Hochschule Düsseldorf
University of Applied Sciences

Fachbereich Maschinenbau und Verfahrenstechnik
Faculty of Mechanical and Process Engineering

# Calculation of Dynamic Wind Loads on Heliostats Using FEM and Multibody Dynamics

A thesis submitted for the degree of

Master of Science

presented to the

Faculty of Mechanical and Process Engineering

University of Applied Sciences Düsseldorf

in the study program

Mechanical Engineering

by

Atli Tobiasson Helmer

Student ID: 920844

Examiner: Prof. Dr.-Ing. habil. Martin Ruess
Co-Examiner: Dr.-Ing. Andreas Pfahl

August 2024

# Calculation of Dynamic Wind Loads on Heliostats Using FEM and Multibody Dynamics

Master's Thesis Assignment, Atli Tobiasson Helmer, Student ID: 920844
DLR SF-SKT

**Background**

Solar tower systems are a promising technology for renewable electricity and process heat generation. The largest cost factor is the mirrors, known as heliostats, which track the sun and concentrate sunlight[1]. At DLR, a heliostat has been developed that is more than 20% cheaper than existing ones and is set to be brought to market[2]. The design of these heliostats is significantly influenced by wind loads, which must be carefully considered, including resonance effects that are not yet fully understood. These effects are now to be determined.

**Research Objective**

The research objective of this thesis is to determine the maximum wind loads on heliostats in the safety position with the mirror surface oriented horizontally. In particular, the focus is on determining the amplification factor between transient and static loads.

**Methodology**

In this thesis, a computationally efficient multibody simulation model will be developed to study the wind loads on heliostats. This model will be verified and optimized using an equivalent finite element model (FEM).

The results are of general interest and should ideally be published in the Journal of Solar Energy and/or presented at the SolarPACES 2024 conference.

**Planned Schedule**

- Determination of the Simulation Method (Weeks 1-3)
  - Research on potential simulation methods
  - Evaluation and selection
- Familiarization with the Simulation Method (Weeks 4-6)
  - Installation of the method
  - Familiarization through tutorials, etc.
- Setup of Calculations (Weeks 7-10)
  - Simplification of the CAD model
  - Division into rigid bodies and elastic bodies
  - Mesh generation
  - Transformation of pressure time series into a suitable format
- Calculations (Weeks 9-11)
  - Mesh study
  - Variation of timestep settings
  - Calculation of deformations, stresses, and inertial forces and moments
- Evaluation (Weeks 12-15)
  - Determination of dynamic load factors
  - Determination of maximum stresses
- Publications (Week 16)
  - Possible contribution to the preparation of a paper for the SolarPACES 2024 conference
  - Possible contribution to writing a paper for the Journal of Solar Energy
  - Writing of the master's thesis

Andreas Pfahl

Digital signiert von Andreas Pfahl
DN: OU=SF-SKT, O=DLR, CN=Andreas Pfahl, E=Andreas.Pfahl@dlr.de
Grund: Ich bestätige die Richtigkeit und Vollständigkeit dieses Dokuments
Ort: Jülich
Datum: 2024.08.14 11:56:16+02'00'
Foxit PDF Reader Version: 2024.2.2

[1] A. Pfahl, J. Coventry, M. Röger, F. Wolfertstetter, F. Vasquez, F. Gross, M. Arjomandi, P. Schwarzbözl, M. Geiger, and P. Liedke. “Progress in Heliostat Development”. Solar Energy (152), pp. 3-37, 2017, https://dx.doi.org/10.1016/j.solener.2017.03.029.

[2] A. Pfahl and V. Dohmen, “Low-Cost Materials for Heliostats”. SolarPACES 2023, Sydney.

## Abstract

At the German Aerospace Center's (DLR) Institute of Solar Research, a new heliostat is being developed with the promise of bringing down the cost of heliostats by more than 20%. The main features in which this heliostat distinguishes itself from other heliostats is in its low part-count and in its size, as its mirror is only 2 $m^2$. This thesis describes the creation of a complex Finite Element Method (FEM) model for stress analyses, then making two simplified models for long-duration studies, which are computationally efficient. Their modeling methods were radically different, one being a multibody model while the other one a FEM shell element model. Only the FEM model led to results of good agreement with the complex model. With this simplified model, the timestamps of maximum deformations can be determined when loaded by a wind load time series that was gained from a wind tunnel test. With the obtained results, a Dynamic Load Factor (DLF) is determined to measure the impact of dynamic loads compared to static ones. Finally, stresses at critical timestamps from the long-duration study are analyzed with the complex model to provide detailed insights into the heliostat's performance.

