly, the ingredients (the modes) for such

methods have been discussed. In the rest of the chapter these ingredients have been mixed in order to describe a number of well known reduction methods: Guyan reduction, the Craig-Bampton method, the methods of Mac Neal and Rubin, but also the relatively new Dual Craig-Bampton technique. The chapter is ended with the new *Mixed Craig-Bampton* method, which is a generalization of the original Craig-Bampton method and the Dual Craig-Bampton method. Although not yet discussed here, the method shows promising results (section 5.4). In order to give a quick overview of all the CMS methods, they have been summarized in table 3.1.

	Craig-Bampton	Rubin & Mac Neal	Dual Craig-Bampton	Mixed Craig-Bampton
Ingredients	$\boldsymbol{\Psi}_C$, $\boldsymbol{\Phi}_i$	$\boldsymbol{\Psi}_{ar}$, $\boldsymbol{\Phi}_r$, $\boldsymbol{\Phi}_f$	$\boldsymbol{\Psi}_{ar}$, $\boldsymbol{\Phi}_r$, $\boldsymbol{\Phi}_f$	$\boldsymbol{\Psi}_{ar}$, $\boldsymbol{\Psi}_C$, $\boldsymbol{\Phi}_m$
Interface DoF	displacements	displacements	forces	displacements & forces
Compatibility	exact	exact	weakened	partially weakened
Accuracy	+	+ & +/–	+	+
Adaptiveness	–	+/–	++	+
Sparsity	+	–	+	+
Implementation	++	+	+/–	–

Table 3.1: Overview of the discussed CMS methods

Here $\boldsymbol{\Psi}_C$ are the constraint modes (section 3.2.4), $\boldsymbol{\Psi}_{ar}$ the residual attachment modes (section 3.2.5), $\boldsymbol{\Phi}_i$ the fixed interface vibration modes (section 3.2.3), $\boldsymbol{\Phi}_r$ the rigid body modes (section 3.2.2), $\boldsymbol{\Phi}_f$ the free interface vibration modes (section 3.2.1) and $\boldsymbol{\Phi}_m$ the mixed interface vibration modes as described in section 3.7.

CHAPTER 4

Interface modeling techniques

4.1 Introduction

Just as important as accurately reduced substructure models are accurate interface models . In many engineering applications interfaces will not just govern the compatibility between the different components, but will have a significant influence on the dynamic behavior of the total structure. Bolted connections, for example, are often modeled and assembled with an exact compatibility. Nonetheless they could account for a significant part of the total structure's flexibility and damping, due to the limited stiffness of the connection and friction between the substructure's interfaces. One of the big challenges in dynamic substructuring is therefore creating accurate, but simple interface models. Complex interface models could lead to an increase of interface DoF, which will automatically lead to a decrease in a computational efficiency. Furthermore a large number of interface DoF will lead to a large reduction basis.

In this section a number of different interface modeling and assembly techniques will be discussed. Assembly of substructures with additional stiffness and damping on the interface will be discussed in section 4.3, section 4.4 will present an option how to assembly incompatible models, but we will start with the simplest interface model: the rigid interface.

4.2 Rigidified interfaces

In case an interface is located on a stiff part of the substructure, or is relatively small (and stiff) in comparison to the total substructure, one could approximate the behavior of the interface by a local rigid section. This assumption will allow for a description of the interface displacements with six rigid motions only. Rigid interfaces are usually created to enable coupling of substructures with non-conforming meshes, since coupling on the original set of interface DoF will pose difficulties (see section 4.4). It is also done in order to reduce the number of interface DoF. Since the set of original interface DoF can be approximated by a set of only six DoF. This approximation can be described by a projection of the original boundary DoF on the six rigid body motions of the

corresponding interface:

$$\begin{bmatrix} \boldsymbol{u}_{b,1} \\ \boldsymbol{u}_{b,2} \\ \vdots \\ \boldsymbol{u}_{b,n_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{T}_1 \\ \boldsymbol{T}_2 \\ \vdots \\ \boldsymbol{T}_{n_i} \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \\ q_\alpha \\ q_\beta \\ q_\gamma \end{bmatrix} \tag{4.1}$$

here n_i is the number of interface nodes and:

$$\boldsymbol{T}_j = \begin{bmatrix} 1 & 0 & 0 & 0 & -d_{j,z} & d_{j,y} \\ 0 & 1 & 0 & d_{j,z} & 0 & -d_{j,x} \\ 0 & 0 & 1 & -d_{j,y} & d_{j,x} & 0 \end{bmatrix} \quad j = 1 \dots n_i$$

for finite elements which have three (translational) DoF per node, or:

$$\boldsymbol{T}_j = \begin{bmatrix} 1 & 0 & 0 & 0 & -d_{j,z} & d_{j,y} \\ 0 & 1 & 0 & d_{j,z} & 0 & -d_{j,x} \\ 0 & 0 & 1 & -d_{j,y} & d_{j,x} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad j = 1 \dots n_i$$

for finite elements which have six (translational and rotational) DoF per node.

$$\boldsymbol{d}_j = \begin{bmatrix} d_{j,x} \\ d_{j,y} \\ d_{j,z} \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \\ z_j \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

Here $\boldsymbol{u}_{b,j}$ is the vector of boundary DoF associated to an interface node j, $\boldsymbol{T}_j$ the corresponding transformation matrix and $\boldsymbol{d}_j$ the corresponding position vector with respect to a reference node $\boldsymbol{u}_0$. One can see that the approach taken here, is similar to what is done in section 3.2.2 in order to determine the rigid body modes. So, the boundary DoF ($\boldsymbol{u}_b$) are now described by six rigid motions, $\boldsymbol{q}_b$, as:

$$\begin{bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_b \\ \boldsymbol{u}_i \end{bmatrix} = \boldsymbol{R}_r \begin{bmatrix} \boldsymbol{q}_b \\ \boldsymbol{u}_i \end{bmatrix} \tag{4.2}$$

By projecting the stiffness and mass matrix on $\boldsymbol{R}_r$, the stiffness and mass of the interface are condensed onto the single interface node. Rigidifying the interface will locally create an infinitely stiff section. Intuitively one can imagine that this will affect mostly the mode shapes in which this rigid section would previously deform, thereby leading to higher eigenfrequencies for these modes after rigidification. If a substructure has a large number of interfaces and/or interfaces take up a large portion of the substructure's surface, this approach will most likely not be desirable, since rigidification of the interfaces would then lead to a substantial increase of the stiffness of the entire structure. This approach

could also be extended by including local interface modes to the basis given in (4.2) to account for some interface flexibility [33–35], this approach is analog to the general interface reduction as outlined later in section 5.2.

Finally, it can be noted that this is in fact a coordinate transformation of all the interface displacement DoF to a set of generalized coordinates $\begin{bmatrix} q_x & q_y & q_z & q_\alpha & q_\beta & q_\gamma \end{bmatrix}^T$. In the case that the interface is described in terms of interface forces, a similar approach can be taken to transform all the interface force DoF to a set of generalized forces $\begin{bmatrix} F_x & F_y & F_z & M_\alpha & M_\beta & M_\gamma \end{bmatrix}^T$, where F denotes an generalize force and M a generalized moment. This can be interpreted as the dual form of the above discussed interface rigidification.

4.3 Assembly with additional interface stiffness and damping

In many situations, the interface between two substructures is not "perfect". Consider for example two components that have been connected by a bolt, a situation encountered very often in practice. Due to this connection some flexibility and/or damping is introduced on the interface that is not present in the separate components. Many other examples of connections are imaginable where some physics are added to the system simply through the coupling of components. The usual approach is to neglect these interface effects. However, this cannot always be done. Let us therefore investigate this issue in more detail. To this end, consider the coupling of two general substructures as depicted

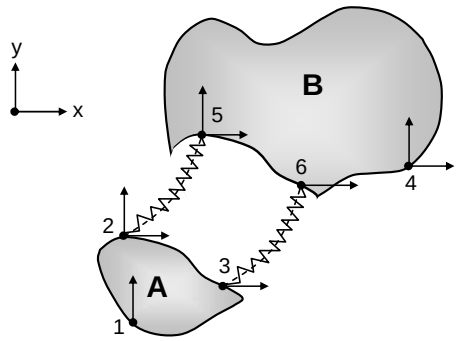


Figure 4.1: Coupling of two general substructures with stiffness on the interface.

in figure A.1. As before, one can write the equations of motion of the subsystems in block diagonal format as:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} + \boldsymbol{g} \tag{4.3}$$

Also, since the springs exert equal forces to both substructures, the equilibrium condition still holds:

$$\boldsymbol{L}^T\boldsymbol{g} = \boldsymbol{0} \tag{4.4}$$

However, due to the interface flexibility the compatibility condition no longer holds. Indeed, due to the flexibility the interface DoF are free to have a relative displacement. This means that two additional equations need to be obtained. One way to eliminate one unknown is to choose the interface forces as:

$$\boldsymbol{g} = -\boldsymbol{B}^T\boldsymbol{\lambda}$$

This way the interface forces in $\boldsymbol{g}$ are chosen such that, due to the construction of the Boolean matrix $\boldsymbol{B}$, the interface forces are always equal and opposite. As before, the Lagrange multipliers $\boldsymbol{\lambda}$ describe the force intensities. Hence, the equilibrium condition is always satisfied, which can be illustrated mathematically since $\boldsymbol{B}^T$ is in the nullspace of $\boldsymbol{L}^T$ and hence

$$\boldsymbol{L}^T\boldsymbol{g} = -\boldsymbol{L}^T\boldsymbol{B}^T\boldsymbol{\lambda} = \boldsymbol{0} \qquad \forall\boldsymbol{\lambda}$$

Now there is still one equation lacking to close the set of equations. However, we know that the spring on the interface behaves such that the interface force intensity can be written as:

$$\boldsymbol{\lambda} = \mathcal{K}_b\Delta\boldsymbol{u}_b$$

Here, $\mathcal{K}_b$ is a matrix containing the stiffness's of the interface springs ($\mathcal{K}_b = \mathrm{diag}\,(\mathrm{k}_1, \mathrm{k}_2)$). From the construction of the Boolean matrix $\boldsymbol{B}$ we also know that:

$$\Delta\boldsymbol{u}_b = \begin{bmatrix} u_2 - u_5 \\ u_3 - u_6 \end{bmatrix} = \boldsymbol{B}\boldsymbol{u}$$

Hence we can write for the Lagrange multipliers

$$\boldsymbol{\lambda} = \mathcal{K}_b\boldsymbol{B}\boldsymbol{u}$$

and subsequently for the connection forces:

$$\boldsymbol{g} = -\boldsymbol{B}^T\mathcal{K}_b\boldsymbol{B}\boldsymbol{u} \tag{4.5}$$

Inserting this expression for the connection forces $\boldsymbol{g}$ into the equations of motion of the subsystems in eq. (4.3) gives the assembled system as:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \left(\boldsymbol{K} + \boldsymbol{B}^T\mathcal{K}_b\boldsymbol{B}\right)\boldsymbol{u} = \boldsymbol{f} \tag{4.6}$$

Note that there is no longer any choice whether to assemble the equations of motion in a dual or primal way; they are automatically assembled by the action of the interface spring. A primal formulation is not possible (since there are no longer redundant interface DoF) and a dual formulation would be trivial. Furthermore, note that assembly of systems with perfect connections can be regarded as a special case of the above situation, namely when $\mathcal{K}_b = \mathrm{diag}\,(\infty, \infty)$. Then $\Delta\boldsymbol{u}_b = \boldsymbol{B}\boldsymbol{u} = \boldsymbol{0}$ and the compatibility condition indeed holds. It is interesting to know that the same formulation is found when one wants to enforce compatibility with a penalty method (not discussed here); the interface stiffness $\mathcal{K}_b$ is then the penalty.

Finally, note that the above is also true when (linear) damping is introduced at the interface. Suppose that in the system in figure 4.1, in addition to the interface stiffness $\boldsymbol{K}_b$, there is interface damping $\mathcal{C}_b$. One can then write for the interface force intensity:

$$\boldsymbol{\lambda} = \mathcal{K}_b \Delta \boldsymbol{u}_b + \mathcal{C}_b \Delta \dot{\boldsymbol{u}}_b$$

As before, this can be written as:

$$\boldsymbol{\lambda} = \mathcal{K}_b \boldsymbol{B}\boldsymbol{u} + \mathcal{C}_b \boldsymbol{B}\dot{\boldsymbol{u}}$$

Hence one can write the equations of motion of the connected systems, with (linear) stiffness and damping effects on the interface, as:

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \left(\boldsymbol{C} + \boldsymbol{B}^T \mathcal{C}_b \boldsymbol{B}\right)\dot{\boldsymbol{u}} + \left(\boldsymbol{K} + \boldsymbol{B}^T \mathcal{K}_b \boldsymbol{B}\right)\boldsymbol{u} = \boldsymbol{f}$$

The approach described above can be easily generalized to systems consisting of multiple substructures with multiple types of interfaces. The total $\boldsymbol{B}$ matrix can be partitioned into interfaces that are perfect (i.e. where the substructures are perfectly connected) and those where flexibility and/or damping is present between the interface DoF:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_f \\ \boldsymbol{B}_p \end{bmatrix}$$

The subscripts $_f$ and $_p$ denote "flexible" and "perfect", respectively. The total system can then be described as:

$$\begin{cases} \boldsymbol{M}\ddot{\boldsymbol{u}} + \left(\boldsymbol{C} + \boldsymbol{B}_f^T \mathcal{C}_b \boldsymbol{B}_f\right)\dot{\boldsymbol{u}} + \left(\boldsymbol{K} + \boldsymbol{B}_f^T \mathcal{K}_b \boldsymbol{B}_f\right)\boldsymbol{u} = \boldsymbol{f} + \boldsymbol{g}_p \\ \boldsymbol{B}_p \boldsymbol{u} = \boldsymbol{0} \\ \boldsymbol{L}_p^T \boldsymbol{g}_p = \boldsymbol{0} \end{cases} \tag{4.7}$$

For the "perfect" interfaces a choice needs still to be made as to assemble the associated DoF in a primal or dual fashion. This is exactly done as described in section 2.2.

4.4 Assembly of component models with non-conforming meshes

One of the benefits of the DS approach is that it allows to combine substructures models created by different engineering groups. These models are often created without any knowledge of, or consideration for, the neighboring substructures, resulting in models with incompatible meshes . Since the models are meshed independently, the nodes at both sides of the interface are usually not collocated (i.e. at the same geometric position) and/or the models are meshed with different types of elements, leading to non-conforming meshes . Global geometric compatibility is usually not an issue, since the geometry of the substructures often originates from one large 3D CAD model.

One approach would be to re-mesh the substructure models such that they become compatible. This leads to an additional computational step and hence reduces the overall

efficiency of the DS strategy. A more efficient approach is to use the interpolation functions of the interface elements in order to enable assembly of non-conforming substructure meshes [36]. In this paper the simple but effective *node collocation method* and its least square variant will be discussed.[1]

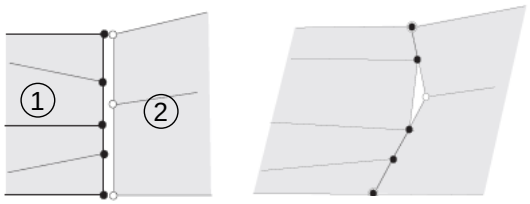


Figure 4.2: Non conforming meshes on the interface [36]

4.4.1 The node collocation method using interface displacements

Suppose two substructures need to be assembled, but the interfaces are not matching as depicted in figure 4.2. One option is to define an intermediate reference interface field $\boldsymbol{u}_b^{(ref)}$ and use the element shape functions of the substructures to interpolate and attach the nodes to the reference interface. This can be expressed as:

$$\boldsymbol{u}_b^{(s)} = \boldsymbol{D}^{(s)} \boldsymbol{u}_b^{(ref)} \tag{4.8}$$

where $\boldsymbol{D}$ is the "collocation" matrix that needs to be computed for both substructures. A special case is obtained if the number of "reference nodes" is taken as the minimum of the number of nodes on each interface. In other words, taking the interface with the smallest number of nodes on the interface as the reference interface field:

$$n_b^{(ref)} \leq min\left(n_b^{(1)}, n_b^{(2)}\right) \tag{4.9}$$

From figure 4.2 it now becomes clear that $\boldsymbol{u}_b^{(2)}$ is the set of *master* interface nodes and $\boldsymbol{u}_b^{(1)}$ is the set of *slave* interface nodes . As a result, $\boldsymbol{D}^{(2)}$ becomes an identity matrix and only the collocation matrix of substructure 1 ($\boldsymbol{D}^{(1)}$) has to be computed. In the collocation method the matrix $\boldsymbol{D}^{(1)} = \boldsymbol{D}$ contains the values of the shape functions on the interface of substructure 2 at the locations of the interface nodes on substructure 1. This imposes that the nodes of substructure 1 remain on the interface of substructure 2. So:

$$\boldsymbol{u}_b^{(1)} = \boldsymbol{D} \boldsymbol{u}_b^{(2)} \tag{4.10}$$

[1]Note that in the last two decades, the assembly of structural models with non-conforming discretizations has become a research field on its own. An important contribution is the so-called Mortar element methods , as described in [37]. However, it is out of the scope of this work to treat such advanced methods.

The compatibility condition of (2.7) now transforms to:

$$\left[\begin{array}{cc} \boldsymbol{B}^{(1)} & \boldsymbol{D}\boldsymbol{B}^{(2)} \end{array} \right] \left[\begin{array}{c} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \end{array} \right] = \boldsymbol{B}\boldsymbol{u} = \boldsymbol{0} \tag{4.11}$$

The matrices denoted by $\boldsymbol{B}^{(s)}$ are signed local Boolean matrices acting on the set of boundary DoF within the total set of substructure DoF. Since $\boldsymbol{D}$ contains interpolation values between zero and one, the resulting matrix $\boldsymbol{B}$ will clearly no longer be a true Boolean matrix, although the part associated to substructure 1 will still be.

4.4.2 Discrete least-squares compatibility using interface displacements

The interface constraint (4.8) together with condition (4.9) implicitly limits the behavior of the degrees of freedom on the sides of the interface that have more DoF then the number of reference DoF, thereby stiffening the interface behavior. This can also be seen for the condition in (4.11), that is when the coarsest side is chosen as reference. Equation (4.11) requires the nodes of the finest side of the interface to be exactly collocated with the interface on the coarse side as illustrated in figure 4.2. In a primal assembly eq. (4.11) would be satisfied by choosing $\boldsymbol{u}_b^{(2)}$ as the DoF in the global set, $\boldsymbol{u}_b^{(1)}$ being substituted using eq. (4.10). The collocation condition (4.8) or (4.11) however can lead to a severe stiffening of the interface model.

A way to render some flexibility to the interface is to relax the collocation condition. For that we look now at eq. (4.8) as an equation from which the reference DoF must be computed for arbitrary substructure DoF. Obviously, given condition (4.8), this is an overdetermined problem that can only be solved in a least square sense:

$$\frac{\partial}{\partial \boldsymbol{u}_{b,i}^{(ref)}} \left(\left(\boldsymbol{u}_b^{(s)} - \boldsymbol{D}^{(s)} \boldsymbol{u}_{b,i}^{(ref)} \right)^T \left(\boldsymbol{u}_b^{(s)} - \boldsymbol{D}^{(s)} \boldsymbol{u}_{b,i}^{(ref)} \right) \right) = 0 \tag{4.12}$$

for:

$$i = 0 \ldots n_b^{(ref)} \quad s = 1, 2$$

By choosing the interface with the smallest number of nodes as the reference interface ($\boldsymbol{u}_b^{(ref)} = \boldsymbol{u}_b^{(2)}$) and recalling that $\boldsymbol{u}_b^{(s)} = \boldsymbol{B}^{(s)} \boldsymbol{u}^{(s)}$, (4.12) becomes:

$$\frac{\partial}{\partial \boldsymbol{u}_b^{(2)}} \left(\left(\boldsymbol{B}^{(1)} \boldsymbol{u}^{(1)} - \boldsymbol{D}\boldsymbol{B}^{(2)} \boldsymbol{u}^{(2)} \right)^T \left(\boldsymbol{B}^{(1)} \boldsymbol{u}^{(1)} - \boldsymbol{D}\boldsymbol{B}^{(2)} \boldsymbol{u}_b^{(2)} \right) \right) = 0$$

Again, matrices denoted $\boldsymbol{B}^{(s)}$ are signed local Boolean matrices (as described in section 2.2) acting on the set of boundary DoF within the total set of substructure DoF. Now the compatibility condition of (2.7) is found in discrete least squares form as:

$$\left[\begin{array}{cc} \left((\boldsymbol{D}^T \boldsymbol{D})^{-1} \boldsymbol{D}^T \right) \boldsymbol{B}^{(1)} & \boldsymbol{B}^{(2)} \end{array} \right] \left[\begin{array}{c} \boldsymbol{u}^{(1)} \\ \boldsymbol{u}^{(2)} \end{array} \right] = \boldsymbol{B}\boldsymbol{u} = \boldsymbol{0} \tag{4.13}$$

The number of constraints imposed by (4.13) is now equal to the number of DoF on the coarsest side, and not to the number of DoF of the finest side like in (4.10). As a matter of fact $\boldsymbol{u}_b^{(2)}$ can be computed for any arbitrary $\boldsymbol{u}_b^{(1)}$ so that if the interface would be assembled in a primal way one would keep all $\boldsymbol{u}_b^{(1)}$, $\boldsymbol{u}_b^{(2)}$ being eliminated by using (4.13). All $\boldsymbol{u}^{(1)}$ are independent but the DoF in $\boldsymbol{u}^{(2)}$ now should be such that the collocation conditions in (4.10) are satisfied in a least square sense. The compatibility stated in (4.13) will therefore lead to a "best" fit, thus minimizing the interface incompatibility.

Both in the node collocation and in the discrete least square methods only local compatibility at nodes is considered. By doing so one disregards the compatibility error along the interface between the nodes, which leads to bad overall compatibility for non-uniform and highly incompatible meshes. Nonetheless, these methods are still used (also in many commercial software packages) since they are easy to implement and will in general not significantly alter the global dynamic behavior.

4.4.3 Node collocation method using interface forces

When dealing with interface displacements the node collocation matrix $\boldsymbol{D}$ obtained in 4.4.1 is used to find the Boolean matrix $\boldsymbol{B}$ which enforces compatibility. This $\boldsymbol{B}$ can now be used to either assemble the substructures in a primal or dual fashion (see section 2.2.1). When assembling substructures using interface forces (section 2.2.2), the same Boolean matrix $\boldsymbol{B}$ is used (in the case of primal assembly). Recalling (2.13) from section 2.2.1:

$$\boldsymbol{g}^{(s)} = -\boldsymbol{B}^{(s)^T}\boldsymbol{\lambda}$$

The full matrix $\boldsymbol{B}$ is given in (4.11) for the collocated case, this can thus be written as:

$$\begin{bmatrix} \boldsymbol{g}^{(1)} \\ \boldsymbol{g}^{(2)} \end{bmatrix} = -\begin{bmatrix} \boldsymbol{B}^{(1)^T} \\ \left(\boldsymbol{D}\boldsymbol{B}^{(2)}\right)^T \end{bmatrix}\boldsymbol{\lambda} \tag{4.14}$$

By substituting (4.14) into $\tilde{\boldsymbol{L}}$ (2.21), which denotes the unique set of DoF.

$$\begin{bmatrix} \boldsymbol{q}^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{q}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \left[-\boldsymbol{B}^{(1)^T}\right] \\ \boldsymbol{0} & \boldsymbol{0} & \\ \boldsymbol{0} & \boldsymbol{I} & \left[-\left(\boldsymbol{D}\boldsymbol{B}^{(2)}\right)^T\right] \\ \boldsymbol{0} & \boldsymbol{0} & \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^{(1)} \\ \boldsymbol{q}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \tilde{\boldsymbol{L}}\bar{\boldsymbol{q}} \tag{4.15}$$

Here $\boldsymbol{q}^{(s)}$ denotes any possible set of generalized coordinates and $\bar{\boldsymbol{q}}$ denotes the unique set of DoF. Using the now obtained $\hat{\boldsymbol{L}}$ matrix one can perform a primal assembly of two substructures with non-conforming meshes using interface forces. Note that the result of $\boldsymbol{D}\boldsymbol{B}^{(2)}$ will not be a Boolean matrix. In similar fashion one is also able to do a dual assembly of non-matching substructures using interface forces (see section 2.2.2). The method described here is equal to the node collocation method described in section 4.4.1 and will thus have the same issues as mentioned earlier.

4.4.4 Discrete least-squares compatibility using interface forces

Since in the node collocation method the number of interface forces will be equal to the number of DoF on the finest side of the interface, (4.15) gives an overdetermined problem (similar to the node collocation method using interface displacements), which could best be solved in a least-square sense. This leads to the $\boldsymbol{B}$ matrix as given in (4.13), substituting this into (2.21) gives a unique set of DoF.

$$\begin{bmatrix} \boldsymbol{q}^{(1)} \\ \boldsymbol{g}_b^{(1)} \\ \boldsymbol{q}^{(2)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \left[-\left((\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T\boldsymbol{B}^{(1)} \right)^T \right] \\ \boldsymbol{0} & \boldsymbol{0} & \\ \boldsymbol{0} & \boldsymbol{I} & \left[-\boldsymbol{B}^{(2)^T} \right] \\ \boldsymbol{0} & \boldsymbol{0} & \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^{(1)} \\ \boldsymbol{q}^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \tilde{\boldsymbol{L}}\bar{\boldsymbol{q}} \tag{4.16}$$

The number of interface forces is now equal to the number of DoF on the coarse side of the interface, thus giving a "best" fit as already described in section 4.4.1.

4.5 Summary

This chapter described a number of interface modeling and assembly techniques. In section 4.2 the rigidified interface is discussed, section 4.3 contained an approach to include dynamic effects on the interface and finally section 4.4 presented several options on the assembly of non-conforming meshes. These techniques are important in order to be able to apply a true "LEGO approach" in dynamic substructuring. With a "LEGO approach" we mean the ability to use independently created substructures in the DS analysis and thereby enabling the best modeling approach for each substructures and the ability to use existing FE models in the analysis.

CHAPTER 5

Assembled system analysis

5.1 Introduction

As a result of the component modeling, reduction and assembly steps described in the previous chapters, the assembled system is obtained. There are numerous analyses one can perform on this assembled system, from a simple modal analysis to an actual load case simulation using time integration of the reduced set of equations of motion. The main challenge in CMS is how much one can reduce the substructure models while maintaining an accurate description of the dynamic behavior of the individual substructures and the assembled global model. The best, and actually the only, way to answer this question, is to perform a set of validation measurements and compare these with the reduced models. However, an actual validation is out of the scope of this chapter and we will limit ourself to model verification.

As already mentioned, we want to reduce the number of degrees of freedom as much as possible. The classical CMS methods described in chapter 3 already reduces the number of DoF significantly. However, large and complex interfaces can result in large numbers of interface DoF and will lead to a loss of efficiency of the DS approach. One option to obtain an accurate description of the interface effects with less degrees of freedom would be to further reduce the reduced system by means of *interface reduction*. This can be seen as a second reduction step and will be discussed in section 5.2 for interface DoF which are in terms of displacements, but also for interface which are in terms of connecting forces.

A useful method is developed to identify which substructure modes are dominant in the global dynamic behavior. This information can for instance be used to build an "optimal" reduction basis, this is done by including the dominant vibration modes. The method is similar to well known the Modal Assurance Criterion and therefore named Substructure Modal Assurance Criterion (SUMAC) and presented in section 5.3

Finally, the effect of the different reduction methods on the accuracy of the global dynamic behavior is reviewed in section 5.4. This is done by comparing eigenfrequencies and mode shapes between the original (full FE) model and the reduced models. In the first subsection (section 5.4.2) a test structure is presented which will be used for the analyses. In section 5.4 modal analyses will be performed on the assemblies of different reduced substructures and on a reference system (which is the full FE model). The eigenfrequencies of the reduced systems will be compared to the eigenfrequencies of the reference solution and a MAC analysis will be used to correlate the eigenmodes.

5.2 Interface reduction

Complex engineering structures, such as a modern wind turbine, commonly consist of a large number of (structural) components, consequently a large number of interfaces between these components exist. Not all interfaces can be assumed to behave rigidly as in section 4.2; the original set of interface DoF sometimes needs to be retained. If a component contains a large number of such interfaces, the number of interface DoF becomes unacceptably high. This is a problem especially when dealing with reduced substructure models, due to the size of the associated full (instead of sparse) reduction matrices. In this section two interface reduction methods will be presented in order to further reduce the total number of DoF.

5.2.1 Reduction of interface displacements

The first method for interface reduction is suited for the reduction of reduced substructures where the interface DoF are in terms of *interface displacements* [38–41]. This is (usually) the case when dealing with so called *superelements*, such as Craig-Bampton and/or Rubin reduced components (see chapter 3).

Determining the interface behavior generally does not require detailed insight in the component's dynamic behavior; an accurate representation of the static behavior at the interface is often sufficient. A static condensation matrix of the substructure is therefore computed as:

$$\begin{aligned} \boldsymbol{u}_i &= -\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}\boldsymbol{u}_b = \boldsymbol{\Psi}_{C,i}\boldsymbol{u}_b \\ \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\Psi}_{C,i} \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{u}_b = \boldsymbol{\Psi}_C \boldsymbol{u}_b \end{aligned} \tag{5.1}$$

Using the so obtained static constraint modes as a reduction basis, the entire substructure is condensed to the interface DoF, resulting in a generalized mass and stiffness matrix:

$$\boldsymbol{M}_{int}\ddot{\boldsymbol{u}}_b + \boldsymbol{K}_{int}\boldsymbol{u}_b = \boldsymbol{f}_b + \boldsymbol{g}_b \tag{5.2}$$

where

$$\begin{aligned} \boldsymbol{M}_{int} &= \boldsymbol{\Psi}_C^T \boldsymbol{M} \boldsymbol{\Psi}_C \\ \boldsymbol{K}_{int} &= \boldsymbol{\Psi}_C^T \boldsymbol{K} \boldsymbol{\Psi}_C \end{aligned} \tag{5.3}$$

This is exactly equal to Guyan reduction (see section 3.3). If one wants to apply interface reduction to substructures reduced with the Craig-Bampton method, the static condensation is already performed and the results of the static condensation is within $\tilde{\boldsymbol{K}}_{bb}$ and $\tilde{\boldsymbol{M}}_{bb}$ in (3.45), and these can directly be used to substitute for $\boldsymbol{K}_{int}$ and $\boldsymbol{M}_{int}$ as can be seen in (5.3). Thereby eliminating the need for these first steps. This is also true for constrained structures that are reduced using the Rubin method, since it can be

shown that the static modes in Rubin's reduction basis describe the same space as the constraint modes. If the substructure is unconstrained, one can alter the reduction basis to include the constraint modes and allow for a static condensation, this is described in [41].