# Contents

# List of Figures

# List of Tables

# Nomenclature

## Symbols

| | |
|---|---|
| $\rho$ | density |
| $V$ | volume |
| $k$ | stiffness |
| $E$ | Young's modulus |
| $A$ | area |
| $L$ | length |
| $\epsilon$ | strain |
| $\sigma$ | stress |
| $t$ | time |
| $i$ | integer |
| $d$ | deviation |
| $f$ | frequency |

## Acronyms

| | |
|---|---|
| **5G** | Fifth Generation (Wireless Communication Technology) |
| **CAD** | Computer-Aided Design |
| **CSP** | Concentrated Solar Power |
| **DLF** | Dynamic Load Factor |
| **DLR** | German Aerospace Center |

**DOF** Degree of Freedom

**FEA** Finite Element Analysis

**FEM** Finite Element Method

**FFT** Fast Fourier Transform

**MSUP** Mode Superposition

**PSD** Power Spectral Density

**PV** Photovoltaics

**ROM** Reduced Order Model

**STFT** Short-Time Fourier Transform

CHAPTER 1

# Introduction

As the global economy undergoes a transformative shift away from fossil fuels and towards renewable energy sources, solar energy has emerged as the pivotal player. While Photovoltaics (PV) have long been the poster child for solar energy generation in a modern context, it's worth acknowledging its intrinsic limitations. Among other options, Concentrated Solar Power (CSP) stands out as a commercially viable solution. CSP operates by concentrating the sun's rays onto a small area called the receiver, thereby generating heat, which serves as a potent fuel for efficient electricity production and a host of other industrial processes. The technology's origins trace back far beyond the discovery of PV, with roots in ancient Greece and the first practical developments occurring in the late 19th and early 20th centuries [1].

There are four CSP technologies that all differ in the way the sun's rays are reflected and in the method of their receiver. Central receiver systems use mirrors tracked on two axis, known as heliostats, to project the sun's rays onto a central receiver, which is installed on a tower (see Figure 1.1). Modern CSP plants have thousands of heliostats, creating a radiation concentration so intense, the receiver unit can reach temperatures of over 1000 °C. The heliostat field makes up to 40%

of the total plant cost, making their cost-effectiveness very relevant [2].

Heliostats usually have mirrors with surface areas ranging from 25 $m^2$ all the way to over 150 $m^2$, but smaller implementations exists too, with mirror areas between 2 to 15 $m^2$. Those fall into a category of 'very small' heliostats. The cost-effectiveness of different mirror sizes is still debated, and previous studies have shown that as the mirror size approaches below 2 $m^2$ per heliostat, they become impractically expensive on a commercial scale [3]. Advancements in technology, such as the roll-out of 5G, which enables better communication with heliostats, more powerful and affordable electronics, and the globalization of supply chains, have reshaped the engineering landscape, making cost-effective small heliostats more feasible than ever before.

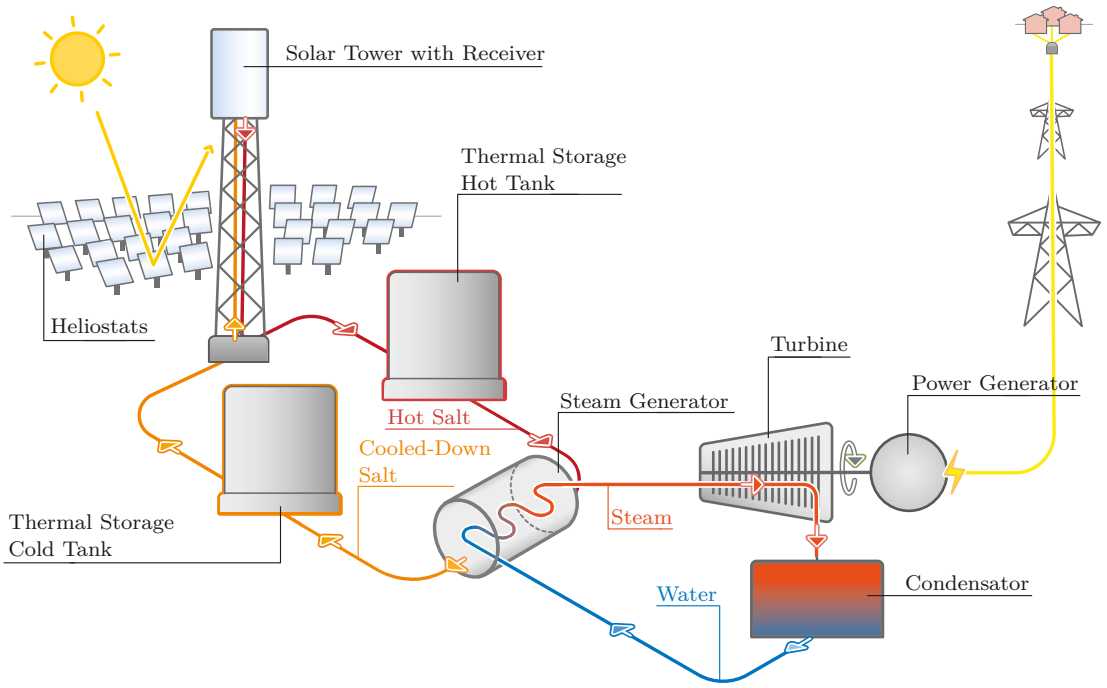



**Fig. 1.1** Central Receiver System. Image: DLR

At the German Aerospace Center's (DLR) institute of solar research, a new heliostat design is being developed with the potential to drive down costs per unit by up to 20% compared to the industry standard. This design distinguishes itself from others through its minimized complexity and small dimensions. The mirror has a surface area of just 2 $m^2$, and many of its components are off-the-shelf, further driving down cost due to the nature of economies of scale. Because of its small size, however, wind has a bigger influence due to the higher natural frequency of smaller

structures, which may more closely match the induced frequencies from gusts of wind.

Heliostats rely on their precision in order function properly. Having sun tracked components that focus sunlight on exact target locations, only minor inaccuracies can render them useless. Since heliostat are exposed to outdoor weather throughout their lifespan, it must be ensured that they don't break and remain accurate. That means ensuring they can survive the most critical wind loads they will encounter during their operational lifetime.

Rather than simulating the entire lifespan of the heliostats, a single dynamic structural simulation, which is considered long-duration within the context of numerical studies, though lasting only a few minutes, is performed using wind exposure data collected over several years. This data, processed with extreme value statistics to highlight the most critical wind conditions, is used to model the heliostat's response to these extreme conditions over the simulation period [4]. A long-duration simulation like this is valuable because it provides a more detailed understanding of the heliostat's performance under extreme conditions, which may not be fully captured by analyzing just the raw wind data.

CHAPTER 2

# Review of Related Work

In 2021, DLR, in collaboration with Wacker Ingenieure, released the final report on the Project MAHWIN. MAHWIN is an acronym for: *Erforschung einer Methode für effiziente und sichere Auslegung von kostengünstigen Heliostaten durch Vermessung und Modellierung dynamischer, nicht-linearer Windlasten*, which translates to: "Researching a method for the efficient and safe design of cost-effective heliostats through measurement and modeling dynamic, non-linear wind loads". This thesis goes into themes similar to those found in the report, particularly regarding the dynamic wind loads applied onto a heliostat. However, there were some issues encountered. The solutions derived from the FEM model didn't converge, likely due to the complexity of the model. However, attempts to replicate the behavior of the full-scale measurements using a computationally efficient mass-spring-damper model resulted in good agreement. Therefore, project MAHWIN can partly serve as a reference for assessing the outcomes of this thesis, comparing the similarities and differences in results. However, the compatibility is limited since the heliostat of MAHWIN was of different dimensions.

CHAPTER 3

# Objective

A map depicting a large area only shows its most basic geographical features, while a map that spans a much smaller region is rich in detail. This analogy can be applied to the two types of models explored in this thesis: one with low resolution, meant for long-duration analysis, and another with high resolution, meant for short-duration analysis. Heliostats experience changing weather conditions throughout their lifetime, which are highly heterogeneous, especially in regards to wind patterns. Two types of heliostat models with different scales and resolutions can be used in tandem to determine critical structural responses. The high-resolution model, intended for short-duration analyses, provides precise results for validating the simplified models and conducting reliable stress analyses. The low-resolution (simplified), long-duration model is used to simulate extended periods and identify critical timestamps where significant structural responses occur.