An interface connects two substructures and hence its dynamic behavior cannot simply be described by a single (unassembled) substructure interface; it is dependent on all substructures participating in this interface. Recalling the primal assembly from section 2.2.1, the condensed stiffness and mass matrices can be assembled. In the case of assembly of two substructures, the equation would write:

$$\bar{\boldsymbol{M}}_{\mathsf{int}}\ddot{\boldsymbol{u}}_b + \bar{\boldsymbol{K}}_{\mathsf{int}}\boldsymbol{u}_b = \bar{\boldsymbol{f}}_b \tag{5.4}$$

where:

$$\bar{\boldsymbol{M}}_{\mathsf{int}} = \boldsymbol{L}_{bb}^T \begin{bmatrix} \boldsymbol{M}_{\mathsf{int}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{\mathsf{int}}^{(2)} \end{bmatrix} \boldsymbol{L}_{bb}$$

$$\bar{\boldsymbol{K}}_{int} = \boldsymbol{L}_{bb}^T \begin{bmatrix} \boldsymbol{K}_{\mathsf{int}}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{\mathsf{int}}^{(2)} \end{bmatrix} \boldsymbol{L}_{bb}$$

Here, $\boldsymbol{L}_{bb}$ is the part of the total Boolean matrix $\boldsymbol{L}$ that operates on the interface DoF (as in section 2.2.1). Since the mass and stiffness is condensed to the interface, (5.4) describes the behavior of the interface. By solving the eigenproblem of the interface equations above, the *interface displacement modes* and *interface displacement eigenfrequencies* are obtained, i.e.:

$$\left(\bar{\boldsymbol{K}}_{\mathsf{int}} - \omega_{u,j}^2 \bar{\boldsymbol{M}}_{\mathsf{int}}\right) \boldsymbol{\phi}_{u,j} = \boldsymbol{0}$$

The obtained interface modes ($\boldsymbol{\Phi}_u$) are mode shapes of the interface displacements. Recalling the principle of mode superposition , the response can also be written as:

$$\boldsymbol{u}_b = \sum_{j=1}^{n_b} \boldsymbol{\phi}_{u,j}\eta_{u,j} \tag{5.5}$$

The interface reduction is performed by only including the first k ($k < n_b$) interface displacement modes ($\boldsymbol{\phi}_{u,j}$) in (5.5). Substituting this in a set of generalized coordinates associated to a reduced substructure, gives:

$$\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_u \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_u \end{bmatrix} = \boldsymbol{R}_u \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_u \end{bmatrix} \tag{5.6}$$

Here $\boldsymbol{q}_i$ are the generalized DoF, that resulted from a classical CMS technique as described in chapter 3 and $\boldsymbol{\eta}_u$ is the set of generalized interface DoF. The interface reduction method described in this section can be an effective way of further reducing the number of DoF for substructures with a large number of interface DoF. Note that interface reduction can also be applied to reduce the interfaces of full FE models. Suppose the

interface reduction basis has been computed, this basis can now be used to further reduce the substructure equation of motion.

$$\boldsymbol{R}_u^T \tilde{\boldsymbol{M}} \boldsymbol{R}_u \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \boldsymbol{R}_u^T \tilde{\boldsymbol{K}} \boldsymbol{R}_u \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R}_u^T \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{f}}_b \end{bmatrix} + \boldsymbol{R}_u^T \begin{bmatrix} \boldsymbol{0} \\ \tilde{\boldsymbol{g}}_b \end{bmatrix} \tag{5.7}$$

Here $\tilde{\boldsymbol{M}}$ and $\tilde{\boldsymbol{K}}$ are the reduced substructure stiffness and mass matrices. The matrix $\boldsymbol{R}_u$ denotes the interface displacements reduction basis. These reduced mass and stiffness matrices can be expanded by using the original reduction basis.

$$\boldsymbol{R}_u^T \boldsymbol{R}^T \boldsymbol{M} \boldsymbol{R} \boldsymbol{R}_u \begin{bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{u}}_b \end{bmatrix} + \boldsymbol{R}_u^T \boldsymbol{R}^T \tilde{\boldsymbol{K}} \boldsymbol{R} \boldsymbol{R}_u \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R}_u^T \boldsymbol{R}^T \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{f}_b \end{bmatrix} + \boldsymbol{R}_u^T \boldsymbol{R}^T \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_b \end{bmatrix}$$

One can thus see that the interface reduction matrix can also be used to first reduce the original reduction basis, thereby leading to a more compact total reduction basis. Since the original reduction basis is in general quite large (due to the high number of interface DoF), the interface reduction significantly reduces the amount of computer memory required for storage and handling.

$$\begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_b \end{bmatrix} = \boldsymbol{R} \boldsymbol{R}_u \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_u \end{bmatrix} = \boldsymbol{R}_{tot} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_u \end{bmatrix} \tag{5.8}$$

Note that all sides of the interface are reduced with the same set of interface displacement modes, the substructures are therefore still compatible in terms of their discretization on the interface.

The interface reduction employs modal truncation in order to reduce the number of interface DoF. As a result, a part of the higher frequency information is discarded, therefore the reduced substructures are no longer statically exact. One option to solve this issue would be to include a residual stiffness mode, containing the static contribution of the higher frequency modes. Another result of the modal truncation is the reduction of the number of DoF to deform in, which, as in any reduction method, leads to stiffening of the overall structure.

5.2.2 Reduction of interface forces

Similar to the interface reduction of substructures where the interface DoF are *displacements* (section 3.6), we would also like to reduce the interface DoF of substructures where the interface DoF are *forces*. The approach taken in this section is quite similar to the one above. The equations of motion of a substructure where the interface forces are included into the set of DoF are repeated:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{K} & -\boldsymbol{b}^T \\ -\boldsymbol{b} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{u}_b \end{bmatrix} \tag{5.9}$$

Again a static condensation on the interface DoF is desired, although this time the interface DoF are forces, not displacements. Assuming that the external forces are equal to zero and neglecting the inertia forces, the first equation of (5.9) writes:

$$\boldsymbol{K}\boldsymbol{u} - \boldsymbol{b}^T\boldsymbol{g}_b = \boldsymbol{0}$$

Rewriting this as static condensation on the interface forces gives:

$$\boldsymbol{u} = \boldsymbol{K}^+\boldsymbol{b}^T\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r = \boldsymbol{G}_f\boldsymbol{b}^T\boldsymbol{g}_b + \boldsymbol{\Phi}_r\boldsymbol{\eta}_r$$

Here $\boldsymbol{G}_f$ is the elastic flexibility matrix as given in section 3.2.5, this matrix is a generalized inverse of the stiffness matrix $\boldsymbol{K}$. Note that if the substructure is unconstrained, the rigid body modes have to be explicitly included, since $\boldsymbol{G}_f$ is orthogonal to the rigid body modes by construction. Using this expression for the nodal displacements, an interface condensation matrix can be build:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_r & \boldsymbol{G}_f\boldsymbol{b}^T \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{g}_b \end{bmatrix} = \boldsymbol{R}_{int} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{g}_b \end{bmatrix} \tag{5.10}$$

Using the now obtained reduction basis to condense the stiffness and mass matrix on the interface gives the reduced system of equations:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{res} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r \\ \ddot{\boldsymbol{g}}_b \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & -\boldsymbol{\Phi}_r^T\boldsymbol{b}^T \\ -\boldsymbol{b}\boldsymbol{\Phi}_r & -\boldsymbol{F}_{res} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r \\ \boldsymbol{g}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{u}_b \end{bmatrix} \tag{5.11}$$

where

$$\boldsymbol{M}_{res} = \boldsymbol{b}\boldsymbol{G}_f\boldsymbol{M}\boldsymbol{G}_f\boldsymbol{b}^T = \boldsymbol{\Psi}_a^T\boldsymbol{M}\boldsymbol{\Psi}_a$$
$$\boldsymbol{F}_{res} = -\boldsymbol{\Psi}_a^T\boldsymbol{K}\boldsymbol{\Psi}_a + \boldsymbol{\Psi}_a^T\boldsymbol{b}^T + \boldsymbol{b}\boldsymbol{\Psi}_a = \boldsymbol{b}\boldsymbol{\Psi}_a$$

If we compare this obtained set of equation to the equations of motions obtained from a Dual Craig-Bampton reduction (section 3.6), it is clear this is a *static* Dual Craig-Bampton system and can therefore be seen as the dual variant of Guyan's method (section 3.3). As already noted in the previous section, an interface connects at least two substructures and hence its dynamic behavior cannot simply be described by a single (unassembled) substructure interface. It is dependent on all substructures participating in this interface. Suppose we want to assembly two substructures whose interfaces are described by forces. From the section on primal assembly using interface forces (section 2.2.2), we know the interface forces on both sides of the interface can be written as a function of one unique set of interface forces ($\boldsymbol{\lambda}$), as:

$$\begin{bmatrix} \boldsymbol{g}_b^{(1)} \\ \boldsymbol{g}_b^{(2)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}^{(1)^T} \\ \boldsymbol{B}^{(2)^T} \end{bmatrix} \boldsymbol{\lambda} = \boldsymbol{B}^T\boldsymbol{\lambda}$$

Assembly, which is described in more detail in section 2.2.2, then gives:

$$\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \bar{\boldsymbol{M}}_{res} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{\eta}}_r^{(1)} \\ \ddot{\boldsymbol{\eta}}_r^{(2)} \\ \ddot{\boldsymbol{\lambda}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_r^{(1)^T}\boldsymbol{B}^{(1)^T} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{\Phi}_r^{(2)^T}\boldsymbol{B}^{(2)^T} \\ \boldsymbol{B}^{(1)}\boldsymbol{\Phi}_r^{(1)} & -\boldsymbol{B}^{(2)}\boldsymbol{\Phi}_r^{(2)} & \bar{\boldsymbol{F}}_{res} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_r^{(1)} \\ \boldsymbol{\eta}_r^{(2)} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}$$

(5.12)

where:

$$\bar{\boldsymbol{M}}_{res} = \boldsymbol{B} \begin{bmatrix} \boldsymbol{M}_{res}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{res}^{(2)} \end{bmatrix} \boldsymbol{B}^T$$

$$\bar{\boldsymbol{F}}_{res} = \boldsymbol{B} \begin{bmatrix} -\boldsymbol{F}_{res}^{(1)} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{F}_{res}^{(2)} \end{bmatrix} \boldsymbol{B}^T$$

The dynamic behavior of the interface is now expressed by (5.12) [1]. It therefore can be solved as an eigenvalue problem, thus obtaining the *interface force modes* and *interface force eigenvalues*, i.e.:

$$\left(\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Phi}_r^{(1)^T} \boldsymbol{B}^{(1)^T} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{\Phi}_r^{(2)^T} \boldsymbol{B}^{(2)^T} \\ \boldsymbol{B}^{(1)} \boldsymbol{\Phi}_r^{(1)} & -\boldsymbol{B}^{(2)} \boldsymbol{\Phi}_r^{(2)} & \bar{\boldsymbol{F}}_{res} \end{bmatrix} - \mu_{int,j} \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \bar{\boldsymbol{M}}_{res} \end{bmatrix} \right) \boldsymbol{\phi}_{int,j} = \boldsymbol{0}$$

Since we are only interested in reducing the set of interface forces and not the set of rigid body modes, we split the obtained interface modes $\boldsymbol{\phi}_{int,j}$ into a part that acts on the rigid body modes and a part that acts on the interface forces, according to:

$$\boldsymbol{\Phi}_{int}^T = \begin{bmatrix} \boldsymbol{\Phi}_{ri}^{(1)^T} & \boldsymbol{\Phi}_{ri}^{(2)^T} & \boldsymbol{\Phi}_{\lambda}^T \end{bmatrix}^T \tag{5.13}$$

The part of the interface modes acting on the modal DoF, is denoted by $\boldsymbol{\Phi}_{ri}^{(s)^T}$ and the part that acts on the interface forces by $\boldsymbol{\Phi}_{\lambda}$, these are the *interface force modes*. The interface force modes are shape functions describing the distribution of the interface forces, as opposed to shape functions of the interface displacements in the previous section. These interface force modes are used in order to write the force response in terms of modes and associating eigenvalues (mode superposition), resulting in:

$$\boldsymbol{\lambda} = \sum_{j=1}^{n_b} \boldsymbol{\phi}_{\lambda,j} \eta_{\lambda,j} \tag{5.14}$$

The interface reduction is performed by only including the first k ($k < n_b$) interface force modes in (5.14). Substituting this in a set of generalized coordinates associated to a reduced substructure, gives:

$$\begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi}_{\lambda} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_{\lambda} \end{bmatrix} = \boldsymbol{R}_{\lambda} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_{\lambda} \end{bmatrix} \tag{5.15}$$

[1] Note that if one starts the interface reduction step with Dual Craig-Bampton reduced substructures, a dynamic interface description is already available. If the approach of this section would be followed, the residual interface flexibility matrix present in the Dual Craig-Bampton matrices will have to be supplemented by the retained free interface vibration modes to obtain the (statically complete) residual interface flexibility matrix. One will thus have to reload the substructure reduction bases and apply (3.33) in a backward fashion or compute a new (static) reduction basis. This will result in additional computational effort. Therefore, the approach taken in practice is therefore to use the assembled Dual Craig-Bampton systems to obtain the interface force modes, instead of the system resulting from the static interface condensation.

The matrix $\boldsymbol{R}_\lambda$ denotes the reduction basis for the interface forces. Similar to what is done in section 5.2.1, one can choose to first reduce the original reduction matrix ($\boldsymbol{R}$) before reducing the original set of equations of motion, as:

$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \boldsymbol{R}\boldsymbol{R}_\lambda \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_\lambda \end{bmatrix} = \boldsymbol{R}_{tot} \begin{bmatrix} \boldsymbol{q}_i \\ \boldsymbol{\eta}_\lambda \end{bmatrix} \tag{5.16}$$

The modal truncation of the *interface displacement modes* (section 5.2.1) leads to less DoF to deform in and will thus stiffen the system, whereas the modal truncation of *interface force modes* reduces the degrees of freedom of the coupling forces. In some cases (e.g. more complex interface deformations) the compatibility can no longer be guaranteed by this reduced set of interface forces, and one thereby allows more freedom of motion for the individual substructures. In fact, one allows some interface incompatibility; since the motions of substructure A are no longer fully coupled with the motion of substructure B this results in "gaps" between the substructures. This incompatibility thus allows the substructure to deform more freely; the interface incompatibility will therefore lead to a softening of the total system.

As already mentioned in section 5.2.1, after interface reduction the static solution is no longer exact, this can be improved by adding residual modes containing the higher order flexibility information.

5.3 Substructure Modal Assurance Criterion (SUMAC)

After a modal analysis on the assembled system, one might be interested in which particular substructure modes are dominant in a certain global mode. In other words, we would like to see some sort of correlation between the substructure modes and the global system modes. A well known method to compute the correlation between modes (between for instance experimentally obtained and FE modes) is the Modal Assurance Criterion (MAC) [42], which will be presented in section 5.3.1. From the MAC analysis we will derive the Substructure Modal Assurance Criterion (SUMAC) in section 5.3.2, which is very similar.

5.3.1 Modal Assurance Criterion (MAC)

The Modal Assurance Criterion quantifies how well two vectors "correspond". A scalar value from 0 to 1 is obtained; if the value is 1 the correlation is exact and if the computation yields a 0 there is no correlation at all. A mathematical explanation of high MAC value is that both vectors have the same direction in a N-dimensional space, irrespective of their amplitudes. A more physical explanation is that both vectors describe the same (mode) shape.

$$\mathrm{MAC} = \frac{|\boldsymbol{\phi}_\mathrm{a}^\mathrm{T}\boldsymbol{\phi}_\mathrm{b}|^2}{\left(\boldsymbol{\phi}_\mathrm{a}^\mathrm{T}\boldsymbol{\phi}_\mathrm{a}\right)\left(\boldsymbol{\phi}_\mathrm{b}^\mathrm{T}\boldsymbol{\phi}_\mathrm{b}\right)} \tag{5.17}$$

The MAC criterion can be computed for any set of (complex or real) vectors that are of the same dimension. The MAC is often used when performing an experimental modal analysis to correlate the modes obtained from a finite element simulation to the identified experimental mode shapes. It will basically tell whether the measured mode is similar to the mode computed in a finite element package.

The MAC analysis is also used in section 5.4 to compare the results of the reduction to the reference solution. The reduced eigenmodes are expanded using the reduction matrices and taken as vector set a and the reference mode shapes are vector set b; in the ideal case this would result in unity entries at the diagonal and small values elsewhere.

5.3.2 Substructure Modal Assurance Criterion (SUMAC)

While the MAC is generally used to check the accuracy of, for instance, a model or a reduction method, the SUMAC is a tool which can be used to check the modal contribution of a certain substructure to the global mode shapes.

$$\text{SUMAC} = \frac{|\boldsymbol{\phi}_{\mathrm{g}}^{(\mathrm{s})^{\mathrm{T}}} \boldsymbol{\phi}_{\mathrm{l}}^{(\mathrm{s})}|^2}{\left(\boldsymbol{\phi}_{\mathrm{g}}^{(\mathrm{s})^{\mathrm{T}}} \boldsymbol{\phi}_{\mathrm{g}}^{(\mathrm{s})}\right)\left(\boldsymbol{\phi}_{\mathrm{l}}^{(\mathrm{s})^{\mathrm{T}}} \boldsymbol{\phi}_{\mathrm{l}}^{(\mathrm{s})}\right)} \tag{5.18}$$

Here, $\boldsymbol{\phi}_g^{(s)}$ is the part of the global mode shape that acts on the DoF of substructure s and $\boldsymbol{\phi}_l^{(s)}$ is the (local) uncoupled mode shape of substructure s. $\boldsymbol{\phi}_g^{(s)}$ is obtained by localization of the global mode shape $\boldsymbol{\phi}_g$ using the Boolean matrix $\boldsymbol{L}$ (or $\tilde{\boldsymbol{L}}$), which is also used in the assembly of the substructures (section 2.2).

$$\begin{bmatrix} \boldsymbol{\phi}_g^{(1)} \\ \boldsymbol{\phi}_g^{(2)} \\ \vdots \\ \boldsymbol{\phi}_g^{(n)} \end{bmatrix} = \boldsymbol{L}\boldsymbol{\phi}_g \tag{5.19}$$

Due to this localization, the participation of a certain substructure to the global mode shape can be extracted from (5.19) and will automatically have the same length as the uncoupled (local) mode shapes of the substructure. A MAC value can now be determined using the extracted part from the expanded global mode shape and the uncoupled (local) mode shape. This procedure is what we will call *Substructure Modal Assurance Criterion* (SUMAC). A value close to 1 will denote a high local mode contribution to the global mode shape and a value close to 0 will denote (almost) no local mode contribution to the global mode shape. One can imagine that the accuracy of the local mode with a high contribution in the global mode, will have a significant effect on the accuracy of the global mode. Hence, the SUMAC can help to identify the substructure modes which are dominant in the global dynamic behavior. This will thus allow for constructing an optimal reduction basis which includes the dominant modes. One can also imagine that significant errors of the dominant substructure modes will result in significant errors in the global dynamic behavior.

5.4 Verification of assembly and reduction methods

In this section the reduction methods described in chapter 3 and this chapter will be applied to the test structure introduced in section 5.4.1. In section 5.4.2 the details of the numerical experiment are given. The results can be found in subsections 5.4.3 to 5.4.4 and this section will end with some conclusions in 5.4.5.

5.4.1 A test structure for Dynamic Substructuring

In order to demonstrate and test the methods presented in this thesis, a test structure has been created. The test structure can be seen in figure 5.1.

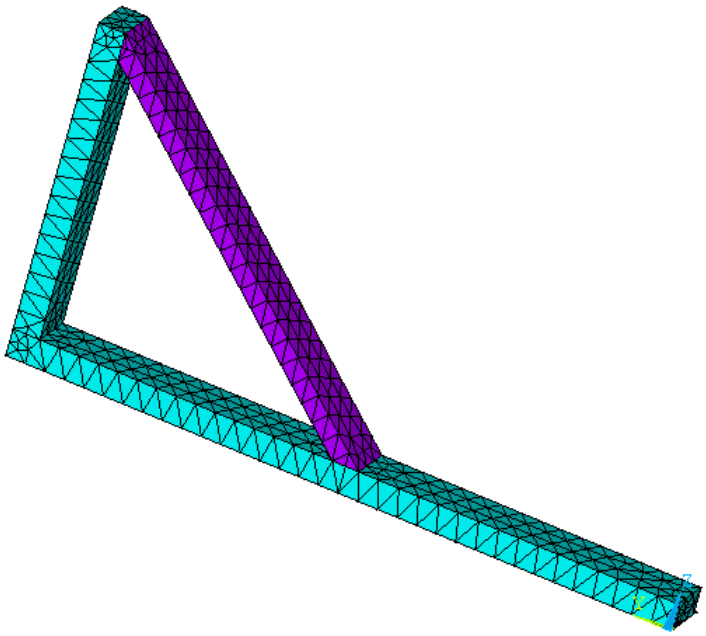


Figure 5.1: DS test structure

The structure is divided into three substructures; the horizontal (beam A), vertical (beam B) and diagonal beam (beam C). On each substructure two interfaces can be identified; one to each neighboring substructure. The horizontal and vertical beam are given the material properties of ordinary construction steel while the diagonal beam is lighter and more flexible. The material properties are:

	Beams A & B	Beam C
Density (ρ)	$7800[kg/m^3]$	$2200[kg/m^3]$
Modulus of elasticity (E)	$210[GPa]$	$21[GPa]$
Poisson's ratio (ν)	0.3	0.3

The mesh has been created in ANSYS using ten-node tetrahedral solid elements resulting in the full model of 10944 degrees of freedom. The model has been exported to MatLab using the FEMLink toolbox; in MatLab all further computations and tests are performed.

5.4.2 Overview of numerical experiments

The purpose of the numerical experiments is:

- to compare the different reduction methods;
- visualize the effect of enriching the reduction basis with more vibrational modes;
- show the results of interface reduction.

Assembly of substructures is performed using the DS Tool , whereas the modal analysis and expansion of mode shapes is done in the PP Tool (see appendix B). The following models are created and compared:

Method	# of vibration modes per substructure	total # of DoF	Abbreviation
Full model	–	10944	–
Craig-Bampton	15	432	CB15
Craig-Bampton	21	450	CB21
Rubin	15	432	R15
Rubin	21	450	R25
Dual Craig Bampton	15	432	DCB15
Dual Craig Bampton	21	450	DCB21
primal Mixed Craig Bampton	15	432	MCBp15
primal Mixed Craig Bampton	21	450	MCBp21
dual Mixed Craig Bampton	15	432	MCBd15
dual Mixed Craig Bampton	21	450	MCBd21

Table 5.1: Overview of the reviewed CMS methods

Firstly, a comparison between the CMS methods described in chapter 3 (table 5.1) is made. Note that the set of vibration modes includes the rigid body modes for the free interface CMS methods (Rubin, Dual Craig-Bampton and possibly Mixed Craig Bampton). In the table, two types of Mixed Craig-Bampton can be seen; the primal and dual. As described in section 3.7, one can choose to reduce the substructures at interfaces with equal stiffness with either free modes (resulting in the dual MCB) or with fixed modes (resulting in the primal MCB), the difference in the results will also be shown.

In the second part the accuracy of the interface reduction methods of section 5.2 will be evaluated. The configurations listed in table 5.2 will be compared to the reference solution.

The two criteria used for the comparison are:

Method	# of vibration modes per substructure	# of interface modes of DoF	# of DoF	Abbreviation
Full model	–	–	10944	–
Craig-Bampton	21	–	450	CB21
Craig-Bampton	21	20	83	CB21-ir20
Craig-Bampton	21	40	103	CB21-ir40
Dual Craig-Bampton	21	–	450	DCB21
Dual Craig-Bampton	21	20	83	DCB21-ir20
Dual Craig-Bampton	21	40	103	DCB21-ir40
Dual Craig-Bampton	21	60	123	DCB21-ir60

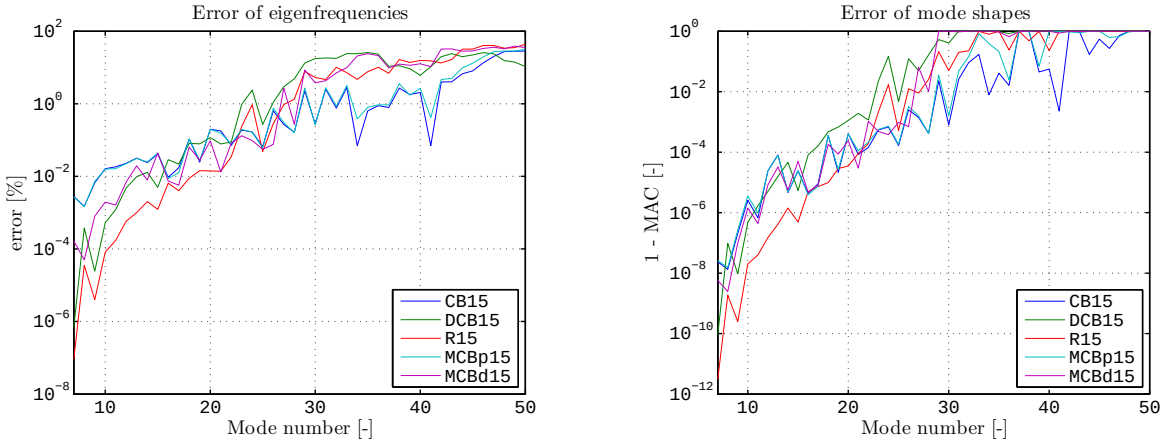
Table 5.2: Overview of the interface reduction configurations

- Frequency error: the frequency error [%] of the eigenfrequencies with respect to the reference solution is given.
- Error of the mode shapes: the MAC matrix is computed to quantify the error of the expanded mode shapes with respect to the reference solution. The MAC values of the diagonal are subtracted from 1 and plotted for each mode.

Next, the results are given in graphs and subsequently discussed in detail.

5.4.3 Model reduction results

The model variants described in table 5.1 are all created and modal analyses are performed.The results of the reduction with 15 vibration modes are shown in figures 5.2 and 5.3 and those using a reduction basis of 21 vibration modes are shown in figure 5.4 and 5.5. Note that the models are reduced by approximately a factor of 25 in comparison to the full model in terms of number of DoF (see table 5.1).



(a) error of the eigenfrequencies (b) error of the mode shapes

Figure 5.2: Comparison of methods using 15 vibration modes

From these results, several observations can be made:

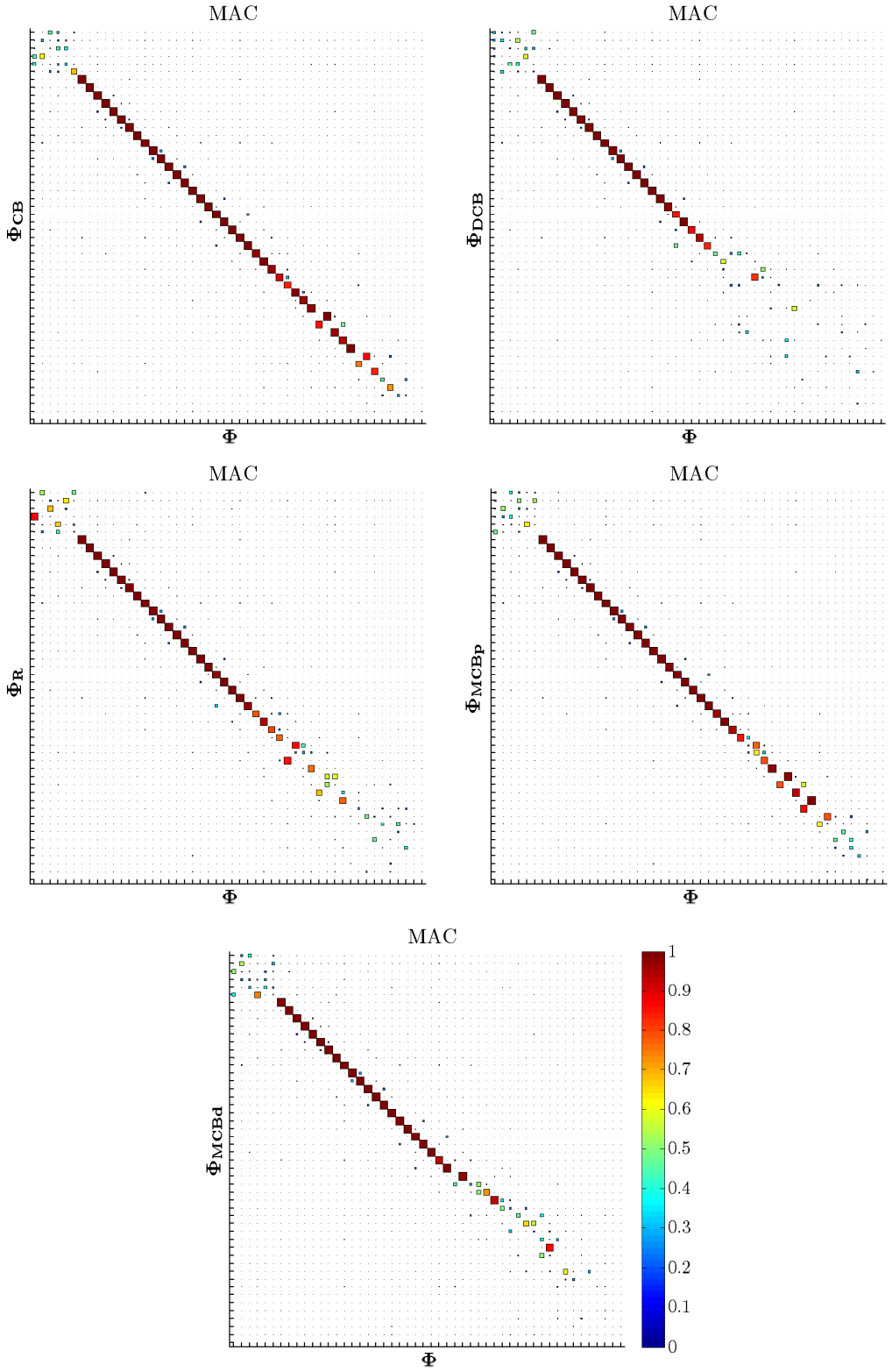


Figure 5.3: Cross MAC between the reduced (and expanded) modes and the reference modes (15 modes)

- **The free interface methods show superior results in the lower frequency range.** It is believed that this is due to the fact that the rigid body modes are explicitly present in the reduction bases. Since the rigid body modes have a significant influence in the low frequency range, the free interface reduced models show very small errors on these first eigenmodes and -frequencies.

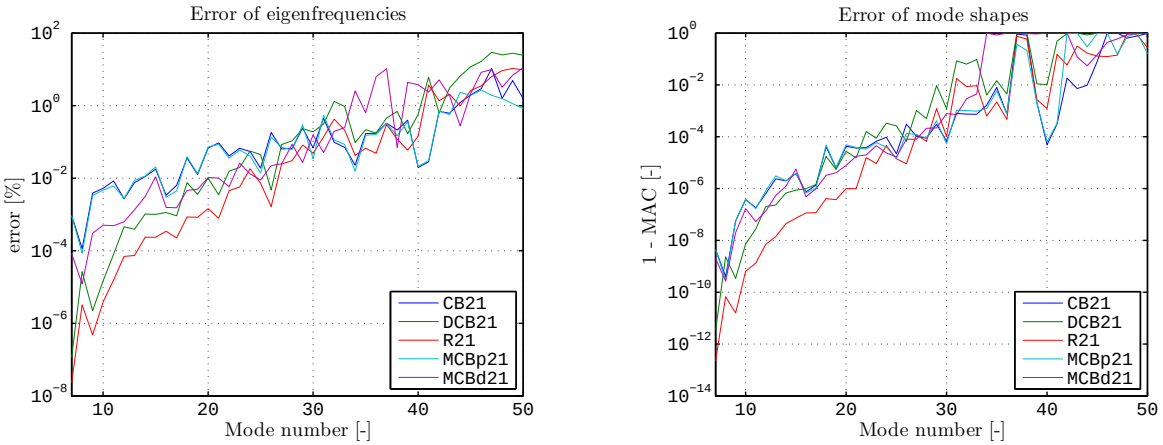
- **The fixed interface methods show superior results in the higher frequency range.** Due to the absence of rigid body modes in the fixed interface reduction bases, there is more vibrational information included in the fixed interface reduction bases. This allows for a better representation of the higher order global mode shapes. It is also thought that the constraint modes create an interface description which is better able to represent the complex interface behavior of the higher modes.