This thesis aims to develop a system capable of conducting long-duration simulations of DLR's heliostat under applied wind load datasets. In this context, long-duration refers to simulations lasting several minutes, which, while brief compared to the heliostat's entire lifespan, are relatively long in the realm of FEM analysis.

This approach helps to identify worst-case conditions from the wind load dataset, which can then be applied in short-duration studies. Such an analysis is essential because standalone wind load datasets do not always accurately reflect the structural response or pinpoint the most critical timestamps for the heliostat.

Applying the most critical dynamic loads will result in the most critical deformations and stresses. Since inertial forces are introduced for structures in transient analysis, these deformations and stresses can be larger than those from an equivalent static analysis with an equal but constant peak load. This difference is used to calculate the Dynamic Load Factor (DLF), a factor that expresses the ratio of dynamic to static load responses.

To meet DLR's requirements, it's necessary to have a benchmark model of the heliostat in order to validate any other model. This is done by building two separate FEA models using different techniques and seeing how they compare. If their deviation in load tests meets common FEA criteria, it can be assumed that their results are reliable, and a benchmark is obtained. It is only then that it becomes appropriate to experiment with simplifications intended for long-duration analysis. Although this validation process is robust, true validation requires real-life experimentation, which is beyond the scope of this thesis. Therefore, comparing two separate numerical models provides a practical, though not perfect, approach to ensure accuracy.

Usually, transient simulations are intended for short durations, reflecting quick events. A long-duration study will have significant repercussions on computational resources, so efficiency needs to be prioritized. While Ansys is the primary tool selected for this thesis, alternatives are still considered. The modern market offers a variety of tools intended for handling these types of simulations. In Matlab's Simulink, the *Simscape Multibody* extension offers a wide range of capabilities for modeling and simulating mechanical models. The connecting of separate components with joints combined with a graphical modeling environment makes it a strong alternative.

Nevertheless, no simulation tool is inherently faster than any other. Exhaustive model simplification and optimization need to be carried out in order to achieve a short computation time. This involves making DLR's heliostat geometry as minimal as possible, minimizing the modes down to only include the most significant ones, and maximizing the efficiency of the compilers and solvers.

The final step involves calculating stresses at critical timestamps with a complex model. The interval in the dataset used for the long-duration study, where the loads induce the most critical responses, are reapplied to the complex model that was originally used to validate the full-duration models. This way, detailed insights are provided into the types and magnitudes of stresses that the heliostat endures.

*CHAPTER* 4

# Methodology & Modeling

This chapter outlines the methodology and modeling process of the thesis, following the sequence shown in Figure 4.1. The main steps include: developing both complex and simplified models, validating the simplified models, conducting a full-duration study with one of them, and finally, calculating critical stresses using the complex model. This approach effectively meets the requirements set by DLR for the heliostat.