- **The Mixed Craig-Bampton methods show good results.** Both the primal and dual Mixed Craig-Bampton methods show results comparable to the classical CMS methods. One can clearly see that the primal Mixed Craig-Bampton method is almost exactly similar to the Craig-Bampton method and is accurate until mode number 31. The dual Mixed Craig Bampton method, whose reduction bases includes rigid body modes for beams A and B, but none for C, shows very good results in the lower frequency range and is accurate until mode number 27. It can also be seen that the accuracy of both Mixed Craig-Bampton methods is always in between the Craig-Bampton and Dual Craig-Bampton methods, which emphasizes its versatility. Choosing the Mixed Craig-Bampton method thus leads to an accurate description in both the lower and higher frequency range.

- **The classical methods show excellent results.** Rubin's method is most accurate in the lower frequency domain and the method of Craig-Bampton (and the primal Mixed Craig-Bampton) is most accurate in the higher frequency range.

Next, the reduction bases are enriched by six vibration mode shapes. The results are shown in figures 5.4 and 5.5.

Clearly, the comments made above still hold if we enrich the bases with six vibrational modes. An additional comment can be made from the results of the dual Mixed Craig-Bampton reduction: From the figures it can be seen that an extra mode is introduced in the set of (reduced) modes, this can be seen from the fact that a number of off-diagonal high valued entries are present in the MAC figures (5.3 and 5.5). This is a so-called "spurious mode" and is probably caused by the fact that the Dual Craig-Bampton part within the Mixed Craig-Bampton reduction (see section 3.7) allows for some interface incompatibility (as described in section 3.6). In other words, motions are allowed which are physically not possible.



(a) error of the eigenfrequencies

(b) error of the mode shapes

Figure 5.4: Comparison of methods using 21 vibration modes

5.4.4 Interface reduction results

Here the effect of interface reduction (see section 5.2) on the accuracy will be shown. The analyses performed in this section are listed in table 5.2. Interface reduction is applied to substructures reduced using the Craig-Bampton and Dual Craig-Bampton techniques; the substructures are reduced using 21 vibration modes. The interfaces are reduced using 20 and 40 (and 60 for DCB models) interface modes and the results are shown in figures 5.6 and 5.7. By comparing the number of DoF of the reference model and the interface reduced models (table 5.2), we can see that the number of DoF can be reduced by a factor of up to 120.

Again, a number of observations can be made:

- The displacement interface reduction methods show very good results. For example the CB21-ir20 is able to accurately represent the dynamic behavior up to the eigenmode number 30.
- Reduction of the interface forces shows good results. However, they require more interface modes in order to remain accurate for the higher order modes. As can be seen from figure 5.7, applying interface reduction to Dual Craig-Bampton systems can lead to spurious modes in the results (figure 5.7). The Dual Craig-Bampton only enforces a weak form of compatibility which is further weakened by reducing the interface DoF, this will thus allow more interface incompatibility. The spurious modes will lead to the sudden jumps in figure 5.6.
- In the low-frequency range, the DCB and DCB-ir methods are exactly the same. This is due to the fact that a “dynamic” interface reduction is performed instead of a static interface reduction (as discussed in section 5.2.2). So from this particular result no conclusions can be drawn with respect to the difference in the reduction of interface displacements and reduction of interface forces.

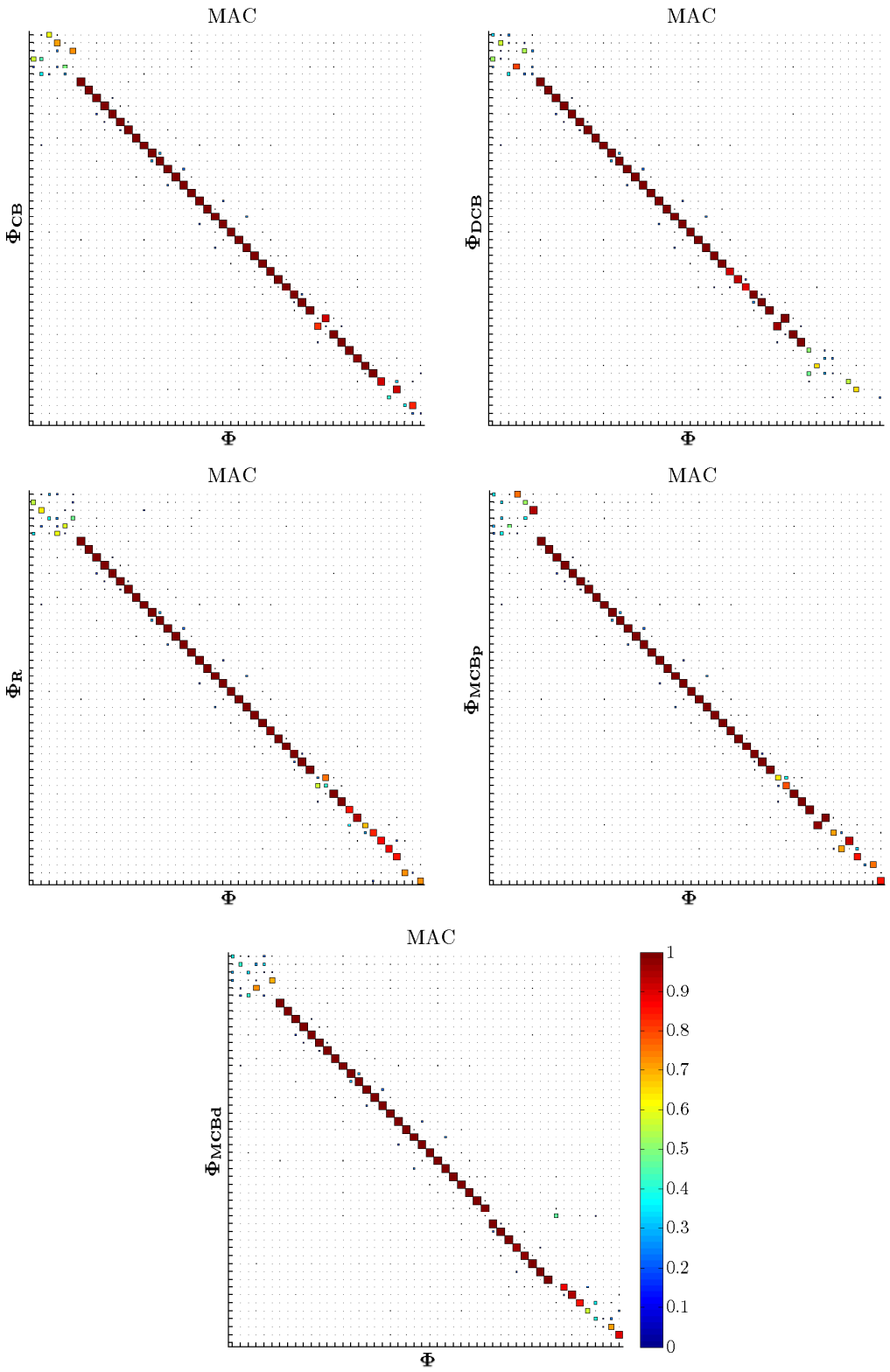
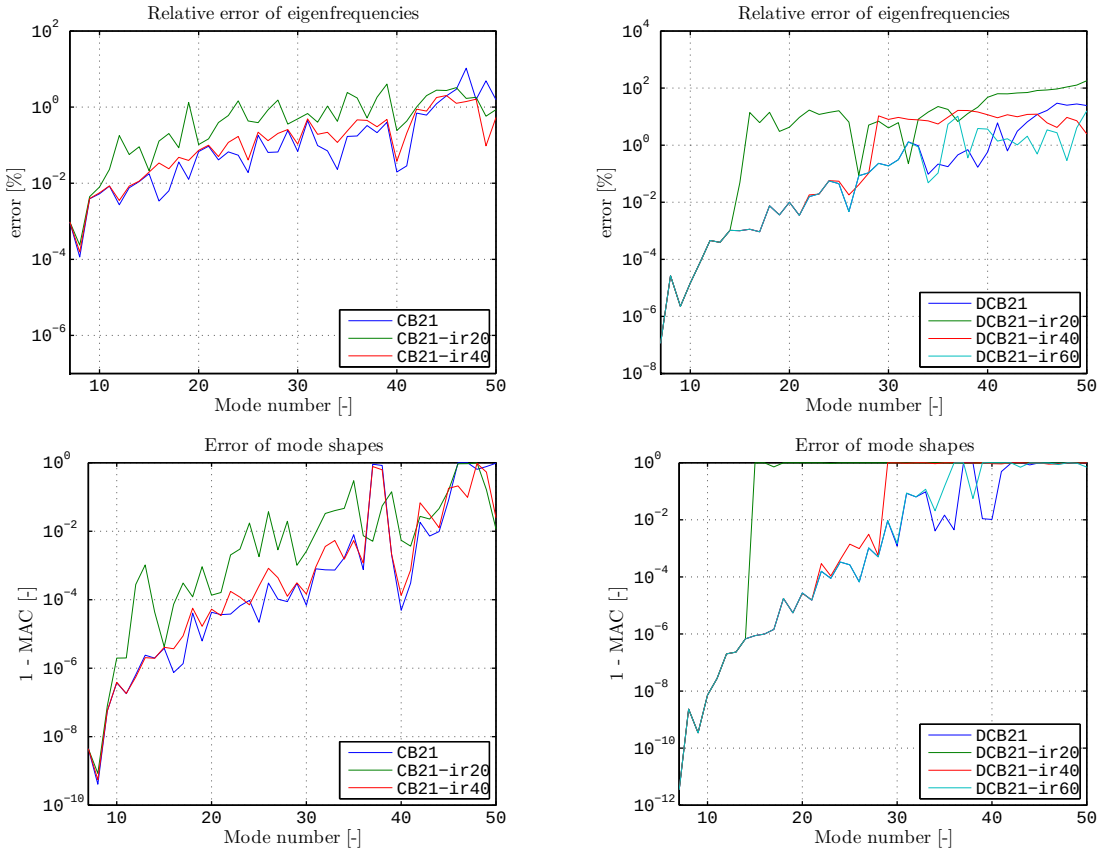


Figure 5.5: Cross MAC between the reduced (and expanded) modes and the reference modes (21 modes)



(a) Difference between CB and CBir (b) Difference between DCB and DCBir

Figure 5.6: Comparison of interface reduction methods

5.4.5 Conclusions

In this section a number of numerical experiments have been performed. A test structure was created on which the different component mode synthesis and interface methods were applied. The eigenfrequencies and modes of the reduced models were compared to the full reference solution and given in sections 5.4.3 and section 5.4.4.

From the results we can draw a number of conclusions:

- All component mode synthesis methods result in accurate and efficient descriptions of the dynamic behavior of the assembled system.
- The free-interface methods show superior results in the low frequency range, whereas the fixed interface methods show better results in the high frequency range.
- The Mixed Craig-Bampton method, introduced in this work, is a true generalization of the Craig-Bampton and Dual Craig-Bampton methods and produces results with similar accuracy.
- The Dual Craig-Bampton and dual Mixed Craig-Bampton methods can result in spurious modes due to the fact that only a weak interface compatibility is enforced.

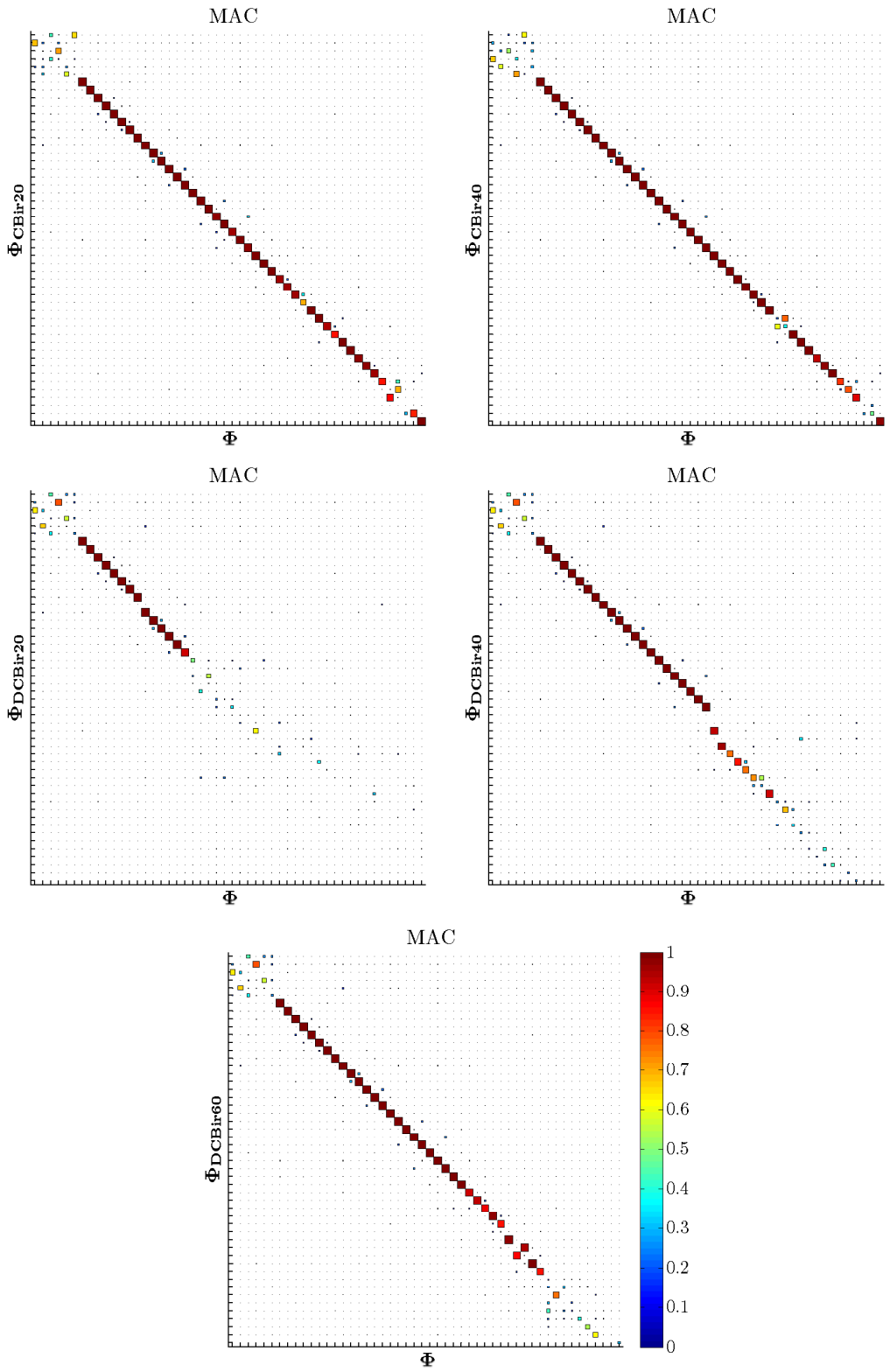


Figure 5.7: Cross MAC between the reduced (and expanded) modes and the reference modes (21 modes)

- By enriching the reduction basis one is able to accurately describe a higher number of eigenmodes of the assembled system.
- Interface reduction leads to an even compacter system of equations. However, a poor interface reduction basis leads to a significant decrease in accuracy, whereas a rich reduction basis limits the effectiveness of the approach. One thus has to find an optimal interface reduction basis in order to obtain a reduced system as compact, but still accurate, as possible.
- Interface reduction applied to Dual Craig-Bampton models can lead to the introduction of (more) spurious modes, since an even weaker interface compatibility is enforced and a cautious approach has to be taken here.

Furthermore, we can conclude that the software tools, created for the reduction, assembly and analysis steps, work excellent.

Part II

Application to a Multi-MW Wind Turbine Yaw System

CHAPTER 6

System description

6.1 Introduction

In society awareness is growing that our current use of natural resources and fossil energy cannot be maintained in the long run. Much attention is given to solutions with which we can maintain our current lifestyle in a sustainable fashion; from recycling to renewable energy sources. One such renewable energy source is wind power, where energy is harvested from the wind and converted to electrical power. In general, all machines able to do so are known as wind turbines. Still, a great variety of different types of wind turbines exists which can be classified as follows:

- Horizontal-axis wind turbine (HAWT) have the main rotor shaft and electrical generator on top of a tower, and must be aligned with the wind. Within the HAWT class, one can identify different subtypes:
  - One, two, three or even more bladed wind turbines.
  - Wind turbines using a gearbox to drive a high speed generator and those that use a low speed generator to which the rotor is directly mounted (direct drive wind turbines).
  - Upwind types , where the rotor is in front of the tower and downwind types where the rotor positions itself behind the tower.

  The most common type of HAWT is the so called "Danish concept" , which is a three bladed, upwind turbine equipped with a high speed generator.
- Vertical-axis wind turbines (VAWT) have the main rotor shaft arranged traverse to the wind stream. Note that the actual orientation of the axis may be at any angle relative to the earth's horizon, even horizontal. In general VAWT have the gearbox and generator near the ground, thereby discarding the need for a tower. Within the VAWT-class a variety of different types of wind turbines exist. However, VAWT's have never been commercially successful.

In general all commercially available multi-MW wind turbines are three bladed, upwind HAWT's with either a high speed or low speed generator. One can imagine that the wind will force to rotor to orient itself behind the tower and orthogonal to the wind direction. The choice for a downwind rotor will thus allow for a passive yaw system, where the rotor will automatically be forced in the correct orientation with respect to the wind direction. This may look beneficial and allow for a relatively simple turbine, this

type of wind turbine will however, endure high cyclic loading which limits the lifetime. During every cycle of the rotor each blade will pass the tower and feel no wind force due to the "tower shadow". This effect is significantly lower if the rotor is in front of the tower, resulting in the upwind configuration. For upwind turbines, passive yawing can be achieved by using a tail vane and a cone-shaped rotor. However, passive yawing can generate high yawing rates, leading to excessive gyroscopic moments on the wind turbine tower. Twisting of the cable that runs from the generator in the nacelle to the transformer in the tower base is also an issue. The problem introduced here is that in order for the rotor to be held orthogonal to the wind direction, the turbine has to be equipped with an active yaw system.

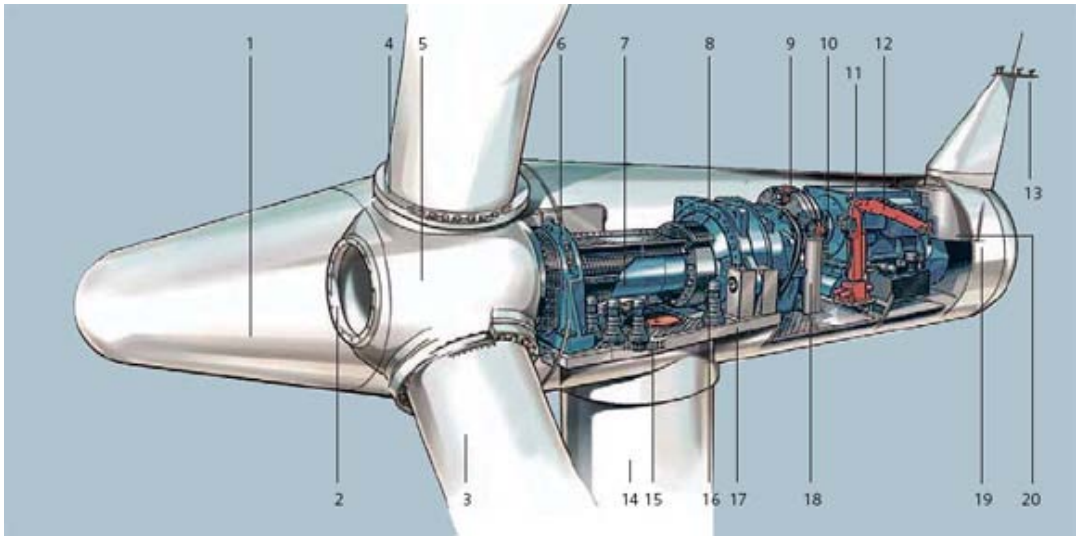
The Siemens wind turbines are no exception to the general design; both the 2.3 MW and 3.6 MW turbines are three bladed upwind HAWT's, equipped with a high speed generator, they are thus equipped with an active yaw system. As already discussed, the yaw system is an important part of every modern wind turbine and is an interesting test case for the DS methodology, since it comprises many components and complex interfaces. Furthermore, the yaw system is generally not taken into account in a detailed way in aero-elastic codes, but is in some cases thought to influence the overall turbine dynamics. The case study will be discussed in more detail in this part, starting in this chapter with a description of the system at hand. First of all, a general description of the Siemens 2.3 MW wind turbine is given in section 6.2. In section 6.3 the yaw system and its system boundaries and components are identified.

6.2 Description of a 2.3 MW Siemens wind turbine

The Siemens 2.3 MW wind turbine is according to the Danish concept and fitted with a variable speed generator. Whereas in the early days of turbine technology the rotor speed was kept constant to generate power at a certain frequency (usually 50 or 60 Hz), nowadays the rotor speed can vary depending on a decrease or increase in wind speed. These turbines are known as variable speed wind turbines. The benefit of variable speed wind turbines is that they can be operated at the optimum energy capture while minimizing the load on the wind turbine for a wide range of wind speeds [43]. As a result of the variable rotor speed, the frequency of the generated electricity is not constant and has to be converted to the right frequency. Therefore the turbine is equipped with a converter in the tower base.

The SWT 2.3 MW can be equipped with different diameter rotors (82 m, 93 m and 101 m) depending on the wind conditions. One can imagine that a turbine at a site with relatively low wind speed requires a larger rotor in order to increase the wind turbine's output than at a site with higher wind speeds. The most common 2.3 MW wind turbine has a rotor with a diameter of 93 meter, the SWT-2.3-93 (Siemens Wind Turbine 2.3 MW, 93 meter rotor).

On a wind turbine one can identify three main parts: the tower, rotor and nacelle . The



1	Spinner	**11**	Generator
2	Spinner Bracket	**12**	Service crane
3	Blade	**13**	Meteorological sensors
4	Pitch bearing	**14**	Tower
5	Rotor hub	**15**	Yaw ring
6	Main bearing	**16**	Yaw gear
7	Main shaft	**17**	Nacelle bedplate
8	Gearbox	**18**	Oil filter
9	Brake disc	**19**	Canopy
10	Coupling	**20**	Generator fan

Figure 6.1: Nacelle arrangement of the 2.3 MW Siemens Wind Turbine

nacelle is basically a big box on top of the tower housing most of the main components (see figure 6.1). Within the nacelle, there a two main (mechanical) subsystems:

- The drive train is the assembly of all mechanical components directly involved in transferring the energy captured by the blades to the generator, which transforms this energy into electrical energy. Main components within the drive train are the main bearing, the low-speed (main) shaft, the gearbox and the generator.
- The yaw system consists of all components of the wind turbine which enable the rotation of the nacelle (and thus rotor) about the tower axis. Since this system is analyzed in this work, it is discussed in more detail in the next section.

6.3 The yaw system and its system boundaries

As already mentioned, yawing denotes the rotation of the nacelle and the rotor about the vertical tower axis. The yaw system of the SWT-2.3-93 is depicted in figure 6.2. In the yaw system of this wind turbine we can identify a number of components:

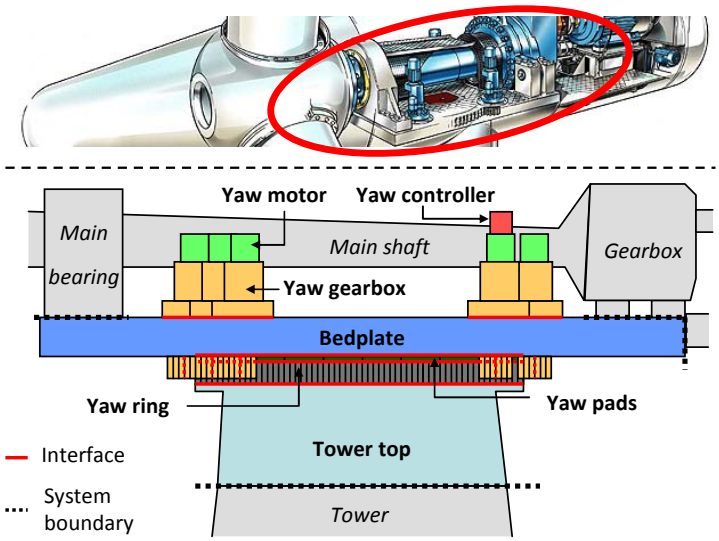


Figure 6.2: Yaw system of a 2.3 MW Siemens wind turbine

- *Bedplate*: The bedplate can be seen as the "chassis" of the nacelle. It serves as a platform for mounting the main turbine components, such as the main gearbox, main bearing, canopy and several smaller parts. Furthermore, the bedplate houses the interface between the tower and the rest of the turbine.
- *Tower top*: The tower top is the upper section of the tapered tower. The tower top flange is welded on top of the tower and bolted to the yaw ring.
- *Yaw ring*: The yaw ring is a big sprocket wheel driven by the yaw gearbox motor assemblies, thereby rotating the entire nacelle. The yaw ring is bolted to the tower top and journaled by the yaw pads.
- *Yaw pads*: The yaw pads are attached to the bedplate and serve as a friction-type bearing for the yaw ring. The yaw pads are made of polyamide material and are lubricated in operation.
- *Yaw gearboxes and motors*: The yaw motors are electric motors controlled by the yaw controller. Through the yaw gearboxes their rotational speeds are greatly reduced, while their torque is increased. This is needed in order to overcome the inertia of the nacelle and the friction of the yaw pads so that the nacelle can be rotated and simultaneously gives low yaw velocities.
- *Yaw controller*: The yaw controller is a central controller for the yaw system and is instructed by the global turbine controller. The yaw controller regulates the rotational speed and torque of the yaw motors.

From figure 6.2 it is clear that all the subsystem components are displayed in color. The grey wind turbine components are outside the system of interest and in between both the system boundaries can be identified:

- The tower and tower top

- The bedplate and the main bearing
- The bedplate and the main gearbox
- The bedplate and the canopy

In this section the components and system boundaries of the yaw system have been briefly described. The yaw system components and interfaces will be discussed in more detail in the next section.

6.4 The yaw system's components and interfaces

The yaw system, its components and the interfaces between the components are shown schematically in figure 6.3. In this section the interfaces in between the yaw system components will be discussed. In the next chapter the component modeling for each substructure will be discussed in more detail.

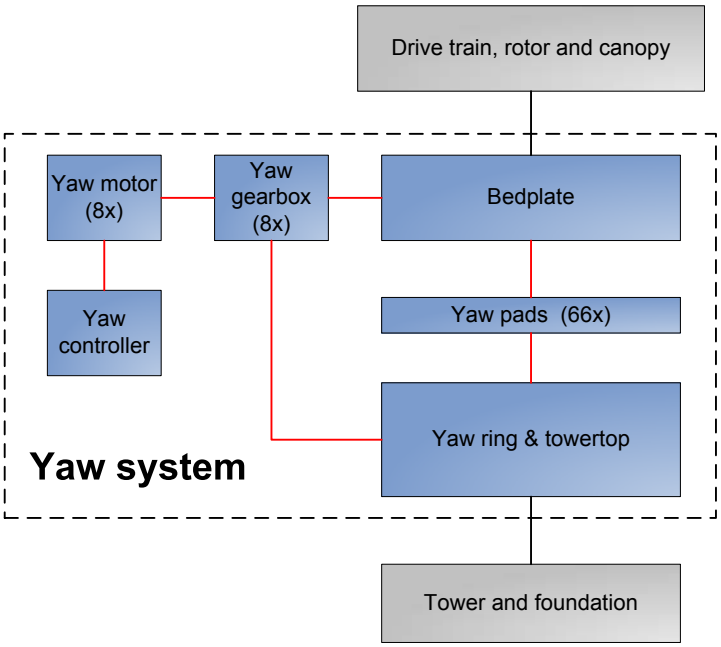
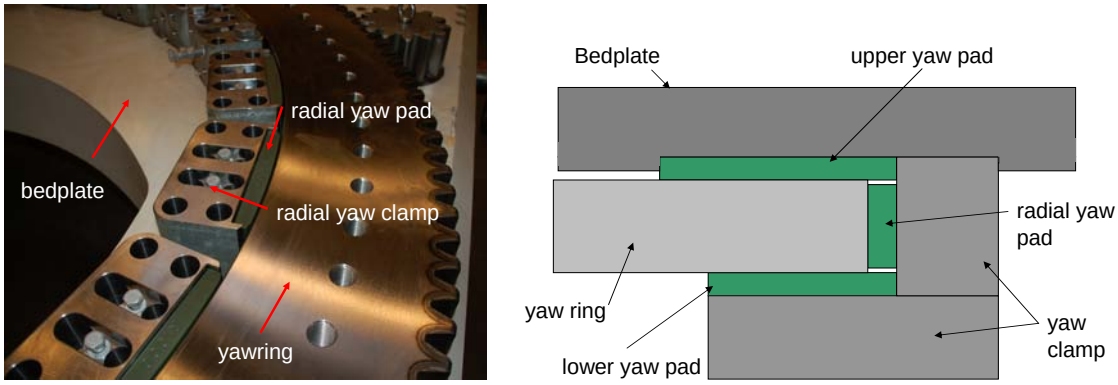


Figure 6.3: Schematic drawing of the yaw system

The identified interfaces are:

- **Interfaces between bedplate and yaw gearboxes.** A total of eight yaw gearboxes are attached to the bedplate. Each gearbox is bolted to the bedplate using 12 high-strength bolts and it is assumed these bolted connections ensure exact compatibility between the bedplate and yaw gearboxes.
- **Connection between yaw pads and bedplate.** In figure 6.4 the configuration of the yaw pads is shown. To mount the set of upper yaw pads (22 pieces in

total), special cavities are milled into the lower side of the bedplate. The yaw pads are secured in the cavities and it is assumed that due to the construction exact compatibility between the bedplate and the set of upper yaw pads is obtained. The set of radial and lower yaw pads (both 22 pieces in total) are mounted in the yaw clamps which are attached to the bedplate. It is assumed that this construction also leads to an exact compatibility between the yaw clamps and the yaw pads. The yaw clamps are included into the bedplate component model.



(a) Photo of a part of the yaw system (b) Schematic lay out of yaw clamp and yaw pads

Figure 6.4: Configuration of the yaw pads

- **Interface between yaw pads and yaw ring** The interface between the yaw ring and yaw pads can be seen in figure 6.4. As described above, the yaw pads create a (lubricated) frictional bearing that allows a rotation about the tower axis. Currently, we consider the case were the interface is in the "stick" regime, that is, the forces exerted are not high enough to overcome the static friction. Which implies that bedplate is fully constrained; i.e. rotation about the tower axis is not possible. This is applied by enforcing exact compatibility between the yaw ring and yaw pads. A second simplification of this interface is made by only including the set of upper yaw pads in the current analysis.

- **Interface between yaw gearboxes and yaw ring** In order for the nacelle to yaw, torque is exerted on the yaw ring by eight yaw gearbox and motor assemblies. The yaw ring and gearboxes are connected through the gear teeth interaction between the yaw ring and the output sprocket pinions of the yaw gearboxes. The interface is modeled by an equivalent gear teeth stiffness (as described in section 7.5.1.1).

- **Connections of yaw motors to yaw gearboxes** The yaw motors are bolted on top of the yaw gearbox housing and the output shaft of the motor is connected to the input shaft of the gearbox. Since the mass of the yaw motor is very small in comparison to the total mass of the yaw system, they are not modeled as separate substructures. However, due to the gear ratio in the yaw gearboxes, it is expected that the inertia of the yaw motor can have a significant effect on the global dynamic behavior. The inertia of the motor is therefore included in the yaw gearbox model.

- **Connection of yaw controller to yaw motors** The yaw controller controls the yaw motors output torque. It is expected that the yaw controllers affect the global dynamic behavior of the yaw system. However, due to time restrictions, they are not included in the current analysis.

As outlined before, a successful DS analysis requires both accurate substructure models and proper interface descriptions. As we will see in the analysis of the yaw system some interfaces will allow "exact" coupling of the substructures (i.e. direct assembly of all interface DoF), while others could show additional effects (see chapter 7).

CHAPTER 7

Component modelling and validation

7.1 Introduction

In chapter 6 the system boundaries have been determined and the yaw system components are identified. The next step is to model the components in order to enable the DS analysis, this is presented in this chapter.

Due to the fact that extensive stress analyses are performed on the structural components of a wind turbine, finite element models of most components are often already available. Furthermore, most components are made from steel and are hence very well suited for FE modeling. Therefore, existing FE models can be used in a DS analysis with only some minor changes, which benefits the practical usability of the DS approach.

In order to gain confidence in the substructure models it is important to validate them using measurements. In a DS analysis one can identify two types of modeling errors; errors in the substructure models and errors in the interface models. The validation strategy taken here is to first validate the substructure models, thereby minimizing the errors in the substructure models, and subsequently assemble the validated substructure models and perform a validation measurement on the assembled structure. One can then use the validation measurement to identify the errors resulting from the interface modeling. Note that this is not possible without validated component models, since it will then be unclear whether the component or interface models caused the errors.

The remainder of this chapter is organized as follows. In section 7.2 the bedplate modeling and model validation are treated, the next section describes the modeling of the tower top and yaw ring. In section 7.4 a model for the yaw pads is presented and finally, in section 7.5 the yaw gearbox model and measurements are discussed.

7.2 Bedplate

The bedplate is a central part of the nacelle and serves as a chassis for all main components of both the yaw system and the drive train. A bedplate is a crucial structure in any wind turbine, since it endures and transmits all the trust forces from the wind and is exposed to continuous variation in loading. Because of the large number of substructures connected to the bedplate, it will have a large number of interfaces. This poses an additional difficulty for the interface modeling since it results in a high number of interface DoF.

7.2.1 Bedplate modeling

Due to its geometry (see figure 7.1) and material properties, the bedplate is very well suited for FE modeling. A CAD model of the bedplate is used to create the FE model. This CAD model contains all the important geometrical information of the bedplate, but also a lot of over-detailed information (e.g. bolt holes). The details in the geometry result in (locally) extremely fine meshes and hence increases the number of DoF of the FE model, while not significantly influencing the global dynamic behavior. Therefore a number of these details have been removed from the CAD model:

- Bolt holes: all bolt holes have been removed from the structure
- Cavities for the yaw pads: in the bedplate shallow cavities are created to mount the yaw pads. The dimensions of these cavities are very small compared to the rest of the structure, hence they are removed from the model.
- Cavities for the yaw gearboxes. Similar to the cavities for the yaw pads, there are also shallow cavities for the yaw gearboxes. These are also removed from the CAD model.

Although the system boundaries were initially set as in figure 6.2, it was later chosen to include the main bearing housing, the connecting points for the main gearbox and the yaw clamps in the bedplate substructure. This leads to more convenient interfaces if one wants to assemble the drive train in a later stadium. In addition the yaw clamps are added to the bedplate; the yaw pads are mounted on the yaw clamps and thereby serve as a bearing between the bedplate and yaw ring. The bedplate is made from a high strength steel with the following homogeneous isotropic material properties:

	Bedplate
Density (ρ)	$7850[kg/m^3]$
Modulus of elasticity (E)	$210[GPa]$
Poisson's ratio (ν)	0.3

The final finite element model of the bedplate substructure is depicted in figure 7.1, meshed using 10-node tetrahedral elements resulting in a finite element model of approximately 130.000 DoF.

As discussed in section 6.4 and as can be seen in figure 6.2, one can identify two interfaces on the bedplate structure:

- *Bedplate* $\leftrightarrow$ *yaw gearbox* This interface is assumed to behave like a rigid section, the interface is therefore "rigidified", as outlined in section 4.2, and coupling is done through a single masternode with six DoF.
- *Bedplate* $\leftrightarrow$ *yaw pad* Rigidification of this interface would significantly stiffen the bedplate model, since the yaw pads cover a large part of the bedplate surface. Hence the interface is modeled fully flexible, i.e. retaining the original set of interface DoF. Since the bedplate and yaw pad substructures are meshed independently

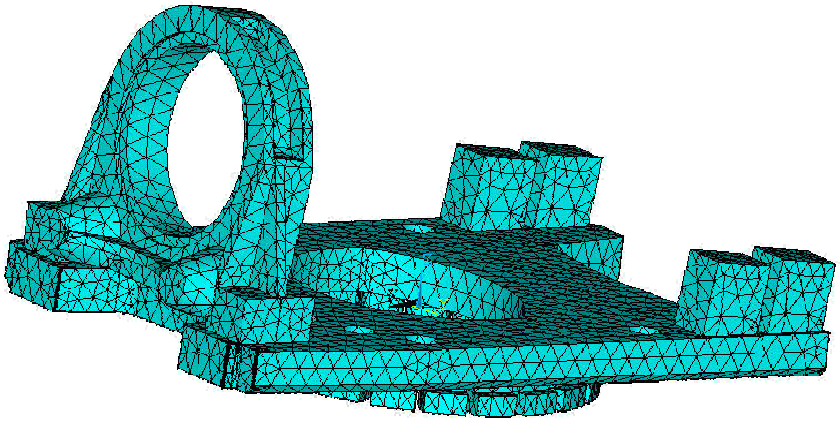


Figure 7.1: Finite element model of the extended bedplate

their meshes are incompatible. To overcome this, the techniques from section 4.4 will be applied.

In addition to these interfaces within the system boundaries, there are two more interfaces at the system boundary, which could be used to assemble a drive train model.

- *Main bearing housing* $\leftrightarrow$ *main bearing* It is assumed this interface will behave rigidly once assembled with the main bearing and main shaft. The housing ring is therefore rigidified and one could include the main bearing as a flexible interface to account for the bearing stiffness (as described in 4.3) between the main bearing housing and main shaft.
- *Bedplate* $\leftrightarrow$ *main gearbox* In order to isolate the gearbox vibrations, the gearbox is suspended in rubber bushings. These rubber bushings can also be replaced by a including flexibility between the gearbox mounts and the gearbox as discussed in section 4.3.

Since these interfaces are not within the current system boundaries, they are not included in the current substructure model of the bedplate.

7.2.2 Bedplate model validation

An experimental modal analysis has been performed to validate the FE model of the bare bedplate, that is, without the main bearing housing, yaw clamps and gearbox mounts. A schematic overview of the measurement setup is given in figure 7.2. The bedplate was suspended using four air springs . These air springs were pressurized at 5.8 bar and created a low stiffness suspension; the rigid body eigenfrequencies were all around 3 Hz and well below the first flexible eigenfrequency. Using a total of nine tri-axial ICP accelerometers , 33 locations were measured in four steps. Excitation of the bedplate was done by a shaker using a random noise signal. Details of the measurement equipment can be found in appendix C.1. The SD Toolbox in MatLab was used to

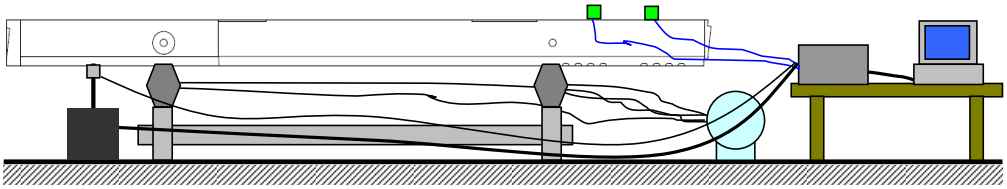


Figure 7.2: Schematic overview of the test setup

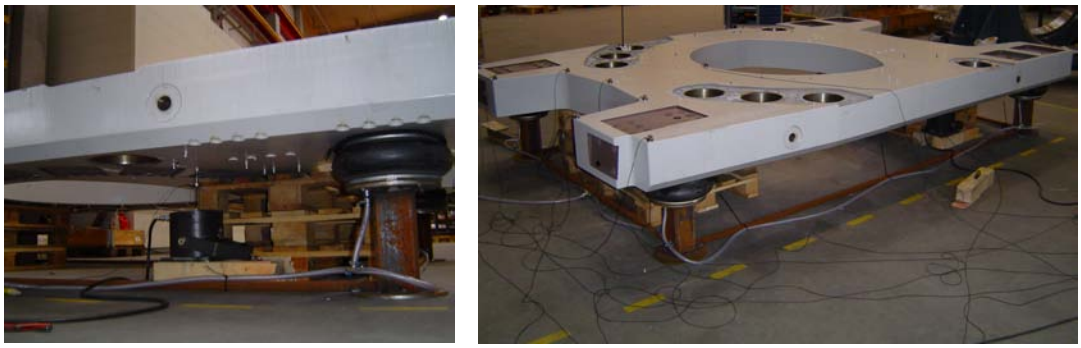


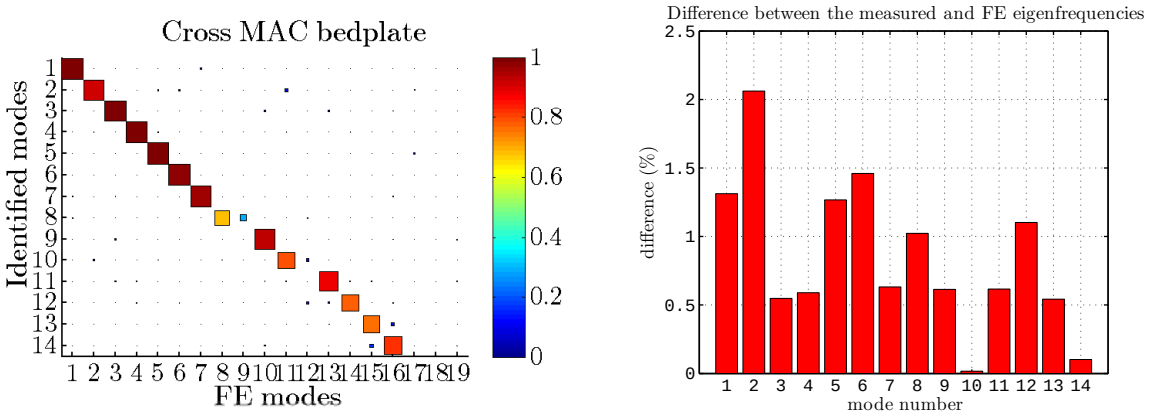
Figure 7.3: Photos of the measurement

identify the eigenmodes and eigenfrequencies. The measured modes were expanded using the SEREP technique [44] and a MAC analysis was performed to visualize correlation between the measured modes and the finite element modes, see figure 7.4 (a). The low cross-correlation at FE mode 8 and mode 9 is due to the fact that both seem to be in-plane modes, whereas the excitation was out-of-plane. Hence, this mode is missing from the set of measured modes. FE mode 10 shows a good correlation to the 9^{th} measured mode. The difference between the measured eigenfrequencies and the FE eigenfrequencies was less than 2%, see figure 7.4 (a).

As described in the modeling section, it was assumed that the interfaces to the yaw gearboxes behave as local rigid sections. In order to validate this assumption, two yaw gearbox interfaces have each been equipped with 4 tri-axial accelerometers during the bedplate measurements, as can be seen in figure 7.5. By projecting the measured FRFs onto the (local) rigid motions and dividing their norm by the norm of the FRFs, a measure for the rigidity is obtained [35]:

$$\text{rigidity} = \frac{\left|\left|\boldsymbol{R}\left(\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}\right)^{-1}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{Y}\right|\right|}{||\boldsymbol{Y}||}100\% \tag{7.1}$$

It can be seen from figure 7.6 that the interfaces of the yaw gearboxes indeed behave rigidly up to a normalized frequency of approximately 0.85, while the frequency range of interest is up to a normalized frequency of 0.5. The rigidities of the interfaces to the main bearing housing and to the main gearbox mounts are also determined and are also shown in figure 7.6. It is clear that the interface to the main gearbox mounts can be assumed to behave rigidly up to a normalized frequency of approximately 0.85, whereas the interface



(a) (b)

Figure 7.4: Results of the bedplate measurement

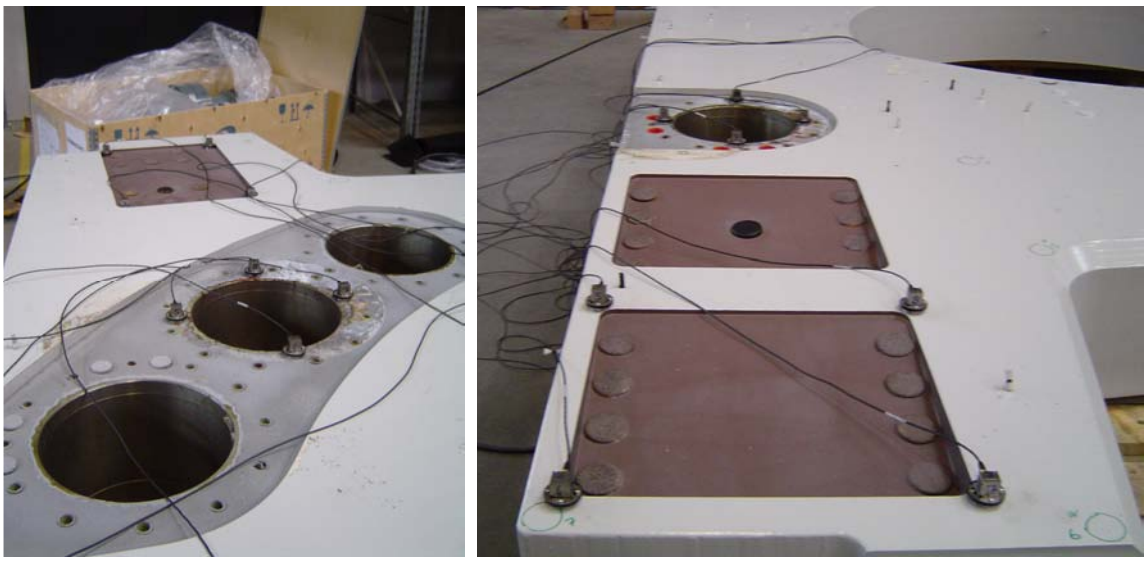
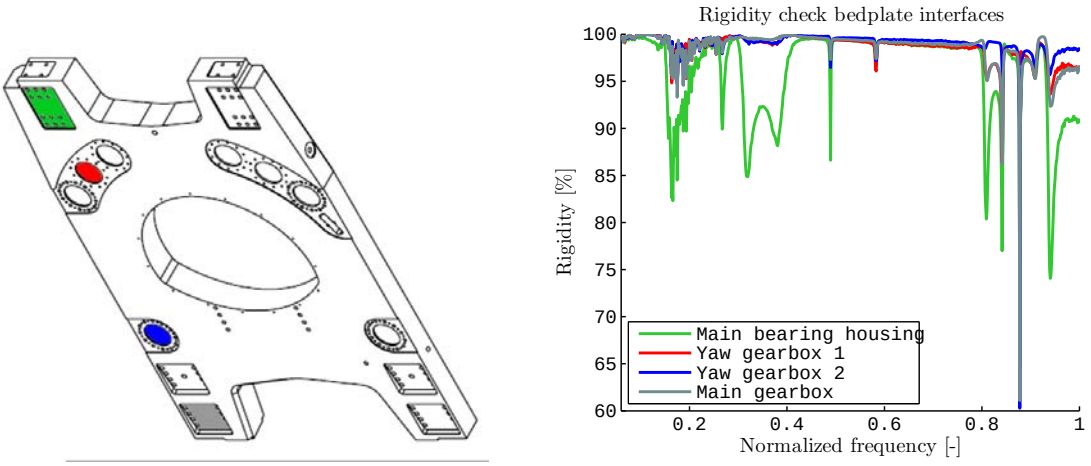


Figure 7.5: Measurements performed for checking the rigidity of the interfaces

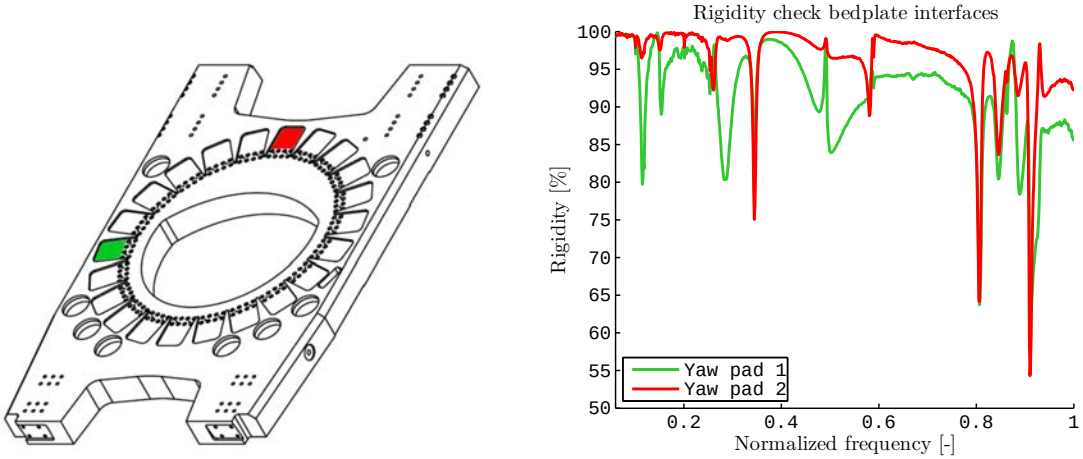
to the main bearing housing clearly shows flexibility within the lower frequency range and can therefore not be considered rigid.

Finally, two of the yaw pad interfaces were measured to determine their rigidity, as can be seen in figure 7.7. It is clear that neither of the yaw pad interfaces can be assumed to behave rigidly and hence need to be modeled as flexible interfaces. From these measurements on the bedplate one can thus conclude that the bedplate FE model itself and the modeling of the interfaces are both valid.



(a) (b)

Figure 7.6: Rigidity of the interfaces on the top surface of the bedplate



(a) (b)

Figure 7.7: Rigidity of the interfaces on the bottom surface of the bedplate

7.3 Tower top and yaw ring

The yaw ring is an important component in the yaw system as can be seen in figure 6.2. As already mentioned, the yaw ring is a big sprocket wheel that is driven by the output pinions of the yaw gearboxes. It is bolted to the tower and journaled to the bedplate by the yaw pads, thereby allowing the yaw gearboxes to generate a torque around the tower axis that results in the yawing motion of the nacelle and rotor. In order to include the stiffening effect of the tower on the yaw ring, the final section of the tower, the tower top, is included in the analysis. A flange is welded to the final section of the tower in order to bolt the yaw ring to the tower, it is assumed this bolted connection ensures exact compatibility between the tower top and the yaw ring, thereby allowing them to be combined to a single substructure.

The yaw ring model is simplified by removing the gear teeth and replacing it by an equivalent ring thickness. This is done since meshing the detailed yaw ring teeth geometry would give an extremely fine mesh size and an unacceptably large number of DoF. The mechanical properties of the substructure are given in table 7.1.

	Yaw ring	Tower top
Density (ρ)	$7830[kg/m^3]$	$7850[kg/m^3]$
Modulus of elasticity (E)	$210[GPa]$	$210[GPa]$
Poisson's ratio (ν)	0.3	0.3

Table 7.1: Yaw ring and tower top material properties

Using the geometries and mechanical properties of the tower top and yaw ring, they are meshed using 10-node tetrahedral solid elements. The tower top however is a cylinder with a small wall thickness, that is suited for meshing with shell elements. Nonetheless it is chosen to mesh it using solid elements for the reason that the number of DoF did not decrease sufficiently to justify the additional effort resulting from the coupling of solid elements to shell elements. The final FE model can be seen in figure 7.8 (a).

From figure 6.2 we see two interfaces for this component:

- *Yaw ring* $\leftrightarrow$ *yaw gearboxes* The interaction between the yaw gearbox output pinion and the yaw ring is through the gear tooth contact. An equivalent gear tooth stiffness has been determined for the connection between the yaw ring and yaw gearbox output pinion using ISO 6336 [45] and the work of Kubur and Peeters [46, 47]. The assembly of these two structures with the interface stiffness is performed as outlined in section 4.3.
- *Yaw ring* $\leftrightarrow$ *yaw pads* This interface behaves fully flexible and therefore the original set of interface DoF is retained. As described in chapter 6, it is assumed that the forces on the interface are not high enough to overcome the static friction. Therefore an exact compatibility between the interface DoF is enforced. Again the meshes on the interfaces between the yaw pads and the yaw ring are non-conforming, this is again solved by using the methods for assembling non-conforming meshes as described in section 4.4.

Due to time limitations the validation of the yaw ring and tower top model is not yet performed.

7.4 Yaw pads

The yaw ring is enclosed by three arrays of yaw pads; one at the top, one at the bottom and one in the inner radius (see figure 6.4). This setup thus constrains the global motion of the bedplate in 5 degrees of freedom and only allows for a global rotation around the tower axis.

The yaw pads are made from a polyamide with a high wear resistance and a low (dynamic) friction coefficient. An important feature of synthetic material is that they usually show frequency dependent behavior and can have a high material damping. Due to this frequency dependent behavior the yaw pad will in fact be a non-linear substructure, which introduces some complex modeling and computational challenges. Since no detailed data of the used polyamide was available and the time did not allow to experimentally determine these, it was decided to approximate their mechanical properties by the modulus of elasticity and Poisson's ratio at 20°C, while the material damping is estimated. It is clear that this approximation is very crude and will give only a limited idea of the substructure's behavior. However, the focus of this work lies in performing a first dynamic substructuring analysis of the yaw system, so creating a non-linear yaw pad model was out of the scope.

In the current analysis only the upper array of yaw pads is included. Since it is assumed that the interface to the yaw ring is in the "stick regime", all interface DoF are coupled and ensure an exact compatibility. It is assumed that the other sets of yaw pads, which are significantly smaller in size, have a negligible effect on the global dynamic behavior. Therefore the other yaw pad arrays are not included in the current analyses, this results in less interface DoF in comparison to the situation when one would include all yaw pads. To allow easy assembly with both the bedplate and the yaw ring, the yaw pads are also meshed using 10-node tetrahedral elements. The yaw pads have an interface at both sides:

- *Yaw pads* ↔ *bedplate* The top side of the yaw pad will be coupled to the bedplate. Since the substructures are created independently, the interface meshes do not match. To overcome this the techniques for assembly of non-conforming meshes are used (section 4.4).
- *Yaw pads* ↔ *yaw ring* The bottom of the yaw pad has an interface with the yaw ring. As already mentioned, an exact compatibility will be enforced here. In addition, the node collocation techniques are also used at this interface since the meshes of the substructures do not match.

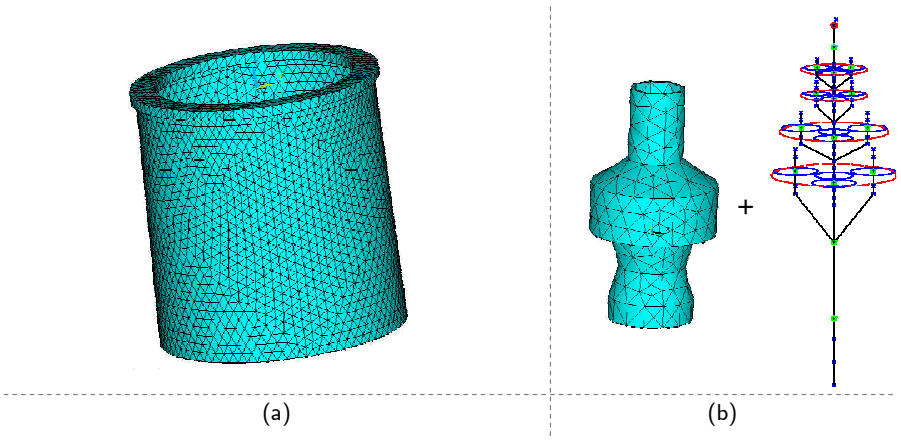


Figure 7.8: (a) Tower top and yaw ring model and (b) model of the yaw gearbox.

7.5 Yaw gearbox

One can imagine the massive loads, especially the torsional moments, associated with yawing a wind turbine. To generate the torque needed to yaw a wind turbine a speed reduction gearbox is essential, making the yaw gearbox a central part of the yaw system of a modern wind turbine. This yaw gearbox converts the high speed/low torque output of the electric yaw motors to low speed/high torque motion at the yaw ring. This avoids the need for large and expensive yaw motors and guarantees a low rotation speed of the nacelle. The latter is important to keep the gyroscopic loads of the yawing rotor to the fixed tower to a minimum. In total, eight yaw motors and yaw gears (when combined often called *yaw drive*) are mounted the 2.3 MW wind turbine. Although the yaw motors are relatively small electric motors, they are able to generate more than 1.000.000 Nm of torque about the tower axis, due to the transmission ratio of the yaw gearboxes.

The gearbox can be divided into two parts; the running gears (internal) and the gearbox housing (external). Both subcomponents are modeled separately and assembled to form the total gearbox model. Furthermore, the final part of this section will discuss the first attempts to determine some parameters to update and possibly validate the gearbox model.

The yaw gearboxes are involved in two interfaces:

- *Yaw gearbox* $\leftrightarrow$ *bedplate* This interface is assumed to behave like a rigid section, the interface is therefore rigidified, as outlined in section 4.2, and coupling is done using a single masternode with six DoF.
- *Yaw gearbox* $\leftrightarrow$ *yaw ring* The interaction between the yaw gearbox output pinion and the yaw ring is through the gear tooth contact. An equivalent gear tooth stiffness has been determined for the connection between the yaw ring and yaw gearbox output pinion using ISO 6336 [45] and the work of Kubur and Peeters [46, 47]. The assembly of these two structures with the interface stiffness is performed as outlined in section 4.3.

The yaw gearbox model is shown in figure 7.8 (b).

7.5.1 Yaw gears

The running gears consist of four planetary gear stages, that result in a final transmission ratio of more than 1000:1. Each stage has 4 planet gears to distribute the torque from the sun gears to the planet carrier. The ring wheel is attached to the housing and hence is stationary.

This section describes in more detail the way the structural dynamics of the yaw gear are modeled in Matlab. A discretised modeling strategy was chosen, as this is the simplest and most flexible way of describing the structural dynamics. This means that the system is discretised into a number of nodes - all having an associated set of degrees of freedom

(DoF) - and elements are defined between these nodes describing the structure's mass and stiffness properties (possibly also damping). This subsection is a summary of the report that describes the yaw gearbox modeling [48].

In order to set up a structural dynamic model, one should first identify all relevant flexibilities and inertias in the gearbox. In this case they are:

- **Internal components**
  The individual components in the gearbox transferring torque will deform under the applied loads. The mass, inertia and stiffness of the shafts, gears and planet carriers are important for the structural dynamics. The structural (mass and stiffness) properties of the shafts and gears are modeled using Euler-Bernoulli beam elements with 6 DoF per node. The planet carriers will be modeled as rigid bodies at first. This seems a reasonable assumption, especially for the high speed stages where the torque is not too large. The inertia properties of the planet carriers must however be taken into account.

- **Bearings**
  The bearing (and the ring wheel) are the interfaces where the running gears and housing are assembled. Although the mass and inertia associated to the bearings can be assumed to be negligible, the stiffness of the bearings probably significantly influences the dynamic behavior and thus needs to be included.

- **Yaw motor**
  The yaw motor drives the input shaft of the yaw gearbox and one can imagine only the inertia of the stator influences the behavior, since the transmission ratio of the yaw gearbox amplifies the inertia. The motor mass is small in comparison to the gearbox mass and therefore neglected. The motor is therefore added as a rotational inertia that acts around the input shaft.

- **Gear teeth interaction**
  The gear teeth interaction accounts for an important part of the running gear flexibility and is also one of the hardest features to model, which will be discussed in more detail in section 7.5.1.1.

In the current gearbox model a number of parameters are not yet known and their values are estimated. The parameters need to obtained in order for the gearbox model to be able to accurately describe the yaw gearbox dynamic behavior. The unknown parameters are:

- **Bearing stiffness**: The stiffness of the bearing connecting the internal gears to the housing is not yet known and estimated.

- **Values for friction and damping**: The energy dissipation and the distribution of damping and friction in the yaw gearbox is not known and neglected. Furthermore, damping resulting from the oil needed to lubricate the gearbox in operation is also neglected.

- **Inertia of the stages**: The inertias of the gears and planet carriers of the different

stages are estimated by their geometry and have to be determined from a detailed CAD-model or measurements.

From these points given here, it is clear that the yaw gearbox model still requires updating. Measurements have been performed to try and find a number of these parameters, these measurements are described in section 7.5.3.

7.5.1.1 Gear interaction

The main challenge in building the yaw gear model was modeling the gear teeth interaction. In order to keep the model relatively simple and avoid the need for modeling the gear teeth contact in detail, a number of assumptions were made:

- The gear teeth stiffness is linear and time invariant. Varying stiffness effects due to changing numbers of gear teeth in contact are assumed to be small and hence neglected.
- Sliding of gear teeth is neglected, so no friction forces are taken into account.
- The gear teeth are assumed to be in contact at all times (no play). Impact forces (backlash) are thus not included.
- Since the yaw gear is constructed mostly of steel, damping is assumed to be small and thus neglected. However, the lubrication (oil) of the gears probably adds damping.
- Gyroscopic effects are at first neglected, but can be added in a later stage in a linearized way (as a function of rotation speed ω). These gyroscopic forces are probably only relevant for the high speed side of the gearbox.
- In the derivation of the gear teeth stiffness matrix, the displacements and rotations are assumed to be small. This simplifies the analysis and results in a linear stiffness matrix. However, in a later stage the formulation can be adapted to allow for large rigid rotations.

Based on these assumptions a model for the gear interaction can be constructed by modeling the gear teeth as a three dimensional linear spring. One can now derive a "gear stiffness element", based on the schematic drawing in figure 7.9 showing two interacting gears. The derivation is based on similar work presented for example in [46, 47, 49].