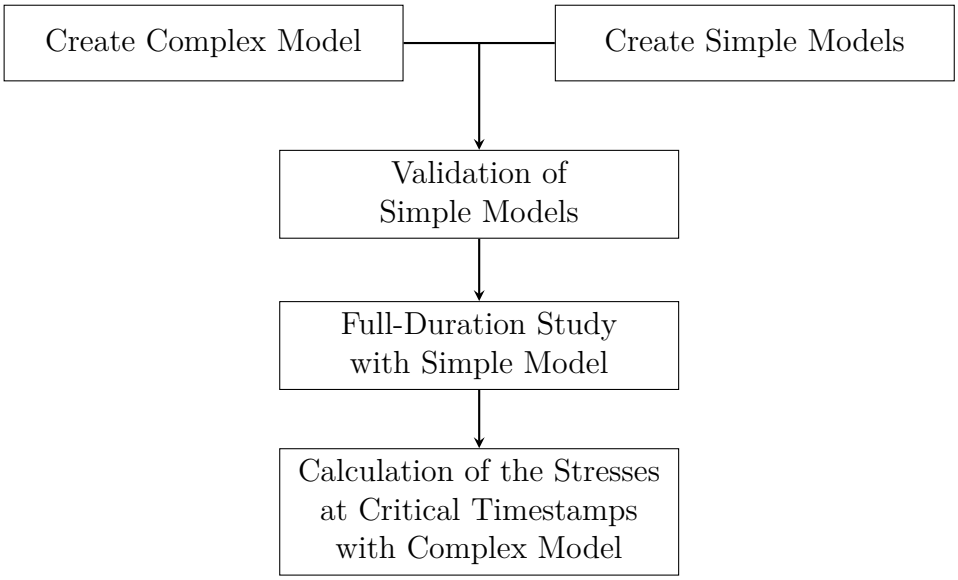



**Fig. 4.1** Flowchart of the Thesis

## 4.1 Geometry

DLR has developed and deployed numerous heliostats since the Institute of Solar Research was established in 2011 [5]. The latest heliostat distinguishes itself from others due to the size of its solar glass mirror, which is on the lower bound in terms of its area compared to the industry average. DLR's heliostat is also unique for its simplicity, and its structural components can be broken down into the following list:

- solar glass mirror
- carrier frame
- T-mount
- pylon

Figure 4.2 shows an exploded view of the heliostat where these components are labeled.

The structure is composed of two materials; mirrored glass and structural steel. A layer of silicone adhesive and sealant, namely Sikasil-AS-785 connects the solar glass mirror to the carrier frame below [6]. The model in its entirety is very thin-walled, with the steel structure of the carrier frame being 3 mm thick at most, and the pylon having a thickness of 0.6 mm, but the solar glass mirror is 4 mm thick. An azimuth and elevation gear drive the heliostat's two axes, enabling spherical rotation for its intended purpose. As they are hard to model and don't serve a

direct structural advantage, they are replaced with simple point masses. This way, their inertial forces are retained in simulations.