Figure 7.9 shows an interacting gear pair. The gear mesh stiffness is shown as a spring with stiffness k_g. This gear mesh stiffness can be determined using the ISO 6336-1:2006 guidelines, as described in [45]. The gears are interacting in the *plane of action*, which is defined by an angle φ with respect to the positive x-axis. This angle is a function of both the geometric angle γ between the gear centers and the *pressure angle* α of the gears. The pressure angle is a design parameter of the gears. Note that the angle of the plane of action is dependent on the driving direction of the gear pair. If the driving direction switches, the pressure angle of the gears changes sign. Hence, one can write the angle of

the plane of action as

$$\varphi = \gamma + \left(\frac{\pi}{2} - \alpha\right) \operatorname{sign}(\tau),$$

where τ is the driving direction of the gears (positive for clockwise rotation of the central gear, negative for counterclockwise rotation). The top view of the interacting gear pair in figure 7.9 shows the helix angle β. When this angle is zero, the gears are called *spur gears*. Note that when the driving direction changes, the helix angle β changes sign.

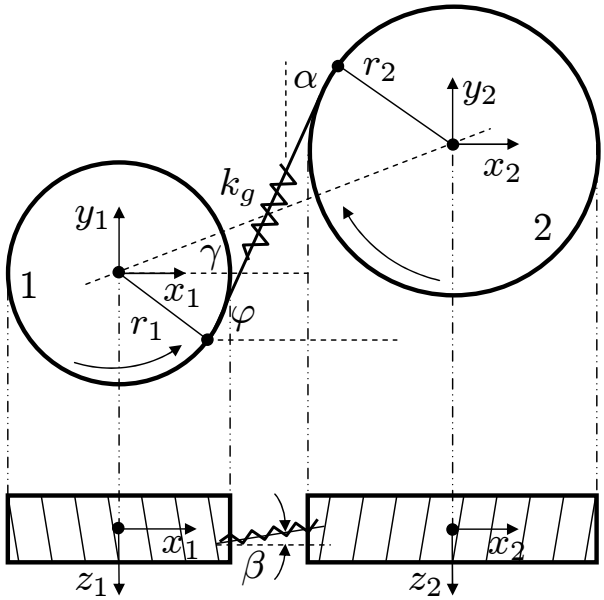


Figure 7.9: Schematic representation of gear interaction.

An energy approach will now be used to derive the gear stiffness element. This requires an expression of the deflection of the gear mesh as a function of the degrees of freedom of the centers of the interacting gears. These degrees of freedom can be assembled in the vector $\boldsymbol{q}$ as

$$\boldsymbol{q} = \left[\begin{array}{cccccccccccc} x_1 & y_1 & z_1 & \theta_{x1} & \theta_{y1} & \theta_{z1} & x_2 & y_2 & z_2 & \theta_{x2} & \theta_{y2} & \theta_{z2} \end{array}\right]^T.$$

Since the gear mesh can be loaded by compressive forces only, a compressive deflection is taken positive. Based on the figure above, one can then derive the deflection of the gear mesh as

$$\begin{aligned} \delta = \; & (x_1 \cos\varphi - x_2 \cos\varphi + y_1 \sin\varphi - y_2 \sin\varphi + r_1\theta_{z1} + r_2\theta_{z2}) \cos(\beta \operatorname{sgn}(\tau)) \\ & + (z_2 - z_1 + \theta_{x1} r_1 \cos\varphi + \theta_{x2} r_2 \cos\varphi + \theta_{y1} r_1 \sin\varphi + \theta_{y2} r_2 \sin\varphi) \sin(\beta \operatorname{sgn}(\tau)) \end{aligned}$$

Under the small displacements assumption, the spring deflection clearly is a linear function of the degrees of freedom of the gears. One can write the potential energy in the spring as:

$$V = \frac{1}{2} k_g \delta(\boldsymbol{q})^2$$

The 12×12 stiffness matrix can then simply be obtained by applying Lagrange equations [50], i.e.:

$$\boldsymbol{K} = \frac{\partial^2 V}{\partial \boldsymbol{q} \partial \boldsymbol{q}}$$

This gear stiffness element can now be defined between and assembled with any other structural element such as rigid, bar or beam elements. This way one can construct a model for the structural dynamics of the gearbox internals. Note that analogous to the derivation above one could derive the stiffness matrix representing the gear mesh stiffness of internal gears, e.i. the interaction between a planet and ring gear. It turns out that this stiffness matrix can be found simply from the derivation above by taking a negative radius for the internal gear [47].

7.5.2 Gearbox housing

The internals of the gearbox are mounted in the gearbox housing, which thereby functions as an interface between the running gears and the bedplate.

Since the geometry of the gearbox housing is not available, a simplified geometry of the gearbox housing is created based on the drawings from the supplier. This geometry is used in ANSYS to build the structural model. The gearbox housing is casted, therefore the mechanical properties of cast steel are used for the structural model, these are given in table 7.2. The housing is connected to the internal gears by the four ring wheels and

	Yaw gearbox housing
Density (ρ)	$7800[kg/m^3]$
Modulus of elasticity (E)	$200[GPa]$
Poisson's ratio (ν)	0.3

Table 7.2: Yaw ring and tower top material properties

three bearings. The ring gears are bolted into the housing and the bearings are pressed into the housing, therefore one can assume a rigid connection (since the housing itself is already very stiff). Using the rigidifying technique described in section 4.2 seven master nodes are created in the gearbox housing to facilitate the assembly of the internal gear model.

7.5.3 Yaw gearbox measurements

Several measurements have been performed on the yaw gearbox to obtain the missing parameters. A number of vibration measurements on the internal gears of the gearbox and one roving hammer measurement on the gearbox housing have been performed. These measurements did however, not give the desired results. Nonetheless we will briefly describe the different attempts to obtain useful data.

Figure 7.10: The yaw gearbox measurement setup

Internal yaw gear vibration measurements

A number of measurements were performed in order to try to measure the dynamic behavior of the gearbox internals. The setup of the measurements is shown in figure 7.10. The yaw gearbox is suspended using an elastic rope. Due to the flexibility in the rope and the mass of the gearbox, its rigid body eigenfrequencies are very low. From figure 7.10 it can be seen that there are two locations where one can excite the internal dynamics: the input pinion and the output pinion. Both will have their disadvantages, which will briefly be described here. Excitation through the input shaft has a number of disadvantages:

- Due to the large transmission ratio, small input signals are reduced to negligible amplitudes at the lower stages. One can imagine that these negligible amplitudes will be difficult (if not impossible) to measure.
- Excitation of the input shaft with an impulse hammer or a shaker would also be quite challenging (or practically even impossible), since even small forces (or impulses) will result in large rotations of the input shaft. Therefore the stinger can not be mounted on the input shaft.
- Due to the play between the gear teeth, gears will shake loose during the measurement, one thus measures a varying system instead of one constant system.
- Fixing or applying pretension through the output pinion is also not feasible, due to the magnification of a factor 1000, the torque at the output pinion will be very large. For the pretensioning an equal force will have to be applied at the output pinion which, due to this large force, is practically impossible in the current setup.

Excitation through the output pinion also has some disadvantages:

- The input displacement amplitude will be amplified. This results in a amplification of the inertia of the upper stages and will also amplify the friction of the upper stages.
- Due to the play between the gear teeth, gears will shake loose during the measurement, one thus measures a varying system instead of one constant system.

However, there are two major benefits that excitation through the output pinion has over excitation through the input pinion:

- It can practically be done. Since exciting the output pinion with (large) forces only leads to small rotations; a shaker stinger can be mounted on the output pinion.
- Since the input force is reduced instead of amplified, the input shaft can be pretensioned (or fixed) using only limited pretensioning forces in order to minimize the gear play.

In order to get some idea of the eigenfrequencies of the system, an impact hammer was used to excite the output pinion. No useful results where obtained from these measurements, which is most likely due to the high damping due to friction and gear play. The energy put into the system is dissipated in a very short time, thereby leading to useless FRFs.

The approach taken next was to replace the hammer by a shaker (as can be seen in figure 7.10), which is able to put energy in the structure during the entire measurement. A number of input signals are used to obtain the measurement data: random noise, a chirp and a full sine sweep signal. However, non of the input signals resulted in reproducible results, one of the FRFs found is shown in figure 7.11.

From these first shaker measurements a number of crucial variables were discovered:

- The gear play (i.e. if the gears where in contact) has a major influence on the measurements. In the extreme cases the input pinion had to be rotated up to five rounds in order for all the gears to be in contact. It was important to make sure that all the gears were in contact before starting a measurement. One could see from the results that the gear teeth lost contact and the system changed during the measurement.
- Pretension is essential in order to keep the gear teeth in contact with each other, to obtain a system which is as constant as possible.
- The force amplitude should be sufficient to overcome the internal gearbox friction.

A second measurement was attempted, taking into account the lessons learned from the first attempt. The internal friction was determined by measuring the moment needed to turn the input pinion of the yaw gearbox, the results are shown in table 7.3.

Using the results from the friction measurements, a pretensioning torque was applied to the input pinion which was able to overcome the friction of the yaw gearbox and ensure

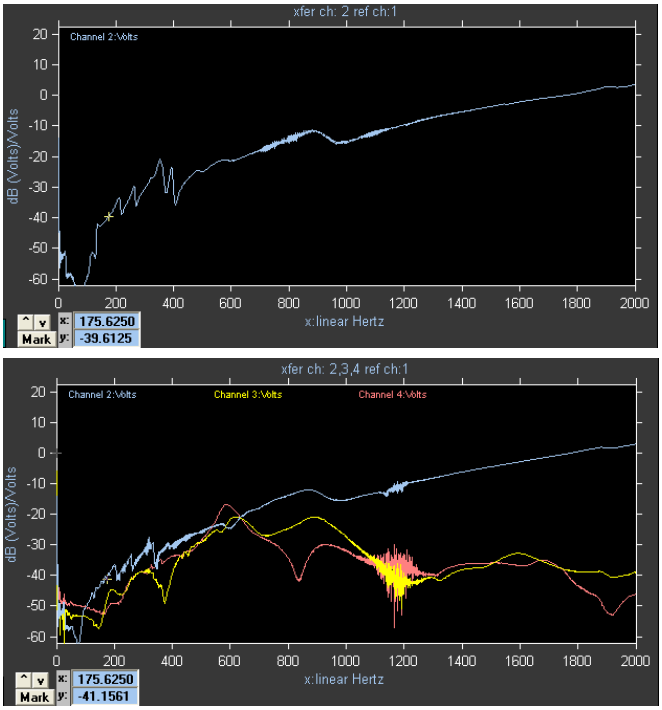


Figure 7.11: FRFs from the first measurements on the yaw gearbox

	Friction of all stages [Nm]	Friction of upper stage [Nm]
1	0.144	0.090
2	0.149	0.096
3	0.157	0.102
4	0.151	0.101
5	0.139	0.101
Average	0.148	0.098

Table 7.3: Results of static friction measurement

gear teeth contact during the measurements. The pretension was applied by using the a mass suspended on a cable, which generated a torque through the use of a lever (the radius of the input pinion), as can be seen in figure 7.12. Using this setup the vibration measurements were performed for a second time. Again no useful results were obtained, which could also result from the pretension applied to the system. Since the pretension was achieved through a mass that generated a constant torque, inertia was also added. Due to the gearbox ratio, this inertia is amplified by a factor of more than a million (since this scales quadratically), which would obviously significantly influence the measurement results.

From these attempts we can conclude that excitation through a shaker and/or hammer did not give the desired results. One idea to come to usable measurement data is to mount an electro motor on the input shaft and to use a constant sinus (constant rotation), which will ensure teeth contact of the gears, and superimpose a random excitation to create an

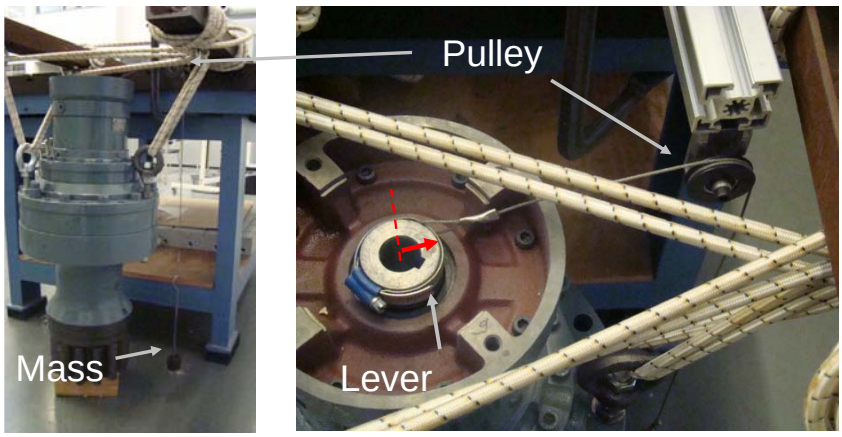


Figure 7.12: Pre-tensioning of the yaw gearbox

input signal. One would then have to resort to operational modal analysis techniques to obtain the eigenmodes and eigenfrequencies [51].

CHAPTER 8

Results of the dynamic substructuring analysis

8.1 Introduction

In order to demonstrate the potential of the dynamic substructuring approach in wind turbine engineering, the yaw system is analyzed in this chapter. A description of the yaw system and its components and interfaces was given in chapter 6. The component models, described in chapter 7, are reduced using the different component reduction techniques (as described in chapter 3) and assembled into a reduced model of the yaw system using the techniques described in chapters 2 and 4. In addition a non-reduced model is build from the full component FE models, which will be used to compute the reference solution. By comparing the results from the reduced models with the reference solution, statements about the accuracy of the different reduction methods can be made and it allows us to verify the dynamic substructuring methodology.

Firstly, in section 8.2, a bare badplate model, that is, without the main bearing housing, yaw clamps and gearbox mounts, is assembled with four yaw gearboxes models and a modal analysis is performed to obtain the eigenfrequencies and mode shapes. This assembly is also build in reality in order to allow for an experimental modal analysis to extract the eigenfrequencies and eigenmodes, which are then used to validate the assembly of component models. In section 8.3 a modal analysis is performed on the reduced models of the complete yaw system and compared with the eigenfrequencies and mode shapes of the reference system. Finally, in section 8.4 a SUMAC analysis is performed and its results will be discussed.

8.2 Validation of the assembly

The setup for the second measurement, which was performed after the bedplate validation measurements described in section 7.2.2, was build by mounting four yaw gearboxes to the bedplate. Subsequently, an experimental modal analysis was performed on this assembly, which can be seen in figure 8.1.

The bedplate was suspended using four air springs. These air springs were pressurized at approximately 6 bar, which was slightly higher in comparison to the first measurement to account for the extra gearbox weight, and created a low stiffness suspension. Again, it was found that the rigid body eigenfrequencies were all around 3 Hz and well below the first flexible eigenfrequency. Using a total of nine tri-axial ICP accelerometers, 33

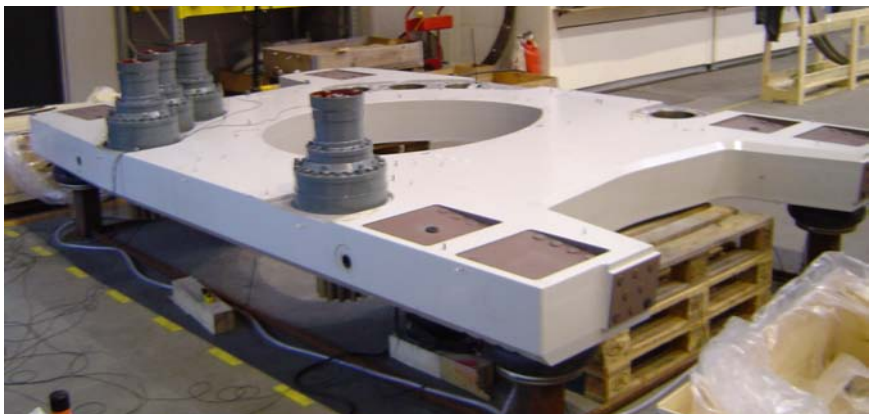
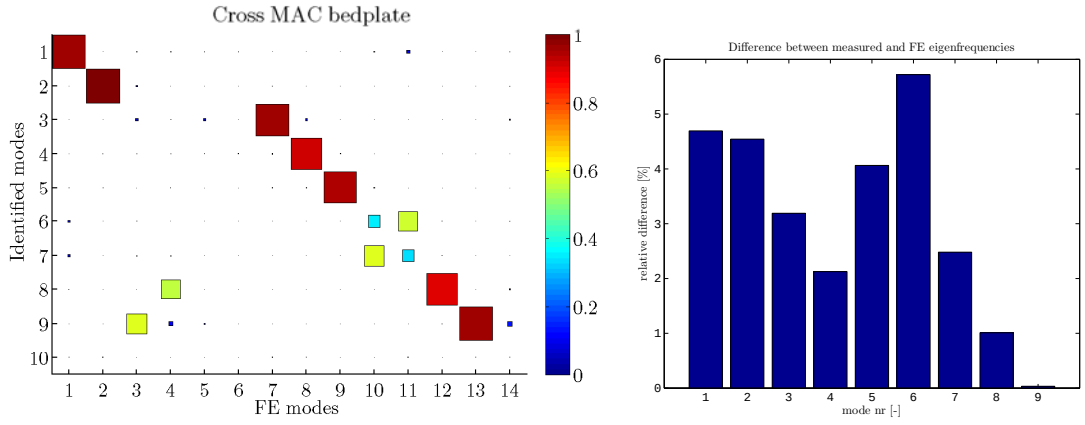


Figure 8.1: Photo of the measurement setup

locations were measured on the bedplate and eight locations were measured on the yaw gearboxes. Excitation of the bedplate was applied by a shaker using a random noise signal. Details of the measurement equipment can be found in appendix C.1. Again, the SD Toolbox in MatLab was used to identify the eigenmodes and eigenfrequencies. The measured modes were expanded using the SEREP technique [44] and a MAC analysis was performed to visualize correlation between the measured modes and the finite element modes, as can be seen in figure 8.2. Note that in figure 8.2 the rigid body modes of



(a) (b)

Figure 8.2: Measurement results of the assembly of a bare bedplate and four yaw gearboxes

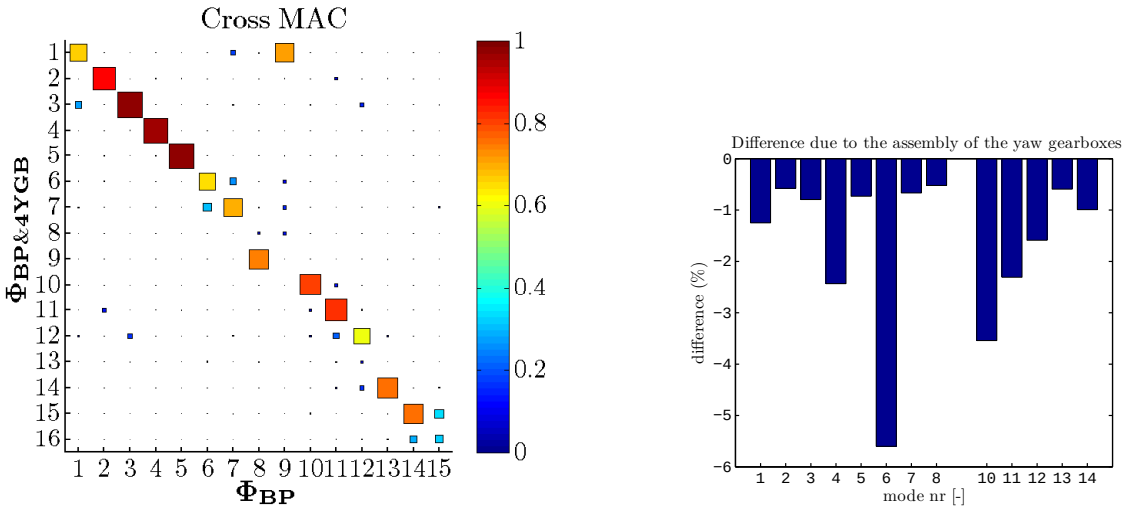
the FE model are not shown. From these results it is clear that the first and second (flexible) eigenmodes computed from the assembled finite element model have a high correlation with the first two measured modes. It can also be seen that there are four modes missing in the measurement (or four "extra" modes in the FE results), these are modes of the gearbox internals. Since no sensors are placed at the gearbox input and output pinion, these could not be measured. In addition, since the yaw gearbox model is not yet validated no conclusions can be drawn from these modes. The next set of

modes show a good correlation to the measurement, except for FE modes 10 and 11. The frequency difference, which is for most modes below 5 %, is somewhat high, but nonetheless not bad, given the fact that the gearbox models still need to be updated with a measured set of parameters.

From these results we can conclude that the added mass and stiffness effects of assembling the yaw gearboxes to the bedplate are also present in the assembled FE model, but still have a slight discrepancy with respect to the measured system. Therefore, the results should be compared at a later stage when the yaw gearbox model has been updated and/or fully validated.

8.2.1 Differences after assembling the yaw gearboxes

The effects of assembling the yaw gearboxes on the eigenfrequencies and modeshapes of bedplate can be found from the measurement data and are visualized in figure 8.3. From the results it is clear that both the modeshapes and eigenfrequencies change due



(a) (b)

Figure 8.3: Effect on results due to the assembly of the yaw gearboxes

to the added mass and stiffness of the yaw gearboxes. Still, the modeshapes of the bare bedplate and the bedplate with the attached yaw gearboxes show a (low) correlation. It can be seen that the correlation of modes 3, 4 and 5 is still very high and it can thus be concluded that these modes are insensitive for the added gearbox stiffness and mass, whereas the other modes clearly change due to the added mass ad stiffness.

8.3 Dynamic substructing analysis of the yaw system

In this section a dynamic substructuring analysis will be performed on the yaw system, containing all the substructure models. Similar to the comparisons made in section 5.4, the modes and eigenfrequencies of the differently reduced yaw system models are compared to a full reference solution. In order to reduce the number of DoF even further, interface reduction is applied and its results are compared to the reference solution.

The two criteria used for the comparison are:

- Frequency error: the frequency error [%] of the eigenfrequencies with respect to the reference solution is given. In order to be accurate, the error of the eigenfrequencies should be no more than 1 [%].
- Error of the mode shapes: the MAC matrix is computed to quantify the error of the expanded mode shapes with respect to the reference solution. The MAC values of the diagonal are subtracted from 1 and plotted for each mode. The error of the mode shapes should be less than 0.05 [-] in order to considered accurate.

The assemblies of reduced components that are compared are given in table 8.1.

	YR-TT	YP	BP	YGB	# of DoF
reference	full	full	full	full	293712
CB30	CB-30	CB-15	CB-30	CB-30	7929
R30	R-30	R-15	R-30	R-30	7881
DCB30	DCB-30	DCB-15	DCB-30	DCB-30	8637

Table 8.1: Overview of the assembled components

The abbreviations for the component models are as follows: YR-TT for the yaw ring and towertop substructure, YP for the yaw pad model array, BP for the bedplate model and YGB for the yaw gearbox model array. The abbreviation in the first column gives the name of the assembly and corresponds to the names given in the figures. Note that for the yaw pad array and the gearbox array, the number of modes per pad (15) and per gearbox (30) are given in 8.1. The results of the reduced yaw system models are given in figures 8.4 to 8.8. Note that in figure 8.4 and 8.5, the results from the Dual Craig-Bampton assembly are corrected for the so called "spurious modes" (as discussed in sections 3.6 and 5.4). These "spurious modes" can be seen in the MAC plot given in figure 8.8, where at mode number 14 and mode number 51 an extra mode is introduced, which is an artifact of the reduction procedure.

From the obtained results, a number of observations can be made.

- All the reduction methods show excellent results and are accurate up to at least the 80^{th} eigenmode and eigenfrequency. Still, the total number of DoF are reduced by a factor of approximately 35.
- The methods have similar accuracy, so no actual "winner" can be found from these results. The assemblies of Craig-Bampton and Dual Craig-Bampton show

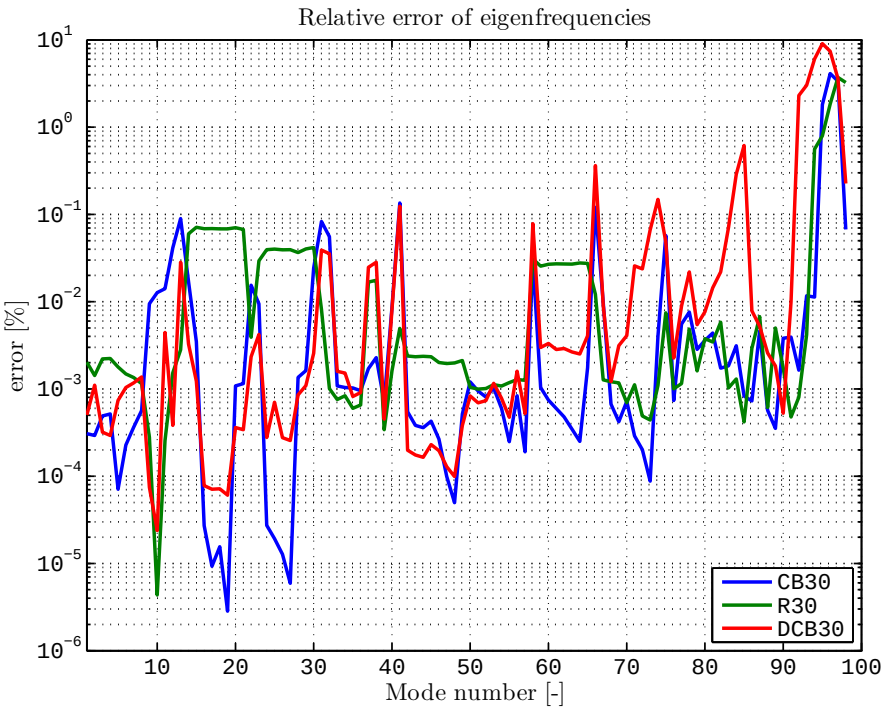


Figure 8.4: Frequency error of the reduced systems with respect to the full system

better results in the lower range of eigenmodes in comparison to the assembly of Rubin reduced substructures. In the higher frequency range it can be seen that the Craig-Bampton and Rubin methods give slightly better results than the Dual Craig-Bampton method.

- In figure 8.8 the MAC plot between the modes of the reference system and those of the Dual Craig-Bampton system is given. From this figure the spurious modes introduced by the reduction can clearly be seen, as they give sudden jumps in the MAC plot. These spurious modes originate from the fact that the Dual Craig-Bampton only enforces a weak form of compatibility, it therefore allows for motion of the interface DoF which is physically not possible (e.g. relative sliding of the interfaces).

The next step is to apply interface reduction to these assemblies of reduced components, as described in section 5.2. The results of these "double" reduced assemblies are given in figures 8.9 to 8.20. Firstly, the different analyzed configurations can be found in table 8.2.

Again, a number of observations can be made from the results.

- All the interface reduction methods show excellent results and are accurate up to at least the 80^{th} eigenmode and eigenfrequency. Nevertheless, the interface reduced systems are less accurate then the original CB30, R30 and DCB30 systems. The big advantage however, is that by applying the interface reduction, the total number of DoF is reduced up to a factor of approximately 400. This allows for much shorter

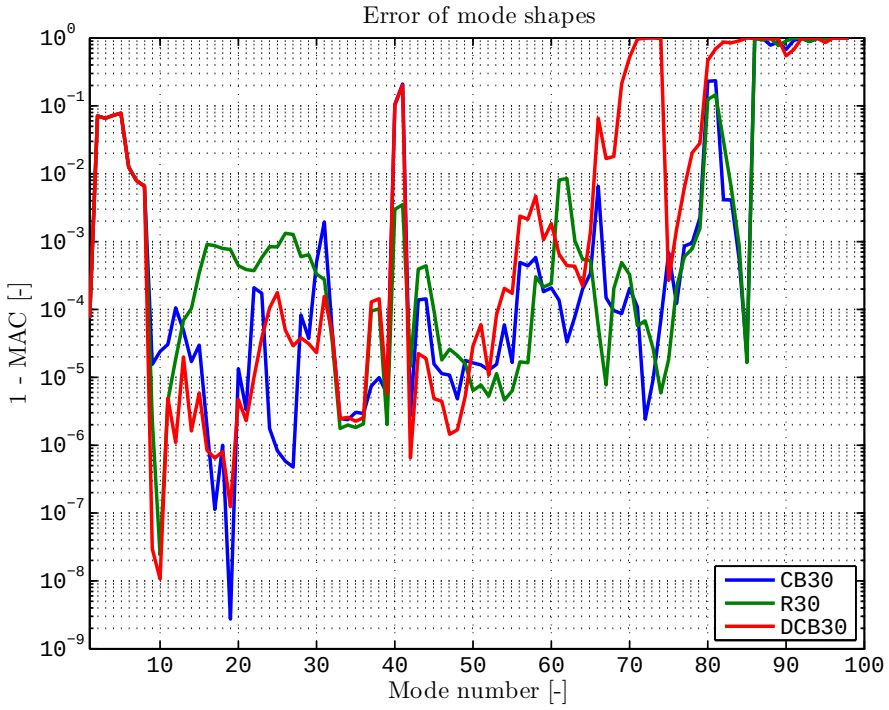


Figure 8.5: 1-MAC value between the reduced systems' eigenmodes and the full system eigenmodes

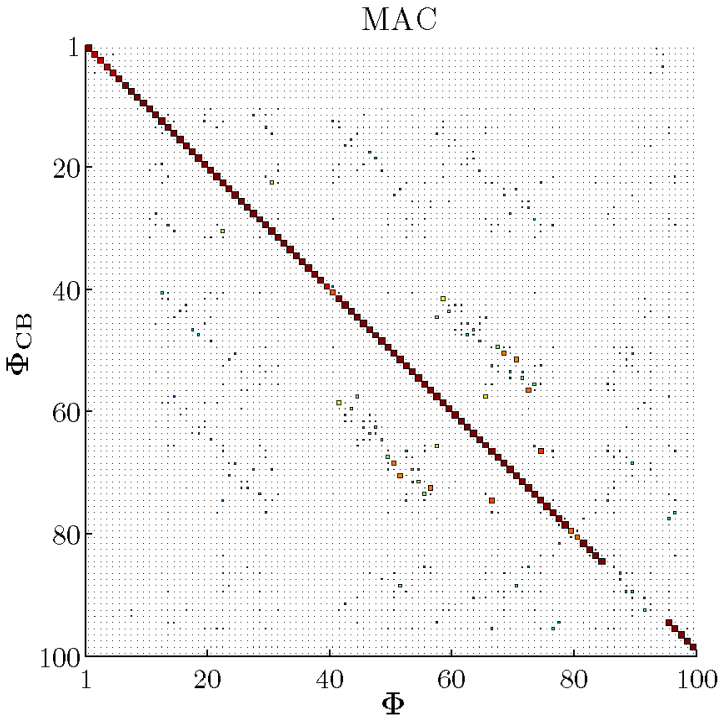


Figure 8.6: Cross MAC between the reduced (and expanded) modes of the CB system and the reference modes

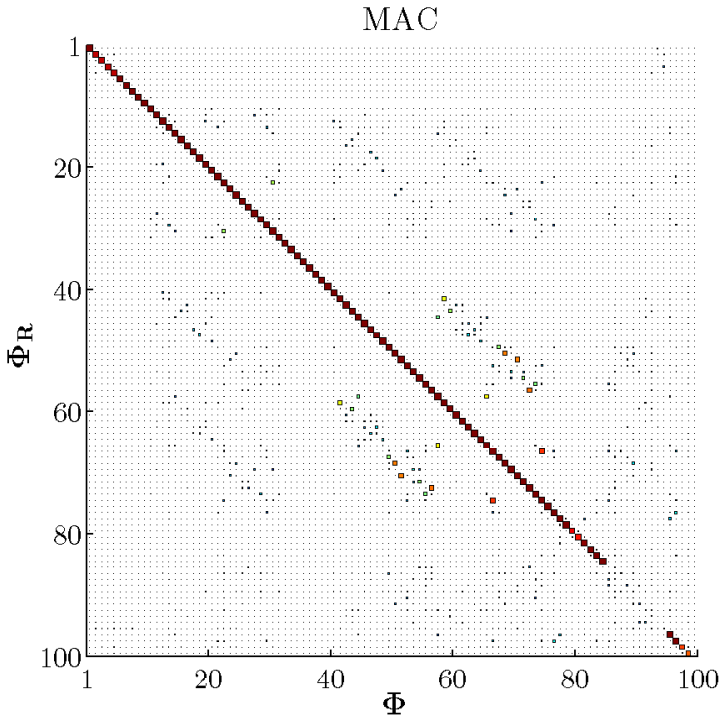


Figure 8.7: Cross MAC between the reduced (and expanded) modes of the R system and the reference modes

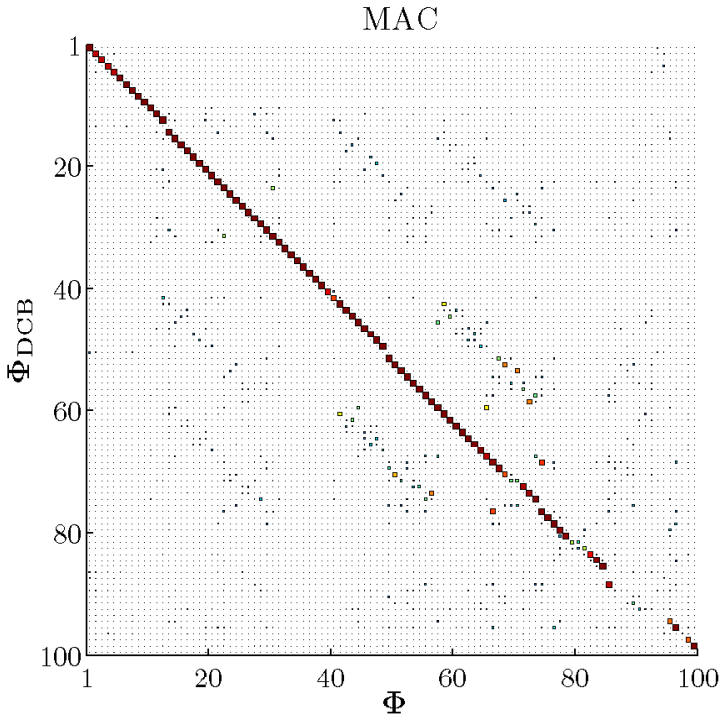


Figure 8.8: Cross MAC between the reduced (and expanded) modes of the DCB system and the reference modes

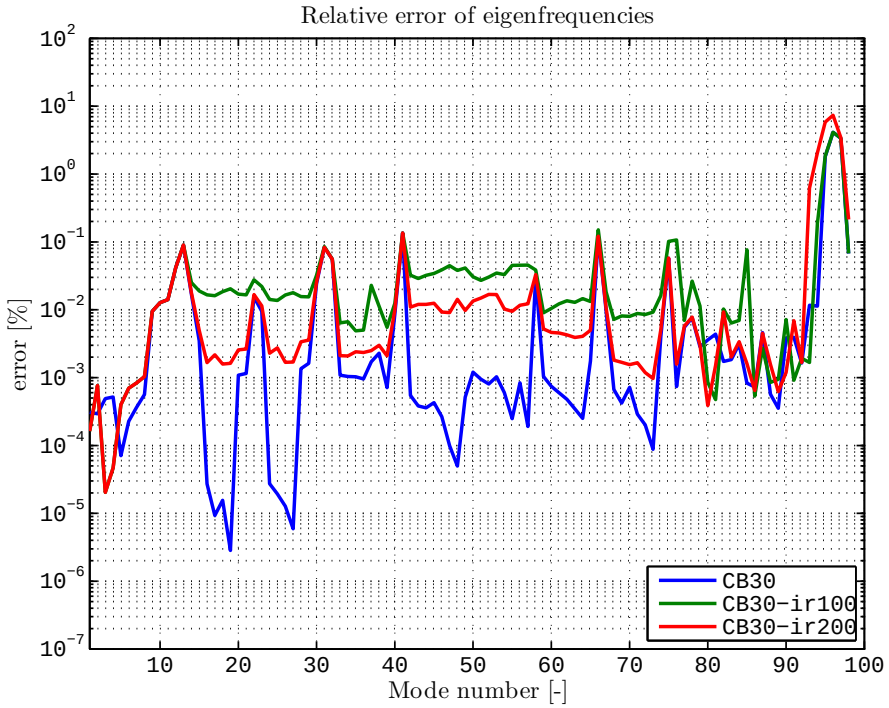


Figure 8.9: Frequency error between the reference and CB and CBir reduced models

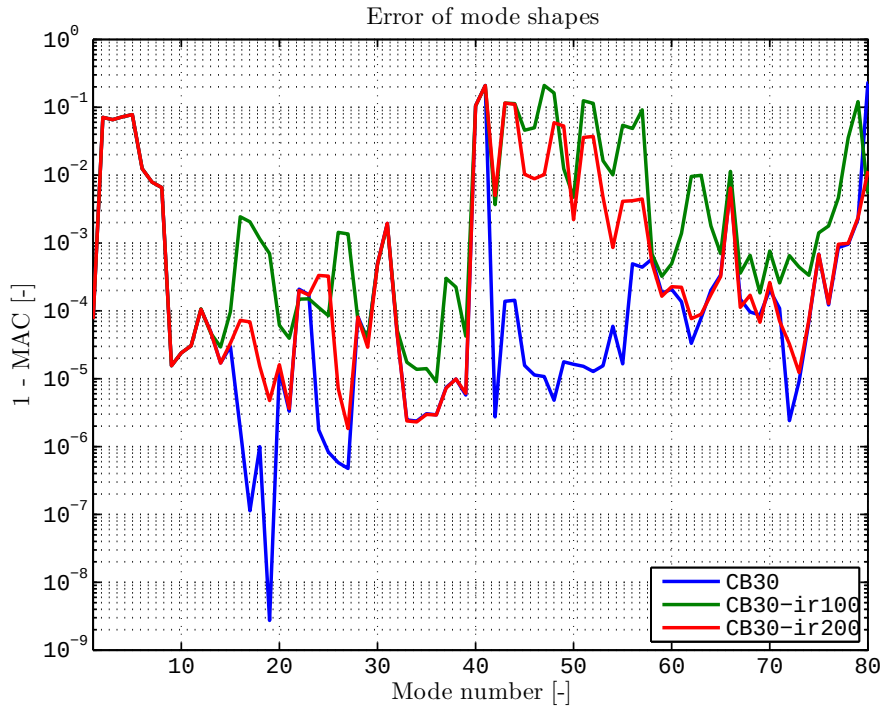


Figure 8.10: Error on the modeshapes of the CB and CBir reduced models

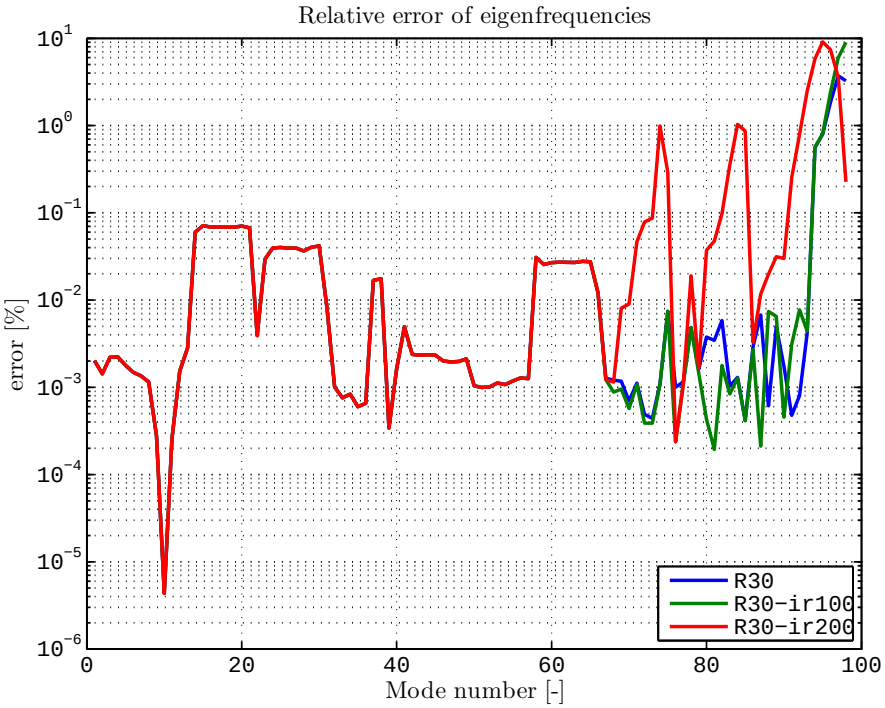


Figure 8.11: Frequency error between the reference and R and Rir reduced models

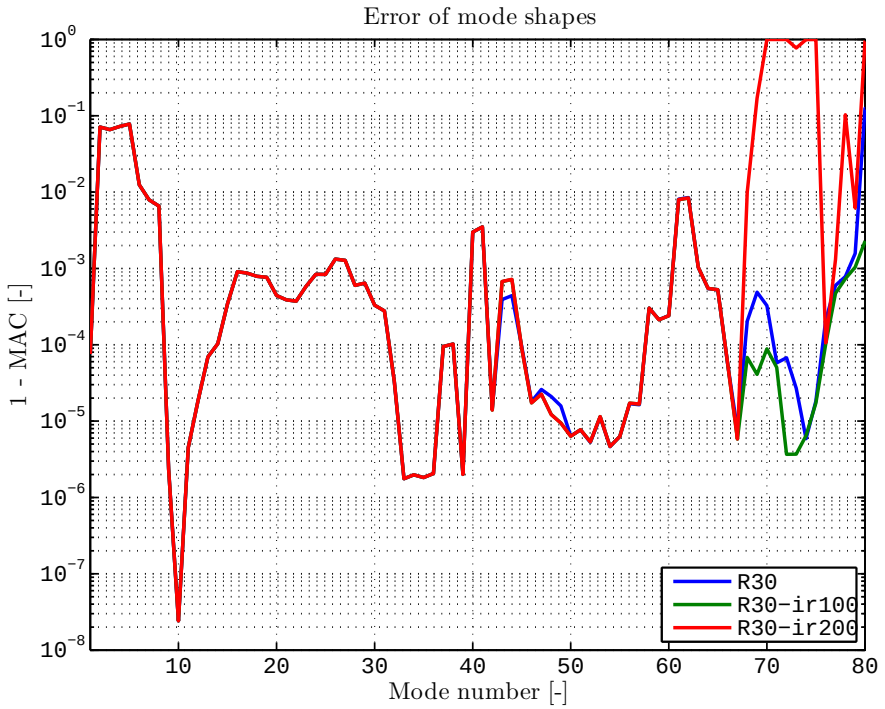


Figure 8.12: Error on the modeshapes of the R and Rir reduced models

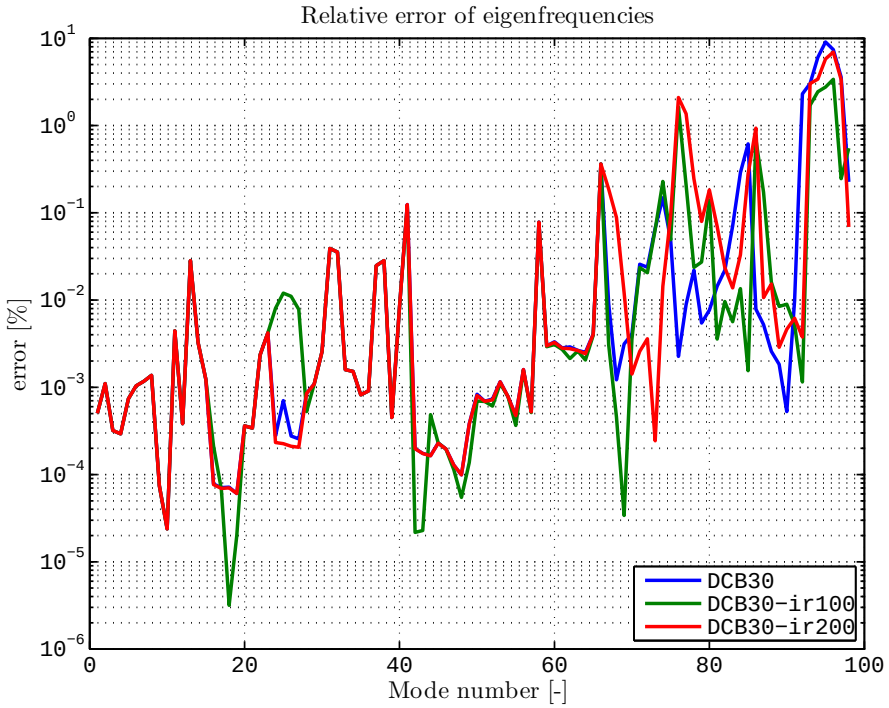


Figure 8.13: Frequency error between the reference and DCB and DCBir reduced models

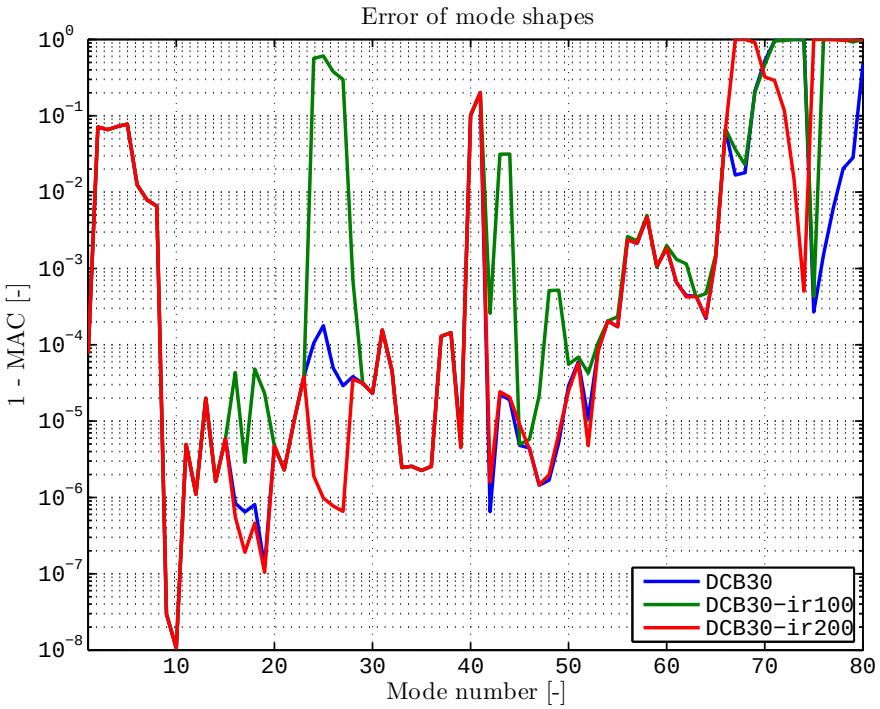


Figure 8.14: Error on the modeshapes of the DCB and DCBir reduced models

	YR-TT	YP	BP	YGB	# of interface modes	# of DoF
reference	full	full	full	full	–	293712
CB30-ir100	CB-30	CB-15	CB-30	CB-30	100	730
CB30-ir200	CB-30	CB-15	CB-30	CB-30	200	830
R30-ir100	R-30	R-15	R-30	R-30	100	730
R30-ir200	R-30	R-15	R-30	R-30	200	830
DCB30-ir100	DCB-30	DCB-15	DCB-30	DCB-30	100	730
DCB30-ir200	DCB-30	DCB-15	DCB-30	DCB-30	200	830

Table 8.2: Overview of the assembled components

computation times and one does not require a supercomputer to perform load case simulations or other types of analyses.

- As was already discussed in section 5.4, the number of interface modes in the interface reduction is a crucial parameter in obtaining accurate, but compact models.
- Due to the presence of rigid body modes in their reduction bases, the Rubin and Dual Craig-Bampton interface reductions are performed by a so called "dynamic interface reduction" (see section 5.2). The Craig-Bampton interface reduction bases on the other hand, are computed using a "static interface reduction". It is therefore not fair to compare the results to determine which of the methods is superior here, hence no conclusions can be drawn from these results.
- In figures 8.11 and 8.12 it can be seen that the Rubin system reduced with 100 interface modes performs better than the one reduced with 200 interface modes. This seems contradictory, since an increase in the number of modes should always lead to a more accurate approximation. It is therefore believed that this is an issue which probably results from the solver used to determine the eigenmodes. Since higher modes will generally be less accurate (and are thus more "contaminated"), this can result in errors in the reduced system. In addition, as can be seen from figure 8.18, the high MAC values move off-diagonal at the higher modes. The figures 8.11 and 8.12 are computed from the diagonal entries, thereby resulting in less favorable results for the Rubin system reduced with 200 interface modes (R30-ir200).

8.4 SUMAC analysis

Here, a SUMAC is performed using the modes of the total yaw system assembly ($\boldsymbol{\Phi}_{tot,assem}$) and modes of the assembly of the yaw ring and towertop, yaw pads and bedplate ($\boldsymbol{\Phi}_{YRTT\&YP\&BP}$), as can be seen in figure 8.21. The theory of the SUMAC analysis is described in section 5.3. The SUMAC shows the correlation between the global mode shapes and the mode shapes of the yaw ring and towertop, yaw pads and bedplate assembly. From the SUMAC figure, one can thus visualize the effects on the mode shapes, caused by mounting the yaw gearboxes onto the bedplate. First of all, due to the shifts

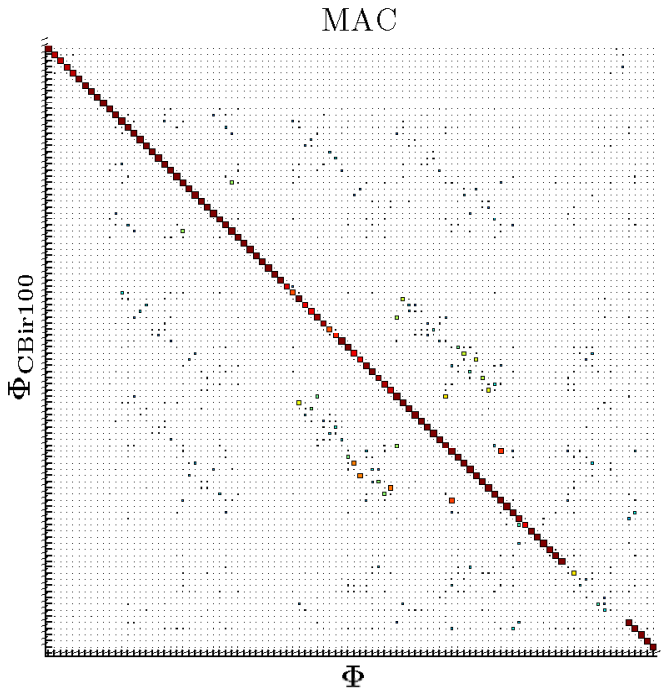


Figure 8.15: Cross MAC between the reduced (and expanded) modes and the reference modes

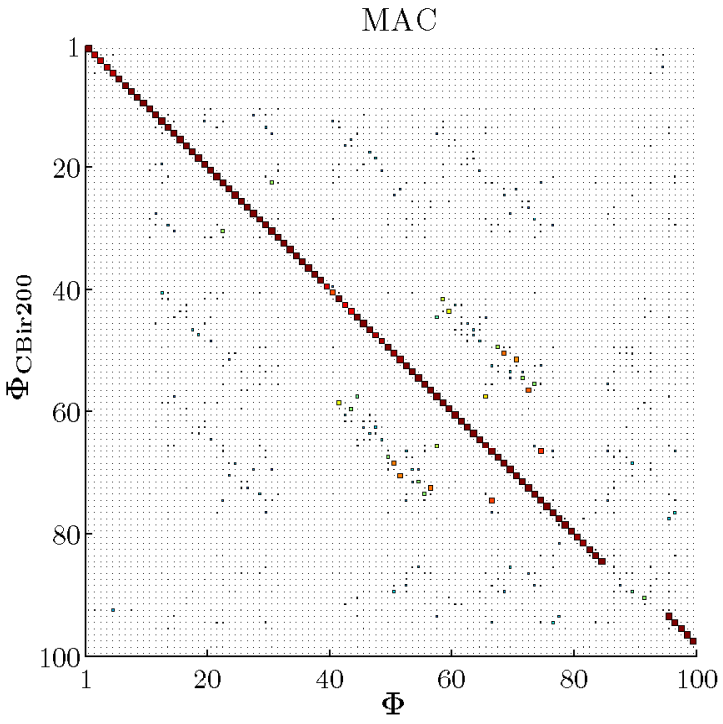


Figure 8.16: Cross MAC between the reduced (and expanded) modes and the reference modes

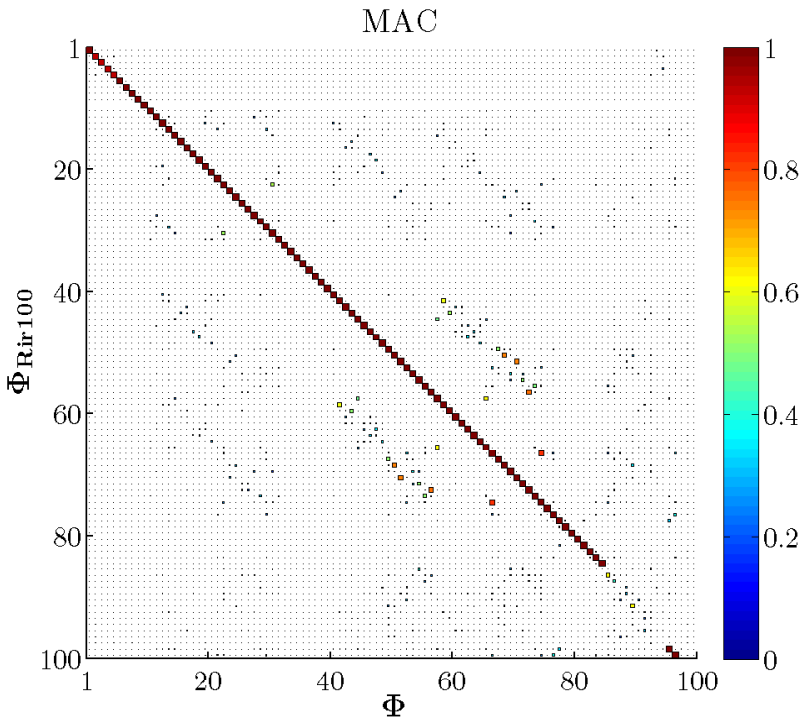


Figure 8.17: Cross MAC between the reduced (and expanded) modes and the reference modes

of the red blocks, which denote a high correlation, one can identify the gearbox modes in the set of global modes. But one can also see from, for instance, modes 14, 22 and 30 that there is an interaction between the gearboxes and the rest of the assembled system. Finally, it is clear that the added mass and stiffness affect some modes more than others, thereby leading to higher frequencies for some modes and lower frequencies for other modes. This can be seen at modes 37 to 41 of $\boldsymbol{\Phi}_{tot,assem}$, which have a significant correlation with their associated mode (13 to 17 of $\boldsymbol{\Phi}_{YRTT\&YP\&BP}$), but where the order of the modes is altered.

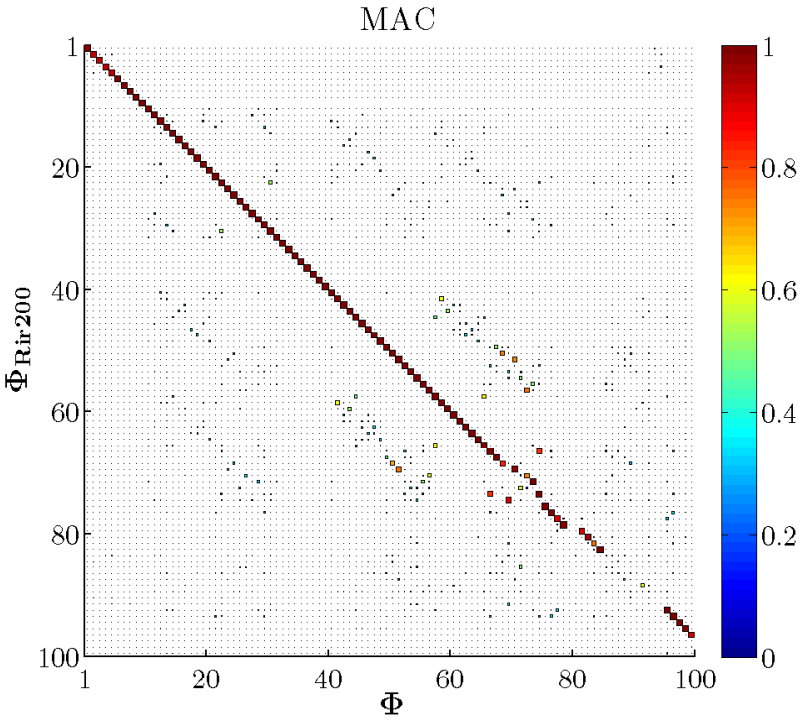


Figure 8.18: Cross MAC between the reduced (and expanded) modes and the reference modes

Figure 8.19: Cross MAC between the reduced (and expanded) modes and the reference modes

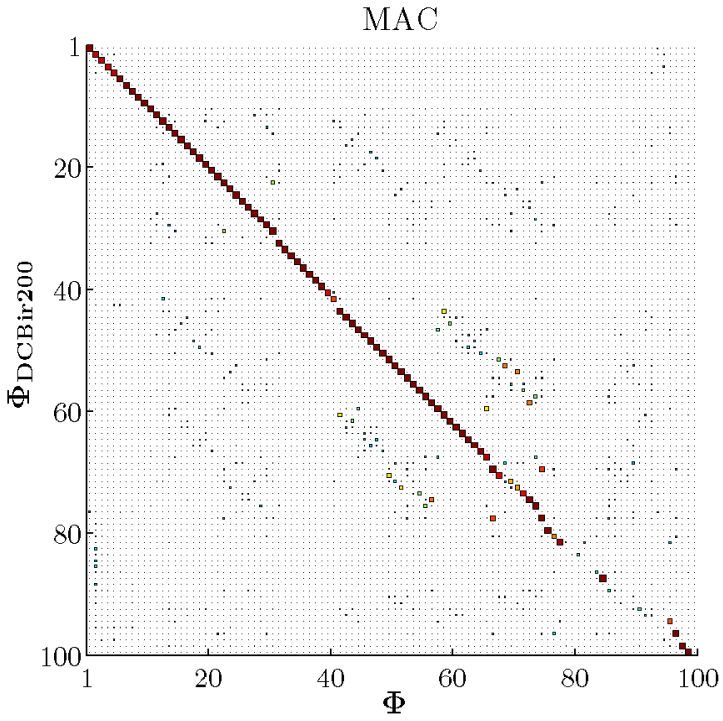


Figure 8.20: Cross MAC between the reduced (and expanded) modes and the reference modes

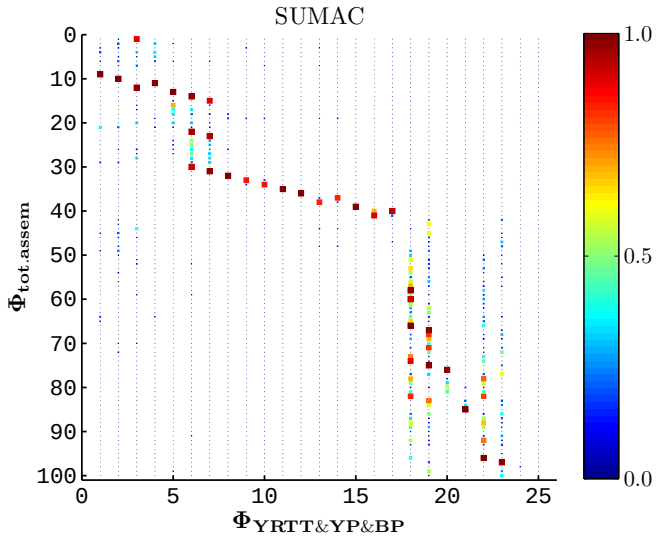


Figure 8.21: SUMAC of the modes from the total assembly and the modes of the YR-TT, YP and BP assembly

Part III

Conclusions and Recommendations

CHAPTER 9

Conclusions and recommendations

9.1 Conclusions

In this thesis, the paradigm of dynamic substructuring (DS) was proposed to fill the need for a more detailed structural dynamic analysis tool within wind turbine engineering. It was felt that there is a need for more accurate and versatile dynamic modeling techniques in addition to the commonly used aero-elastic dynamic simulation codes. The goal of this work was therefore to set up and implement a general framework for the application of dynamic substructuring in wind turbine engineering. This framework has been created and was successfully applied to model the yaw system of a 2.3 MW Siemens wind turbine. From this result, it can be concluded that the goal of the MSc. project has been achieved.

In chapter 2 a general framework for substructure assembly was presented. This framework is more extensive and complete than those found in literature, since it does not only include the assembly of interface displacements. It also allows assembly of interface forces and mixed assembly, which enables coupling of interface forces to interface displacements. Using this framework for assembly, one is able to assemble all types of full and reduced FE substructure models. This framework is therefore an important step in further generalizing the dynamic substructuring approach.

The concept of mixed assembly enabled the generalization of the Craig-Bampton and Dual Craig-Bampton methods to the Mixed Craig-Bampton method, which was introduced in chapter 3. It was shown that the accuracy of the method is similar to the Craig-Bampton and Dual Craig-Bampton methods. The benefit of the method is that a simple selection criterion per DoF is used to determine whether free of fixed interface modes should be used in the reduction basis. Compared to other mixed boundary CMS methods [30–32] this method is unique in the fact that both the reduction and assembly are performed in a mixed sense.

The Craig-Bampton, Rubin and Dual Craig-Bampton methods have been used in the dynamic substructuring analysis performed on the yaw system of a 2.3 MW Siemens wind turbine. It was shown that all methods were able to accurately describe the dynamic behavior of the yaw system with a limited number of generalized DoF. Still, due to the relatively large interface areas, a high number of interface DoF was present in the set of reduced DoF, which limited the effectiveness of the reduction. It was concluded that an additional reduction step was needed to improve the effectiveness of the DS approach, which led to the implementation of interface reduction.

An approach for interface reduction using interface forces, similar to interface reduction

using interface displacements, is introduced in chapter 5. Both methods enable to reduce the models even further while maintaining an accurate description of the dynamic behavior of the global model. One has to be aware of the fact that a poor interface reduction basis can significantly affect the results and thus a cautious approach should be taken here. The latter is especially important when reducing the interface forces of the Dual Craig-Bampton substructures, since it will further weaken the compatibility between the substructures, which was shown to lead to spurious modes. Nonetheless the method is successfully applied in chapter 8. From this we can conclude that the interface reduction is an effective technique for further reducing the number of DoF, while maintaining an accurate description of the global dynamic behavior.

From the results obtained in chapter 8, one can conclude that the dynamic substructuring approach shows great potential for use in wind turbine engineering. Using the reduction methods treated in the thesis, we were able to reduce the number of DoF of the yaw system from almost 300.000 to a little less than 750, while maintaining an accurate description of the dynamic behavior up to the 80^{th} mode. Note that, since all reduced substructure models originate from the three dimensional CAD models, all the geometrical information and details are accounted for in the reduced models. Even though some models are significantly simplified and not all the models used in the analysis are validated, it is clear that the techniques presented in this thesis allow for creating compact and accurate descriptions of the dynamic behavior of wind turbine components and assemblies.

9.2 Recommendations

In this MSc. thesis the basis for the application of dynamic substructuring in wind turbine engineering has been presented. However, a number of challenges remain. From these challenges, which were encountered during the MSc. project, a number of recommendations for future work can be extracted. These recommendations can be split into two categories, one for the theoretical part of the work and one for the application of dynamic substructuring to the wind turbine.

9.2.1 Recommendation on the theoretical challenges

Firstly, a number of recommendations on the theoretical side of the work can be made, which are discussed here.

- Apply interface reduction per substructure. In the ideal case, one wants to create all (reduced) substructure models in advance, since this allows for a true "LEGO" approach. This would require that the interface reduction is also performed *a priori*, but this would introduce the problem of non-conforming interface discretization, similar to assembling substructures with non-conforming meshes. However, one can imagine that these reduced interfaces can be far more incompatible than those

encountered when assembling non-conforming meshes. As a result the discrete compatibility as obtained using the methods described in section 5.2, can result in significant errors and one will have to resort to more sophisticated methods to assemble these substructure models which have *a priori* reduced interfaces (i.e. Mortar methods [37]).

- Generalize the interface reduction methods of section 5.2, to a mixed interface reduction. Interface reduction of interface displacements and interface forces is already presented. The open issue is therefore reduction of mixed interfaces, where some interface DoF are in terms of displacements and some in terms of forces.

- Extend the work presented here by including controller and non-linear component models. Since not every component can be modeled in a linear fashion, the framework will have to be extended to include non-linear component models. Another issue arising here is the reduction of non-linear component models, since the techniques presented in chapter 3 are not applicable to non-linear component models. One option would be to use Proper Orthogonal Decomposition (POD) techniques to build non-linear superelements [52,53]. In addition, if an integrator is present in the controller model, one needs to switch to a description into state space in order to allow assembly, which will lead to a doubling of the number of DoF.

- Implement time integration schemes in order to enable simulation of load cases. In order to use the assembled substructure models to simulate certain load cases, a time integration scheme will have to be implemented in the PP Tool. One can identify a number of difficulties here:

  - Non-linear component and/or interface models. If the assembled model contains non-linearities iterative solvers will have to be used to find the solution. Preferably, one would like to split the total model in a linear part and a non-linear part such that different algorithms can be used for the linear and the non-linear part in order to reduce the time needed for computations.

  - Another difficulty is to find a proper time step to take into account all the desired effects. For example when trying to model stick slip, very high frequent oscillations can occur, requiring very small time steps.

  - Controller models. Due to the presence of controller models, one might have to switch to state space time integrators for the entire system. This will require one to rewrite the equations of motion in a state space form and thereby doubling the number of DoF of the total system.

- Similar to most types of machinery, a wind turbine contains a lot of rotating parts, such as the rotor, nacelle, gearboxes, etc. In order to be able to compute the response to actual load cases, the model should be able to cope with these large component rotations. One can achieve this by implementing a corotational approach as described in [54]. In the corotational approach, a substructure is described in its own local coordinate system, since these local axis can move and rotate with respect to the global coordinate system, large component rotations are

enabled.

9.2.2 Recommendations for future modeling and applications

In addition to the improvements on the theoretical side of the work, also a number of recommendations for improving the modeling of components and the application to wind turbine engineering can be made.

- Include the yaw controller in the yaw system assembly and model the complete set of (non-linear) yaw pads, in order to obtain a better representation of the dynamic behavior of the yaw system. Due to time limitations this could not be performed within this work.
- Perform measurements on all components in order to validate their substructure models. And subsequently, use these validated substructure models to try to validate and/or update the interface models in order to obtain a validated assembly of the yaw system.
- Add the remaining wind turbine components to build a full superelement model of the Siemens 2.3 MW wind turbine. This total model can then be used to investigate the effect of the yaw system dynamics on the global wind turbine dynamics.
- One could expand the modeling strategy to wind turbine foundation modeling. For example, one could try to create reduced non-linear soil and fluid models acting on the foundation, in order to obtain accurate and compact models, which can be efficiently used for load simulation purposes.

Bibliography

[1] S.N. Voormeeren, P.L.C. van der Valk, and D.J. Rixen. Practical aspects of dynamic substructuring in wind turbine engineering. In *Proceedings of Twenty Eighth International Modal Analysis Conference, Jacksonville, FL*, 2010.

[2] P. Passon, M. Kühn, S. Butterfield, J. Jonkman, T. Camp, and T.J. Larsen. OC3-Benchmark Exercise of Aero-elastic Offshore Wind Turbine Codes. *Journal of Physics: Conference Series*, 75:1–12, 2007.

[3] Etienne Balmés, Jean-Philippe Bianchi, and Jean-Michel Leclére. *Structural Dynamics Toolbox and FEMlink*.

[4] H. A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2, pages 133–143. 1890. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, volume 15, 1870, 272-286.

[5] A. Hrennikoff. Solution of problems of elasticity by the frame-work method. *ASME Journal of Applied Mechanics*, 8:619–715, 1941.

[6] R. Courant. Variational Methods for the Solutions of Problems of Equilibrium and Vibrations. *Bulletin of the American mathematical Society*, 49, 1943.

[7] Walter C. Hurty. Vibrations of Structural Systems by Component Mode Synthesis. *Journal of Engineering Mechanics, Division American Society of Civil Engineers*, 86(4):51–69, 1960.

[8] W.C. Hurty. Dynamic Analysis of Structural Systems using Component Modes. *AIAA Journal*, 3(4):678–685, 1965.

[9] G.M.L. Gladwell. Branch Mode Analysis of Vibrating Systems. *Journal of Sound and Vibration*, 1:41–59, 1964.

[10] R.R. Craig and M.C.C. Bampton. Coupling of Substructures for Dynamic Analysis. *AIAA Journal*, 6(7):1313–1319, 1968.

[11] R.H. MacNeal. A Hybrid Method of Component Mode Synthesis. *Computers & Structures*, 1(4):581–601, 1971.

[12] S. Rubin. Improved Component-Mode Representation for Structural Dynamic Analysis. *AIAA Journal*, 13:995–1006, 1975.

[13] J.R. Crowley, A.L. Klosterman, G.T. Rocklin, and H. Vold. Direct Structural Modification using Frequency Response Functions. In *Proceedings of the Second Interna-*

tional Modal Analysis Conference, Orlando, FL, pages 58–65, Bethel, CT, February 1984. Society for Experimental Mechanics.

[14] B. Jetmundsen, R.L. Bielawa, and W.G. Flannelly. Generalized frequency domain substructure synthesis. *Journal of the American Helicopter Society*, 33:55–65, January 1988.

[15] D. de Klerk, D.J. Rixen, and J. de Jong. The Frequency Based Substructuring (FBS) Method Reformulated According to the Dual Domain Decomposition Method. In *Proceedings of the Twenty Fourth International Modal Analysis Conference, St. Louis, MO*, Bethel, CT, February 2006. Society for Experimental Mechanics.

[16] J.S. Leiva, P.J. Blanco, and G.C. Buscaglia. Iteartive strong coupling of dimensionally-hetrogeneous models. *International Journal for Numerical Methods in Engineering*, To appear.

[17] K. Cuppens, P. Sas, and L. Hermans. Evaluation of the frf based substructuring and modal synthesis technique applied to vehicle fe data. In *Twenty Fifth International Conference on Noise & Vibration Engineering*, 2000.

[18] D. de Klerk, D.J. Rixen, and S.N. Voormeeren. General Framework for Dynamic Substructuring: History, Review and Classification of Techniques. *AIAA Journal*, 46(5):1169–1181, May 2008.

[19] D. de Klerk. *Dynamic Response Characterization of Complex Systems through Operational Identification and Dynamic Substructuring*. PhD thesis, Delft University of Technology, Delft, the Netherlands, October 2008.

[20] R.R. Craig. Coupling of Substructures for Dynamic Analyses – An Overview. In *Proceedings of AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*, pages 1573–1584, April 2000.

[21] D.J. Rixen. *Numerical methods in Engineering Dynamics*. TU Delft, 2008.

[22] D.J. Rixen. *Engineering Dynamics*. TU Delft, 2007.

[23] Wen-Hwa Shyu, Zheng-Dong Ma, and Gregory M. Hulbert. A new component mode synthesis method: Quasi-static mode compensation. *Finite Elements Anal. Des.*, 24(4):197–298, 1997.

[24] R.J. Guyan. Reduction of Stiffness and Mass Matrices. *AIAA Journal*, 3:380, February 1965.

[25] Yee-Tak Leung. An Accurate Method of Dynamic Condensation in Structural Analysis. *International Journal of Numerical Methods in Engineering*, 12:1705–1715, 1978.

[26] D. J. Rixen. A dual Craig-Bampton method for dynamic substructuring. *Journal of Computational and Applied Mathematics*, 168:383–391, 2004.

[27] Michel Géradin and Alberto Cordona. *Flexible Multibody Dynamics – A Finite Element Approach*. John Wiley & Sons, Chichester, England, 2001.

[28] D.J. Rixen. A dual Craig-Bampton method for dynamic substructuring. *Journal of Computational and Applied Mathematics*, 2004.

[29] D.J. Rixen. Dual Craig-Bampton with enrichment to avoid spurious modes. In *Proceedings of the Twenty Seventh International Modal Analysis Conference, Orlando, FL*, 2009.

[30] R.M. Hintz. Analytical Methods in Component Modal Synthesis. *AIAA Journal*, 13(8):1007–1016, 1975.

[31] A. Majed, E.E. Henkel, and C. Wilson. Improved Method of Mixed-Boundary Component-Mode Representation for Structural Dynamic Analysis. *AIAA Journal of Spacecraft and Rockets*, 42:825–831, 2005.

[32] Ji-Bao Qiu, F.W. Williams, and Ren-Xi Qiu. A new exact substructure method using mixed modes. *Journal of Sound and Vibration*, 266:737–757, 2003.

[33] R.L. Mayes and E.C. Stasiunas. Combining Lightly Damped Experimental Substructures with Analytical Substructures. In *Proceedings of the Twentyfifth International Modal Analysis Conference, Orlando, FL*, Bethel, CT, 2007. Society for Experimental Mechanics.

[34] M.S. Allen and R.L. Mayes. Comparison of FRF and Modal Methods for Combining Experimental and Analytical Substructures. In *Proceedings of the Twentyfifth International Modal Analysis Conference, Orlando, FL*, Bethel, CT, 2007. Society for Experimental Mechanics.

[35] D. de Klerk, D.J. Rixen, S.N. Voormeeren, and F. Pasteuning. Solving the RDoF Problem in Experimental Dynamic Substructuring. In *Proceedings of the Twenty Sixth International Modal Analysis Conference, Orlando, FL*, Bethel, CT, February 2008. Society for Experimental Mechanics. Paper no. 129.

[36] Daniel J. Rixen. *Substructuring and Dual Methods in Structural Analysis*. PhD thesis, Université de Liège, 1997.

[37] C. Bernardi, Y. Maday, and T. Patera. A new non conforming approach to domain decomposition: the Mortar Element Method. In H. Brezis and J.L. Lions, editors, *Nonlinear Partial Differential Equations and their Applications*. Pitman, London, 1994.

[38] R.R. Craig and C.J. Chang. A review of substructure coupling methods for dynamic analysis. *NASA. Langley Res. Center Advan. in Eng. Sci.*, 2(CR-2781):393–408, 1976.

[39] E. Balmès. Use of generalized interface degrees of freedom in component mode synthesis. In *Proceedings of the Fourteenth International Modal Analysis Conference, Dearborn, MI*, February 1996.

[40] Wolfgang Witteveen and Hans Irschik. Efficient modal formulation for vibration analysis of solid structures with bolted joints. In *Proceedings of the Twenty Fifth*

International Modal Analysis Conference, Orlando, FL, Bethel, CT, February 2007. Society for Experimental Mechanics.

[41] D.-M. Tran. Component Mode Synthesis Methods Using Interface Modes: Application to Structures with Cyclic Symmetry. *Computers & Structures*, 79:209–222, 2001.

[42] Randall J. Allemang. The modal assurance criterion - twenty years of use and abuse. In *Proceedings of the Twenty First International Modal Analysis Conference, Kissimmee, FL*, 2003.

[43] E. Muljadi and C.P. Butterfield. Pitch-Controlled Variable-Speed Wind Turbine Generation. In *IEEE Industry Applications Society Annual Meeting*, 1999.

[44] J.C. O'Callahan, P. Avitabile, and R. Riemer. System Equivelant Reduction Expansion Process (SEREP). In *Proceedings of the Seven International Modal Analysis Conference, Las Vegas, NV*, pages 29–37, Bethel, CT, February 1989. Society for Experimental Mechanics.

[45] ISO, Geneva, Switzerland. *ISO 6336-1: Calculation of load capacity of spur and helical gears – Part 1: Basic principles, introduction and general influence factors*, second edition, 2006.

[46] M. Kubur, A. Kahraman, D. M. Zini, and K. Kienzle. Dynamic Analysis of a Multi-Shaft Helical Gear Transmission by Finite Elements: Model and Experiment. *Journal of Vibrations and Acoustics*, 126:398–406, July 2004.

[47] Joris Peeters, Dirk Vandepitte, and Paul Sas. Analysis of internal drive train dynamics in a wind turbine. *Wind Energy*, 9:141–161, 2006.

[48] S.N. Voormeeren. Structural dynamic modeling of a wind turbine yaw gearbox. Technical report, TU Delft and Siemens Wind Power, 2009.

[49] R-W. Ziekman. Dynamic modelling of a planetary gearbox of a wind turbine. Master's thesis, Delft University of Technology, faculty of 3mE, June 2006. - Confidential -.

[50] M. Géradin and D. Rixen. *Mechanical Vibrations – Theory and Application to Structural Dynamics*. John Wiley & Sons, Chichester, England, December 1997.

[51] Svend Gade, Richard Schlombs, Christoph Hundeck, and Christian Fenselau. Operational modal analysis on a wind turbine gearbox. In *Proceedings of the Twenty Seventh International Modal Analysis Conference, Orlando, FL*, 2009.

[52] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, 78(7):808–817, April 2000.

[53] W.Z. Lin, K.H. Lee, S.P. Lim, and Y.C. Liang. Proper orthogonal decomposition and component mode synthesis in macromodel generation for the dynamic simulation of a complex mems device. *Journal of Micromechanics and Microengineering*, 13:646–654, 2003.

[54] Ahmed A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, Cambridge, United Kingdom, second edition, 1998.

[55] S.N. Voormeeren. Coupling Procedure Improvement & Uncertainty Quantification in Experimental Dynamic Substructuring – Validation and Application in Automotive Research. Master's thesis, Delft University of Technology, Delft, the Netherlands, October 2007.

[56] H.A.D. den Dekker. *Dynamic Substructuring Tool User Manual*, 2009.

APPENDIX A

Construction of Boolean Matrices

This appendix [55] illustrates the construction of the Boolean matrices $\boldsymbol{B}$ and $\boldsymbol{L}$. To this end, the general system shown in figure A.1 is considered: this figure schematically shows the coupling of two general substructures. Both substructures consist of 3 nodes; substructure A has 4 degrees of freedom while substructure B holds 5 DOF.

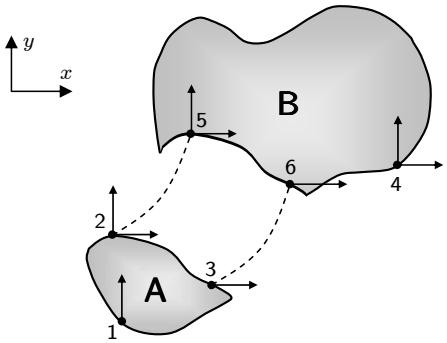


Figure A.1: Coupling of two general substructures.

In this example, nodes 2 and 3 of substructure A are coupled to nodes 5 and 6 of substructure B, respectively. So, three compatibility conditions should be satisfied:

$$\begin{cases} u_{2x} = u_{5x} \\ u_{2y} = u_{5y} \\ u_{3x} = u_{6x} \end{cases} \tag{A.1}$$

To express this condition as in equation 2.7, i.e. $\boldsymbol{Bu} = \boldsymbol{0}$, the signed Boolean matrix $\boldsymbol{B}$ must be constructed. The total vector of degrees of freedom $\boldsymbol{u}$ is:

$$\boldsymbol{u} = \begin{bmatrix} u_{1y} & u_{2x} & u_{2y} & u_{3x} & u_{4x} & u_{4y} & u_{5x} & u_{5y} & u_{6x} \end{bmatrix}^T$$

The signed Boolean matrix $\boldsymbol{B}$ is now found as:

$$\boldsymbol{B} = \begin{array}{c} \begin{matrix} u_{1y} & u_{2x} & u_{2y} & u_{3x} & u_{4x} & u_{4y} & u_{5x} & u_{5y} & u_{6x} \end{matrix} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

Every coupling, or equivalently, every compatibility condition , corresponds to a line in the Boolean matrix $\boldsymbol{B}$. Therefore, in the general case where the coupled substructures

comprise n degrees of freedom of which m are coupled interface DOF, the matrix $\boldsymbol{B}$ has size m-by-n. In this example, $n = 9$ and $m = 3$; the size of $\boldsymbol{B}$ is 3-by-9. It can easily be seen that the condition $\boldsymbol{Bu} = \boldsymbol{0}$ is equivalent to the three compatibility equations in equation (A.1).

From this signed Boolean matrix, the Boolean localization matrix $\boldsymbol{L}$ is found by computing the nullspace. In this example, this gives:

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The set of unique interface DOF that is chosen for this example is found as[1]:

$$\boldsymbol{q} = \begin{bmatrix} u_{1y} & u_{4x} & u_{4y} & u_{5x} & u_{5y} & u_{6x} \end{bmatrix}^T$$

Indeed, the Boolean matrix $\boldsymbol{L}$ transforms this unique set of degrees of freedom to the total set of DOF:

$$\boldsymbol{u} = \boldsymbol{Lq} = \begin{bmatrix} u_{1y} \\ u_{5x} = u_{2x} \\ u_{5y} = u_{2y} \\ u_{6x} = u_{3x} \\ u_{4x} \\ u_{4y} \\ u_{5x} \\ u_{5y} \\ u_{6x} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1y} \\ u_{4x} \\ u_{4y} \\ u_{5x} \\ u_{5y} \\ u_{6x} \end{bmatrix}$$

In addition, the Boolean localization matrix $\boldsymbol{L}$ describes the force equilibrium naturally as well:

$$\boldsymbol{L}^T\boldsymbol{g} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ g_{2x} \\ g_{2y} \\ g_{3x} \\ 0 \\ 0 \\ g_{5x} \\ g_{5y} \\ g_{6x} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ g_{2x} + g_{5x} \\ g_{2y} + g_{5y} \\ g_{3x} + g_{6x} \end{bmatrix} = \boldsymbol{0}$$

[1] the interface DOF of substructure B are retained.

In order to satisfy the equilibrium condition , the connection forces on dual degrees of freedom must thus sum to zero.

Finally, it should be noted that there is another way to obtain the matrix $\boldsymbol{L}$ from $\boldsymbol{B}$. To this end, partitioning of the global set of DOF into sets of unique ($\boldsymbol{u}_u$) and redundant coordinates ($\boldsymbol{u}_r$) is required. The unique coordinates are all the internal DOF plus one set of interface DOF. The redundant coordinates are formed by the dual interface DOF. Partitioning equation 2.7 then gives:

$$\begin{bmatrix} \boldsymbol{B}_{rr} & \boldsymbol{B}_{ru} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_r \\ \boldsymbol{u}_u \end{bmatrix} = \boldsymbol{0}$$

Here $\boldsymbol{B}_{rr}$ is a non-singular square submatrix of $\boldsymbol{B}$. From this partitioned equation, it is clear that the redundant DOF can be found from the unique DOF as:

$$\boldsymbol{u}_r = -\boldsymbol{B}_{rr}^{-1}\boldsymbol{B}_{ru}\boldsymbol{u}_u$$

Since the Boolean localization matrix $\boldsymbol{L}$ builds the set of global DOF from a set of unique DOF (i.e. $\boldsymbol{u} = \boldsymbol{L}\boldsymbol{q}$), one finds $\boldsymbol{L}$ directly from the partitioned compatibility equation:

$$\boldsymbol{u} = \boldsymbol{L}\boldsymbol{q} = \begin{bmatrix} \boldsymbol{u}_r \\ \boldsymbol{u}_u \end{bmatrix} = \begin{bmatrix} -\boldsymbol{B}_{rr}^{-1}\boldsymbol{B}_{ru} \\ \boldsymbol{I}_{uu} \end{bmatrix} \boldsymbol{u}_u \tag{A.2}$$

In this example, this gives:

$$\boldsymbol{B} = \begin{array}{c} \begin{matrix} u_{2x} & u_{2y} & u_{3x} & u_{1y} & u_{4x} & u_{4y} & u_{5x} & u_{5y} & u_{6x} \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

Computing the Boolean localization matrix from equation (A.2) then gives:

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

It can easily be verified that this is indeed equal to the nullspace of the partitioned $\boldsymbol{B}$ matrix.

APPENDIX B

The DS Tool and PP Tool

The framework presented in this MSc. thesis has been integrated into a MatLab based toolbox, consisting of the DS (Dynamic Substructuring Tool) and the PP Tool (Post-Processing Tool). The DS Tool is used for the assembly of component models and uses the assembly methods presented in chapter 2 and also enables assembly of substructures with non-conforming meshes as described in chapter 4. The interface reduction methods presented in chapter 5 are also integrated within the DS Tool. Assembled structures can be saved in a standard format and loaded into the PP Tool for further analyses. In addition, the PP Tool is also used to visualize and save the analyses results.

This appendix contains a brief manual for the DS Tool, which is taken from the more extensive manual written by H.A.D. den Dekker [56]. The PP Tool is presented in section B.3 and is still in development.

B.1 The DS Tool

If the substructures are properly built and the interfaces or interface nodes are correctly identified, the *Dynamic Sub structuring Tool* is easy to use. The user interface is opened and the Matlab `.mat` files of the substructures are loaded into the program. Once this is done, the domain of the assembly is selected and the type of interface connection is set. Then, the coupling is defined, and checked in the assembly list. Now the system can be assembled and possibly reduced using Mixed Craig Bampton reduction and suiting assembly. Finally, the interfaces can be reduced and the tool outputs the assembled `.mat` model. This can be saved and analyzed in the PP Tool.

This section explains how the user interface can be used to assemble substructures. Every subsection is split up into a description and the example. The latter is a walk though [WT] of assembling a simple structure of three beams as shown in figure B.2.

WT: Goal of Assembly	In this example the full physical dynamic models of three beams are going to be assembled as shown in figure B.2(a) and exported to the PP Tool for further analysis, as shown in figure B.2(b). This system will be assembled per node and all displacement DoF (x, y, z) will be coupled in assembly. To reduce computation time in the future, the 'physical' models will be reduced using the Mixed Craig Bampton algorithm. furthermore, interface reduction will be applied to the entire structure after assembly.

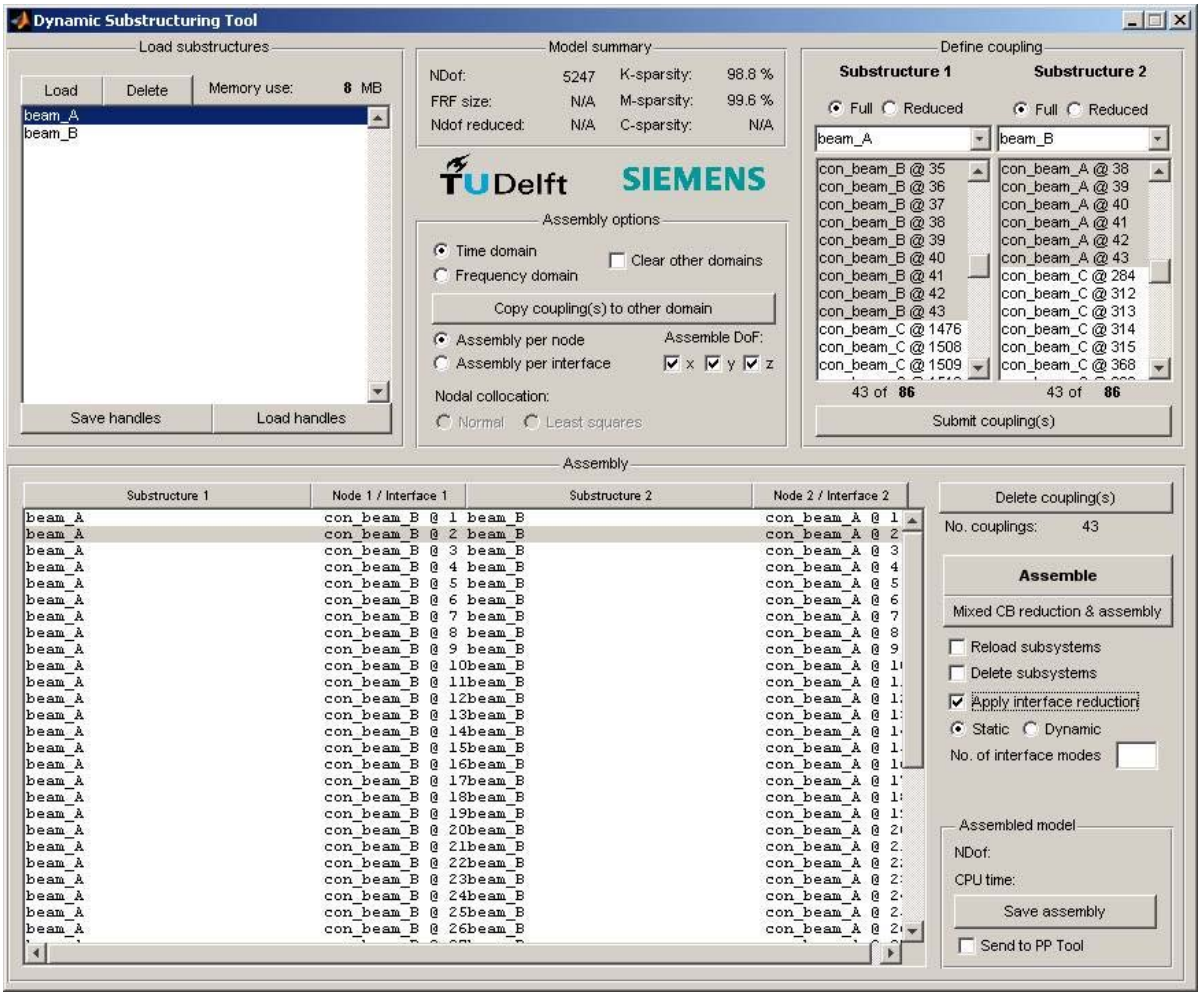


Figure B.1: Graphical User Interface

(a) ANSYS (b) PP Tool

Figure B.2: Sample Assembly

B.1.1 Load Substructures

The `Load Substructures` section in the graphical user interface (figure B.1) is straightforward. The section is used to load dynamic models of the substructures to be assembled.

- By pressing Load, a window opens where the models of the substructures can be connected. Note that even though any `.mat` file can be loaded, only the models that are structured correctly (see section B.2.1) will be assembled without errors.
- The Delete button will delete the selected model file.
- The Save handles Load handles buttons are not implemented yet.
- The `Memory use:` indicator shows how much ram memory will be loaded into the RAM of the computer when the systems assemble. when this value exceeds the available ram space of the processing computer, the system will be forced to save the data in a 'virtual memory' space, located on the hard drive. This will have a large effect on the computation time, and it is therefor advised to prevent this.

WT Step 1: loading substructures	Load the three structures `prep_beam_A.mat`, `prep_beam_B.mat` and `prep_beam_C.mat` into the DS Tool. The total data to be loaded into the RAM should be around 12 MB.

B.1.2 Model Summary

The model summary section provides information about a loaded substructure. When one of the loaded structures is selected it reads the following data:

- The `NDoF:` indicator shows how many DoF the full physical mode counts.
- the `FRF size:` shows the $(n \times m \times f)$ size of the three dimensional FRF matrix in the model, where n are the number of responses (rows), m the number of excitations (collums) and f the number of frequencies (third dimension).
- The `NDoF reduced:` indicator shows how many DoF the reduced model counts.
- the `K-` `M-` and `C-sparsity:` indicators indicate how sparse the component matrices are. In fact they represent the ratio between zeros in and the size of the matrix.

WT Step 2: checking the substructures	Check the beam models loaded into the tool. Beam A for example, should have 5,247 DoF. As the model is not represented in the frequency domain nor reduced, the FRF size and reduced NDoF readings should be unavailable. The K amd M sparsity should be respectively 98.8% and 99.6%, hence containing few non diagonal terms.

Figure B.3: Warning: delete couplings

B.1.3 Assembly Options

In the Assembly Options section the required options for the assembly of the substructures can be set. Note that these options apply to all the couplings that are assembled and it is not possible yet to assemble 'subassemblies'. The following options can be set:

- The `Time domain / Frequency domain` selector selects in which domain the substructures will be coupled.
- the `Clear other domains` option clears the domain that is not selected in order to save memory in the assembly process.
- The 

button copies the couplings from the selected domain to the other. If the system is already coupled in one domain, shown in the assembly list at the bottom of the user interface, pressing this button will generate the same couplings in the other domain.
  - Note that the system deletes all the previously defined couplings in the latter. Hence the warning shown in figure B.3 will pop up to ensure the right choice has been made.
  - If both domains need to be coupled but required memory in this process is too large, the assemblies can be performed separately for each domain. However, the coupling can still be defined in one step. When the coupling is defined in one domain, they can be copied in the other domain as well. Following this, one domain can be deleted (however retaining the assembly information of this domain), and the system can be assembled. Now the substructures can be reloaded, and the assembly can take place in the other domain.. This way the user can save time defining the coupling and also reduced the memory used in the assembly processes.
- The `Assembly per node/Assembly per interface` selector selects how the tool connects the nodes. When the first option is selected the nodes are to be coupled manually per interface. If the latter is selected, only the to be coupled interfaces need to be selected.
- The `Assemble DoF:` selector selects which types of DoF will be assembled (x, y, z).
- The `Nodal collocation:` selector selects what type of collocation algorithm is used in coupling the nodes. Hence the selector is only active when the assembly is coupled per interface.

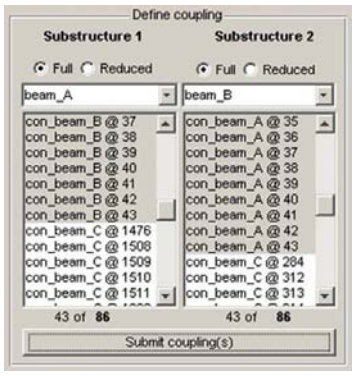


Figure B.4: Selecting multiple nodes to connect interfaces of beam A to B

B.1.4 Define Coupling

The interface nodes can be coupled in the 'Define coupling' section, containing two tables listing the connections available per model. when the assembly is to be assembled per node (this is selected in the assembly options section) a list of interface nodes will appear in the list. When the assembly is connected per interface a set of interfaces will appear in the list. In both tables the to be coupled objects can be selected and when Submit coupling(s) is pressed the coupling will be made and transferred to the assembly list below. Note that if the structure is assembled per node and the bookkeeping of the models is tidy, the connecting interface nodes are listed in the same order for both substructures. In this case, more nodes can be coupled at once by selecting a range of nodes in both substructures.

WT Step 3: Coupling the substructures	Since the substructures are loaded and verified, the coupling can be defined. First the correct assembly options need to be selected. Set the assembly type to `time domain` and select `clear other domains`, as the assembly will be coupled only in the time domain. The structure can be assembled per node or per interface, but for sake of the tutorial we assemble per node for all DoF directions. Now that the options are set the nodes to be coupled can be selected. First, interfaces of `beam_A` and `beam_B` will be connected. As shown in figure B.4, multiple nodes can be selected to speed up the coupling process. once the of nodes of beam A connecting to beam B and the nodes of beam B connecting to nodes of beam A are selected, Submit coupling(s) can be pressed and the nodes should appear in the assembly section. This process is repeated for the connection of interfaces at beam B to C and at beam A to C.

B.1.5 Assembly

The assembly section can be divided into three sections; the left hand side, the right hand side and the `Assembled model` section. The left hand side is a table providing

information regarding the defined coupling and on the right hand side the assembly options can be defined. The `Assembled model` section provides information and options regarding the new model after the assembly.

B.1.5.1 Assembly: Left Hand Side

The table on the left hand side is simple to read. the first and third columns of the table indicate which substructures are concerned in the coupling. The second and fourth columns indicate which node or which interface is concerned in the coupling. Note couplings of all domains can be represented here.

B.1.5.2 Assembly: Right Hand Side

The right hand side provides the assembly options:

- The Delete coupling(s) button deletes the selected couplings in the assembly table on the left hand side.
- The `No. couplings:` indicators indicates how many couplings are created in the tool, which can be used to verify whether the assembled model concurs with its design.
- The `Reload subsystems` selector reloads the substructures after assembly. This option comes to use when with mixed assemblies, where the substructures are altered.
- The `Delete subsystems` selector deletes the substructures after assembly. This to minimize memory use in the process.
- The `Apply interface reduction` applies interface reduction after assembly. *Note that interface reduction has currently not been tested for the Mixed Craig Bampton reduction and assembly method.*
- The `Static / Dynamic` selector selects whether to reduce the interface with the static or dynamic method. Note that Dual Craig Bampton interfaces cannot be reduced statically at this point in the tool.
- The `No. of interface modes` field specifies how many modes should be represented in the interface reduction.
- Finally, the Assemble button initiates the assembly process. The completion of this process is indicated in the Assembled model section.
- The Mixed CB reduction & assembly button starts the mixed reduction and assembly process. When this button is pressed, the window shown in figure B.5(a) pops up. The window asks how many modes should be included to represented to describe the fixes and/or free vibration modes of the system in reduction. After this a second window, shown in figure B.5(b), pops up asking whether to couple using primal or dual assembly

(a) Required number of modes (b) Primal or dual assembly

Figure B.5: Pop ups for Mixed Craig Bampton Reduction

techniques, referring to using fixed or free modes in reduction respectively. After these selections the tool starts to reduce the substructures, assemble the system and reduce the interface DoF when selected.

B.1.5.3 Assembled Model

When the assembly is completed the post processing information and options are shown in the assembled model section:

- The `NDof:` inidicator indicates how many DoF are used to present the assembled model.
- The `CPU time:` indicator indicated how long it took the CPU to reduce and or assemble the structure
- The 

button saves the assembly to a `.mat` file. see section B.2.1 regarding the structure of this data.
- The `send to PP tool` selector opens the model in the PP tool when saved.

WT Step 4: Assembly & Export	Now that the nodes of the beams are properly coupled (129 couplings) , the system can be assembled. Select to delete the subsystems to save memory and to apply interface reduction after assembly. Choose to truncate the interface modes to 10, which is enough for this problem. Then select primal or dual assembly (or assemble twice and select both and compare the results). At this point the system start to reduce, assemble and interface reduce the system. This process should not take too long on todays PC's. When selecting primal assembly, the assembled model should have 804 DoF. This can be saved only or both saved and exported to the PP Tool for further analysis.

B.2 Data Handling in the DS Tool

B.2.1 Data Structure of Component Models

The tool loads the dynamic model of the substructures in Matlab `.mat` format. This file contains the structure `model`, split up into five sections, as shown in figure B.6.

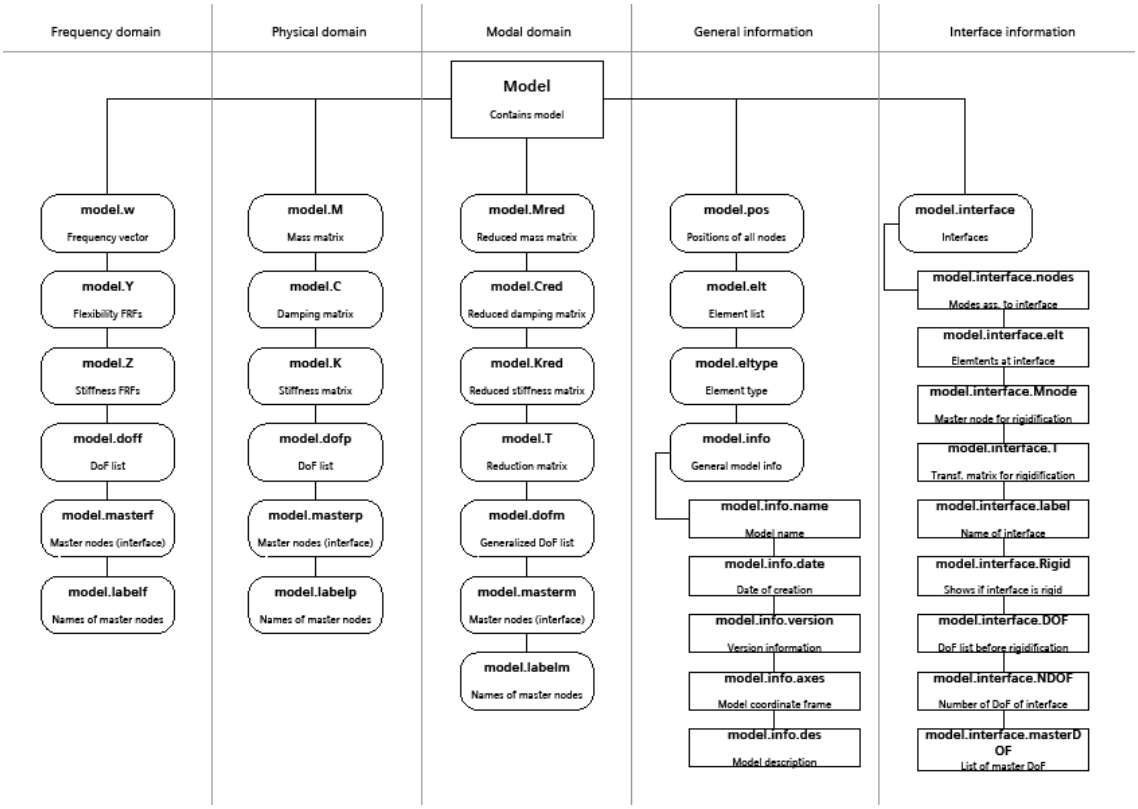


Figure B.6: Data structure of the dynamic models of the substructures

The 'Physical Domain' contains data regarding the 'full' physical model. The domain contains the following data:

- `model.M`: the full $(n \times n)$ mass matrix of the structure, where n is the number of DoF.
- `model.C`: the full $(n \times n)$ damping matrix of the structure, where n is the number of DoF.
- `model.K`: the full $(n \times n)$ stiffness matrix of the structure, where n is the number of DoF.
- `model.dofp`: The complete list of DoF, its length being the number of DoF n. The order of the list is directly related to the mass, damping and stiffness matrices. The coding method of the list is found in section B.2.3.
- `model.masterp`: The list of interface nodes, its length being the number of interface nodes i. In contrary to the complete DoF set mentioned above, the order of the list is irrelevant, as it only lists which of the nodes of the complete set is at the the interface.
- `model.labelp`: The list labeling the interface nodes, its length being the number of interface nodes i. The labels in this list are connected to the interface nodes in the previous list via their order.

The 'Modal Domain' contains information regarding the modal shapes of the system, often truncated to reduce the system. The domain contains the following data:

- `model.Mred`: The reduced $(m \times m)$ mass matrix of the structure, where m is the number of DoF of the reduced dynamic equation set.
- `model.Cred`: The reduced $(m \times m)$ damping matrix of the structure, where m is the number of DoF of the reduced dynamic equation set.
- `model.Kred`: The reduced $(m \times m)$ stiffness matrix of the structure, where m is the number of DoF of the reduced dynamic equation set.
- `model.T`: The $(n \times m)$ reduction matrix of the system, n being the number of DoF of the reduced set and m of the full set.
- `model.dofm`: The reduced array of DoF, its length being the number of DoF of the reduced set m. The order of the list is directly related to the reduced mass, damping and stiffness matrices.
- `model.masterm`: The array of interface nodes of the reduced set, its length being the number of interface nodes j. In contrary to the reduced DoF set mentioned above, the order of the array is irrelevant, as it only lists which of the nodes of the reduced set is at the interface.
- `model.labelm` The array labeling the interface nodes of the reduced system, its length being the number of interface nodes j. The labels in this array are connected to the interface nodes in the previous array via their order.

The 'Frequency Domain' contains the same type of dataset as the physical domain, except now described in the frequency domain. This datatype is not fully developed yet in the DSTool.

The 'General Information' section mainly relates the numeric dynamic model to the real component. Furthermore it contains some administrative data. The following objects are included:

- `model.pos` The $(k \times (l+1))$ position matrix containing the position data of the nodes, k being the number of nodes and l the number of DoF each node is represented by $(l \leq 6)$. The left column indicated which node is referred to and the column right to that indicate its position. For translational positions, units are in m and for rotational positions in rad.
- `model.elt` : The $(o \times (p+3))$ element matrix where o denotes the number of elements and p the number of nodes per element. The first p columns of the matrix define the nodes representing the elements, the $p+1$ and $p+2$ elements describe its properties `MatID` and `ProID` (FEM properties) and the right column assigns the respective element numbers.
- `model.eltype`: Currently unassigned, will be used to connect the type of material of the element to the element by using the same arrangement of elements as the arrangement in the element matrix above.
- `model.info`: The info structure contains qualitative information regarding the substructure:
  - `model.info.name`: Name of the substructure
  - `model.info.date`: Date of creation
  - `model.info.version`: Version number
  - `model.info.axes`: Array containing the axes used in the structure (i.e: $x, y, z, \phi, \varphi, \gamma$)
  - `model.info.des`: Qualitative description of the model and any other relevant information

The 'Interface Information' section includes all the data regarding the interfaces of the substructure. It provides information on the rigid or full interface and is essential when assembling the substructures per interface rather than per node. The section contains the following information:

- `model.interface`: The $(1 \times r)$ structure, where r denotes the number of interfaces on the substructure. It contains other fields describing these interfaces and contain the following fields:
  - `model.interface.nodes`: The $(s \times 4)$ nodes matrix, where s denotes the number of nodes in the interface. the left column of the matrix identifies

which global nodes are interface nodes, and the right columns denote the (x, y, z) coordinates of the nodes.

- `model.interface.elt`: The $(t \times (p+3))$ element matrix, where t denotes the number of elements at the interface and p the number of nodes per element of the substructure. Like in the global element list, the first p columns of the matrix define the nodes representing the elements, the $p+1$ and $p+2$ elements describe its properties `MatID` and `ProID` (FEM properties) and the right column assigns the respective element numbers.
- `model.interface.rigid`: The boolean expressing whether the interface is rigidified or not.
- `model.interface.Mnode`: An integer identifying which node represents the interface when the interface is rigidified.
- `model.interface.T`: The $((s \cdot l) \times 6)$ rigidification matrix, where s denotes the number of nodes in the interface and l the number of DoF each node is represented by $(l \leq 6)$. The matrix transforms the interface DoF to 6 DoF describing the displacements and rotations of the entire (rigid) interface.
- `model.interface.label`: The name of the interface, often describing to which other interface it will be connected.
- `model.interface.DOF`: The $(s \cdot l \times (2 + l))$ matrix, where s denotes the number of nodes in the interface and l the number of DoF each node is represented by $(l \leq 6)$. The matrix identifies the DoF before rigidification; the left column identifies the DoF and the right colums their coordinates. It facilitates the rigidification process and is not used in the DS tool.
- `model.interface.nDOF`: The number of DoF before or after rigidification
- `model.interface.Masterdof`: DoF list after rigidification, its length being the number of DoF of the master node v.

B.2.2 Data Structure of Assembled Model

After assembly, the tool generates output in the form of the dynamic model of the assembled structure as shown in figure B.7 Like the required input structure, this is a `.mat` file. However, the 'physical' and the 'modal' domain and now merged into one 'time' domain, as only one of the models is used in assembly. the file now contains the following data:

The Time Domain contains the data of the assembled model in either the physical or modal domain.

- `assembly.M`: The $(n \times n)$ mass matrix , where n is the number of unique DoF in the dynamic model of the assembly.

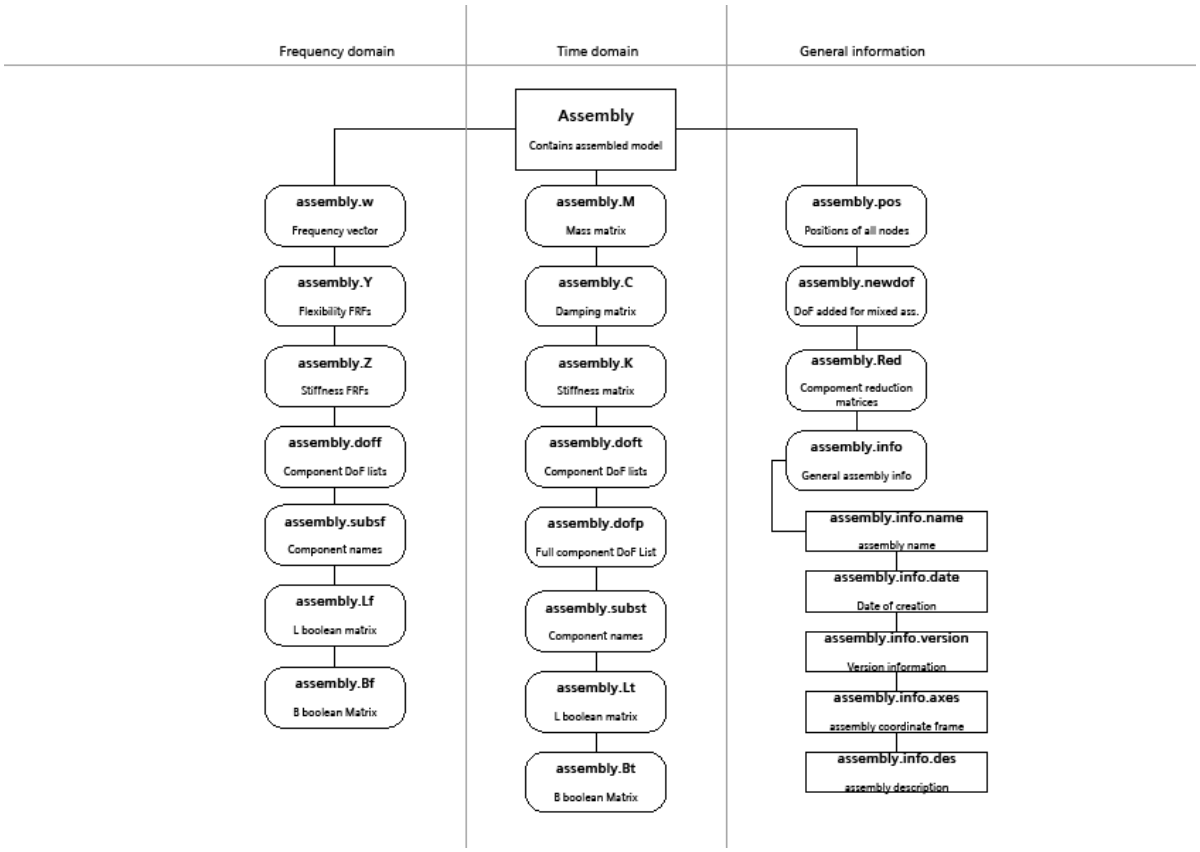


Figure B.7: Data structure of the dynamic model of the assembly

- `assembly.C`: The $(n \times n)$ damping matrix, where n is the number of unique DoF in the dynamic model of the assembly.
- `assembly.K`: The $(n \times n)$ stiffness matrix , where n is the number of unique DoF in the dynamic model of the assembly.
- `assembly.doft`: The $(m \times q)$ Component DoF matrix, where m is the sum of the DoF of the individual components and q is the number of components. If the matrix is divided into sub matrices containing the DoF lists it becomes diagonal, each column representing a component.
- `assembly.dofp`:The $(o \times q)$ Full component DoF matrix, where o is the sum of the DoF of the full dynamic model of the individual components and q is the number of components. If the matrix is divided into sub matrices containing the DoF lists it becomes diagonal, each column representing a component.
- `assembly.subst`: The horizontal array of component names, its length being the number of components q.
- `assembly.Lt`: The $(m \times n)$ L boolean matrix used in assembling the substructure.
- `assembly.Bt`: The $((m - n) \times m)$ B boolean matrix used in assembling the substructure.

The Frequency Domain contains the data of the assembled model in the frequency domain.

- `assembly.w`: The frequency vector, ist length being the number of frequency points f.
- `assembly.Y`: The $(f \times f \times f)$ receptance/dynamic flexibility matrix , where n is the number of unique DoF in the dynamic model of the assembly.
- `assembly.Z`: The $(n \times n \times f)$ dynamic stiffness matrix , where n is the number of unique DoF in the dynamic model of the assembly.
- `assembly.doff`: The $(m \times q)$ Component DoF matrix, where m is the sum of the DoF of the individual components and q is the number of components. If the matrix is divided into sub matrices containing the DoF lists it becomes diagonal, each column representing a component.
- `assembly.subsf`: The horizontal array of component names, its length being the number of components q.
- `assembly.Lf`: The $(m \times n)$ L boolean matrix used in assembling the substructure.
- `assembly.Bf`: The $((m - n) \times m)$ B boolean matrix used in assembling the substructure.

The General Info section contains general information regarding the assembly.

- `assembly.pos`: The $(k \times (l+2))$ position matrix containing the position data of the nodes, k being the number of nodes and l the number of DoF each node is represented by ($l \leq 6$). The left column indicated which node is referred to and the columns right to that indicate its position. For translational positions, units are in m and for rotational positions in rad. The last column indicates the component number.

- `assembly.newdof`: The new DoFlist required with additional DoF introduced in mixed assembly. These are needed later for assembly.

- `assembly.info`: The info structure contains qualitative information regarding the assembly:

  - `assembly.info.name`: Name of the assembly

  - `assembly.info.date`: Date of creation

  - `assembly.info.version`: Version number of the assembly

  - `assembly.info.axes`: Array containing the axes used in the structure (i.e: $x, y, z, \phi, \varphi, \gamma$)

  - `assembly.info.des`: Qualitative description of the system and any other relevant information

- `assembly.red`: The $(1 \times x)$ structure, containing the reduction matrices T per reduced subsystem.

B.2.3 Coding the Degrees of Freedom

The DS Tool needs to know the type of the DoF. This information is provided via their format. The required format per type is presented below:

displacements: `"n.0d"`, where `n` is the number of the node concerned and `d` is the direction of the translation in the reference frame ($d = 1, 2, 3$)

rotations: `"n.0r"`, where `n` is the number of the node concerned and `r` is the axis of the rotation in the reference framce ($r = 4, 5, 6$)

interface forces: `"-n.0d"` where `n` is the number of the node concerned and `d` is the direction of the translation in the reference frame ($d = 1, 2, 3$). Note the `"-"` in front of the identifier, indicating the DoF is a force or moment

interface moments `"-n.0r"` where `n` is the number of the node concerned and `d` is the direction of the translation in the reference frame ($r = 4, 5, 6$). Note the `"-"` in front of the identifier, indicating the DoF is a force or moment

B.3 The PP Tool

Structures assembled in the DS Tool can be loaded into the PP tool for further analyses. A screen shot of the PP Tool is shown in figure B.8. Since the focus of this thesis was on component model reduction techniques and assembly, the PP Tool is still in development. Currently, the PP Tool is able to perform a modal analysis and has several visualization options.

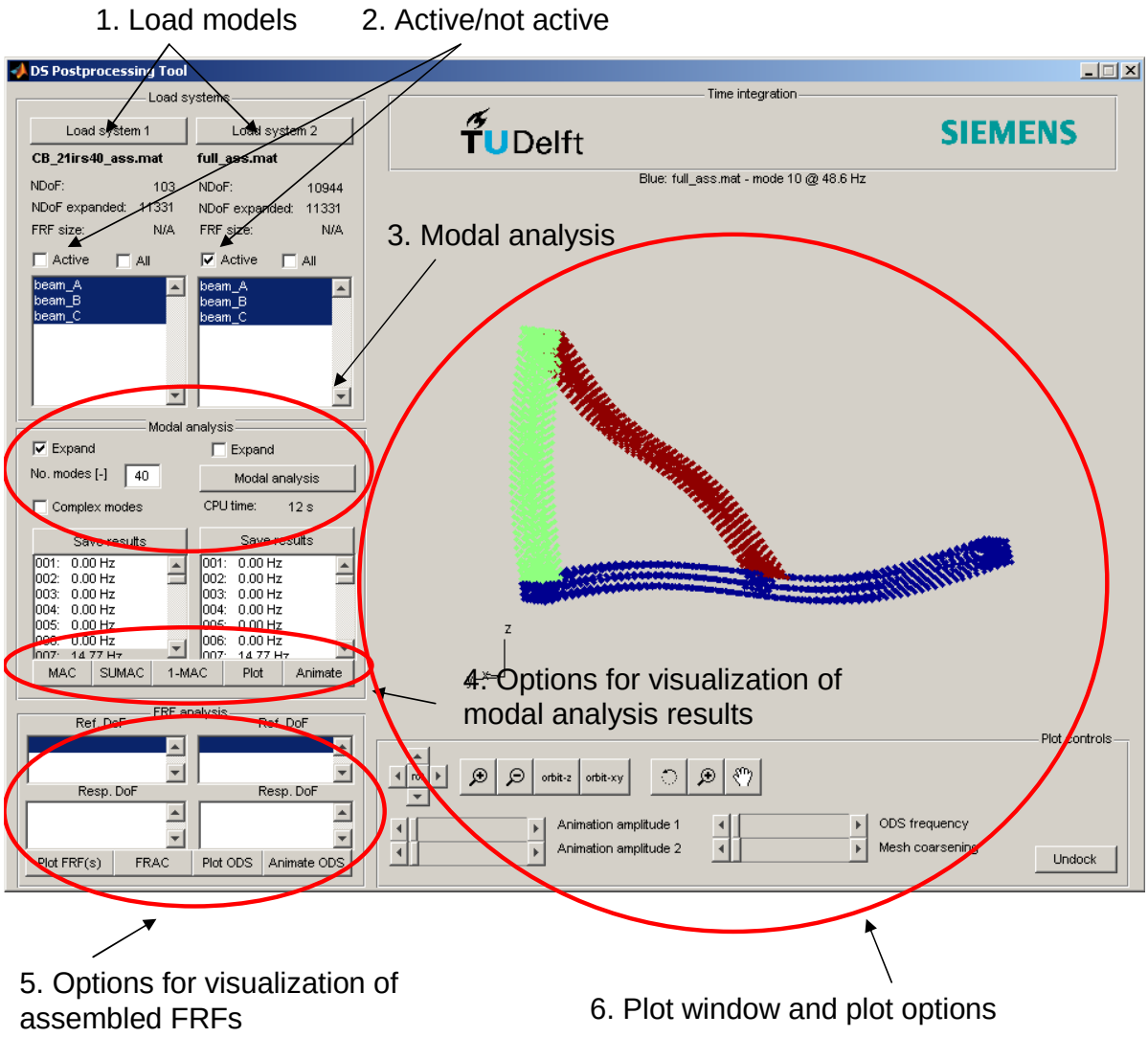


Figure B.8: Screen shot of the PP Tool

The functionalities of the PP Tool are briefly described below.

B.3.1 Load Models

First, one had to load one or two models to use for the analysis. Below the buttons the names and sizes of the different models is shown:

- **NDoF**: The number of degrees of freedom of the time domain model, which can be either a full FE or a reduced model.
- **NDoF expanded**: The sum of all individual substructure degrees of freedom.
- **FRF size**: The number of in and output DoF of the assembled FRF matrices. Note that only square FRF matrices can be handled in this release.

B.3.2 The "active" check boxes

The "active" check boxes are used to denote which of the models is "active" for the computations and/or plotting functions. If one loads a model, the associated check box will automatically be turned on. In figure B.8 it is clear that the second model, `full_ass.mat`, is being used to plot the 10^{th} mode shape.

B.3.3 Modal analysis

Using the "active" check boxes one can determine the model(s) to perform the modal analysis on. Before pressing the "Modal analysis" button, one has to make sure the settings are correct:

- **Expand** If the model contains reduced substructures and one want to visualize the mode shapes and/or perform a (SU)MAC analysis with, for instance a full FE model, one has to check the *Expand* box.
- **No. modes** [-] The number of modes one wants to compute has to be given.
- **Complex modes** If the damping matrix is non-zero, the *Complex modes* box has to checked. In this case a solver is used that allows for complex eigenmodes.

B.3.4 Options for visualization of modal analysis results

Different types of visualization can be done after a model analysis:

- **MAC** A MAC analysis can be performed (see section 5.3.1).
- **MAC** A SUMAC analysis can be performed (see section 5.3.2).
- **1-MAC** The numbers of the diagonal of the MAC matrix are subtracted from 1 and are plotted using a logarithmic scale on the Y-axis.
- **Plot** The mode shape belonging to the selected eigenfrequency is plotted.

- **Animate** The mode shape belonging to the selected eigenfrequency is animated and the actual motion is shown.

B.3.5 Options for visualization of assembled FRFs

If the model contains assembled FRF matrices a number of visualization can be performed:

- **Plot FRF(s)** One can plot the different FRFs by selecting one (or more) input DoF(s) and an output DoF(s).
- **Frequency Response Assurance Criterion (FRAC)** The FRAC is used to compute the correlation between two frequency response functions representing the same input-output relationship and is in function thus similar to the MAC. A more detailed discussion is found in [42]
- **Plot operational deflection shape (ODS)** An operational deflection shape is the deformation of the structure resulting from an excitation at a certain frequency and location.
- **Animate operational deflection shape (ODS)** The operational deflection shapes can also be animated, similar to the animation of mode shapes as described earlier.

B.3.6 Plot window and plot options

All the plots are shown in the plot window. When visualizing (plotting or animating) mode shapes or operational deflection shape one can use the *camera rotation, zoom and orbit* functionalities to visualize the mode shape from any possible point of view. Next to this the amplitudes and/or frequencies can be altered. The results from the MAC, SUMAC and FRAC analyses are also shown in the plot window. Finally, the *Undock* button can be used to open the figure in a new window.

APPENDIX C

Specifications of measurement equipment

This appendix will discuss several details of the Bedplate measurement, which are not discussed in section 7.2.2. In this appendix the list of used equipment and its technical specifications is given in section C.1.

C.1 Specifications of the equipment used for the bedplate measurements

This section gives an overview of the measurement equipment used.

C.1.1 Software

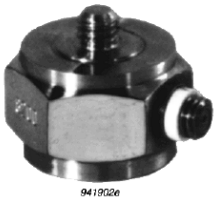
The software used for the bedplate measurement is Pulse LabShop - MTC Exciter, version 13.5.0 from Brüel and Kjær.

C.1.2 Hardware

Brüel and Kjær Vibration Exciter Type 4808

Force rating	112 Newton
Frequency range	5 Hz to 10 kHz
First axial resonance	10 kHz
Maximum bare table accelerations	700 m/s^2

Brüel and Kjær Force Transducer - Type 8200



Force range	1000 N tensile to 5000 N compressive
Resonance frequency	35 kHz
Charge Sensitivity	$\approx$ 4 pC/N
Stiffness	$5 \cdot 10^8$ N/m
Traverse sensitivity	5%
Weight	21 grams
Dimensions (l $\times$ b $\times$ h	18 mm $\times$ 18 mm $\times$ 23 mm

Brüel and Kjær Triaxial DeltaTron Accelerometers - Type 4506B

Sensitivity	10 mV/ms$^{-2}\pm$ 10%
Measurement range	$\pm$ 700 ms^{-2}
Frequency range ($\pm$ 10%)	**X** 0.3 Hz to 5.5 kHz
	Y,Z 0.6 Hz to 3.0 kHz
Mounted resonance frequency	**X** 19 kHz, **Y,Z** 10 kHz
Traverse sensitivity	$<$ 5%
Sensing Element	Piezoelectric, Type PZ 23
Weight	15 grams
Dimensions (l $\times$ b $\times$ h	17 mm $\times$ 17 mm $\times$ 14.5 mm

Brüel and Kjær Magnetic mounting clip - UA 1563

Maximum acceleration (with a 17 gram accelerator)	10 g
(Perpendicular to mounting surface)	50 g
Weight	11 grams

C.2 Specifications of the equipment used for the yaw gearbox measurements

TIRA Vibration Exciter TV 51110

Force rating	100 Newton
Frequency range	2 Hz to 7 kHz
First axial resonance	> 6.5 kHz
Maximum bare table accelerations	440 m/s^2

PCB ICP impedance head Type 288D01

Force range (peak)	± 2244 N
Resonance frequency	5 kHz
Sensitivity (acc.)	100 mV/g
Sensitivity (force)	22.4 mV/N
Resonance frequency	$\geq$ 20 kHz
Stiffness	$3.5 \cdot 10^8$ N/m
Traverse sensitivity	5%
Weight	19.2 grams
Dimensions hex $\times$ h	11/16 in $\times$ 20.83 mm

PCB Triaxial Accelerometers - Type 356B21

Sensitivity	9.73 mV/ms^{-2}± 10%
Measurement range	± 700 ms^{-2}
Frequency range (± 10%)	**X** 2 Hz to 7 kHz
	Y,Z 2 Hz to 10 kHz
Resonance frequency	$\geq$ 55 kHz
Traverse sensitivity	$<$ 5%
Sensing Element	Ceramic
Weight	4 grams
Dimensions (l $\times$ b $\times$ h	10.2 mm $\times$ 10.2 mm $\times$ 10.2 mm

Index