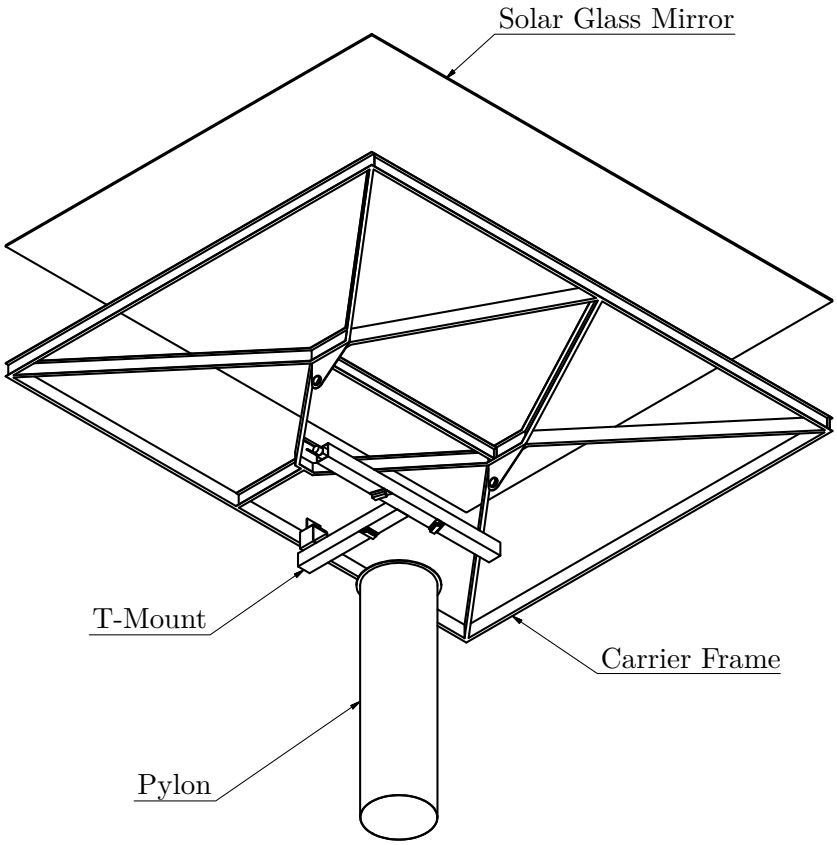



**Fig. 4.2** Heliostat Exploded View

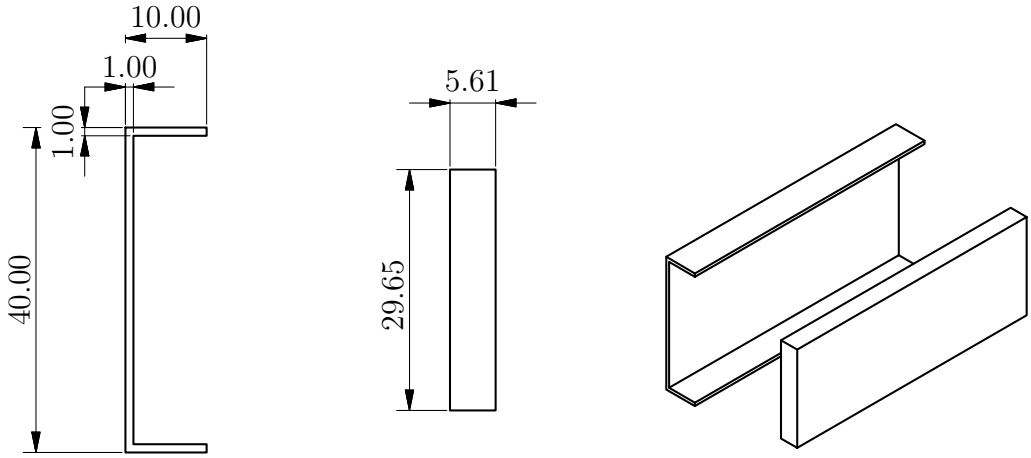
### 4.1.1 Original C-Profiles

The heliostat's carrier frame consists of interconnected beams, which were selected from the start to have C-profiles. C-profiles are advantageous as they are cheap to manufacture, light, and provide good structural rigidity. The deformation energy is absorbed through the warping of the C-shape in its lower portion, which in turn reduces the bending of the solar glass mirror that they support.

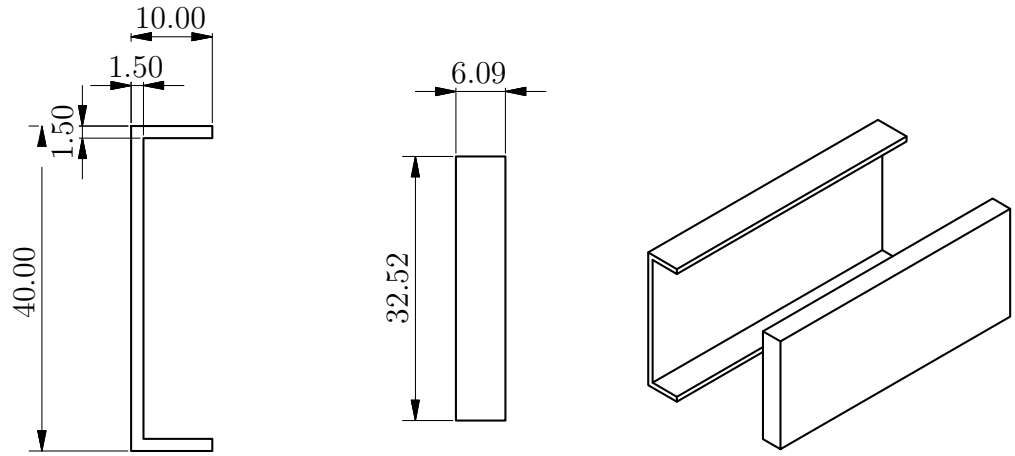
### 4.1.2 Rectangular Profiles

The heliostat was radically simplified for the Simscape Multibody setup (Section 4.4.1), which mainly involved changing the profile of the carrier frame. Measures were taken in order to have these new profiles reflect the original C-profiles as accurately as possible. Using traditional beam theory, the C-profiles were approximated as rectangular profiles while maintaining the same second moment of inertia about

both axes, calculated using Listing A.1 from the appendix. Figure 4.3 shows the C-profiles with their corresponding rectangular profiles for both a wall thickness of $t = 1.0$ mm as well as $t = 1.5$ mm.




**(a)** Main Profiles; Original C-profile and equivalent rectangular profile.




**(b)** Stiffened Profiles; C-profile with 1.5 mm wall thickness, followed by a larger rectangular profile derived from it.

**Fig. 4.3** Beam Profiles in Carrier Frame (Dimensions in mm)

## 4.2 Wind Tunnel Data

A large dataset was created in a previous study, where a heliostat was placed in a wind tunnel, and the resulting loads recorded on 24 sensors spread across the top

of the solar glass mirror over a span of five and a half minutes. In bad weather, heliostats are moved into safety positions, meaning that the concentrator aligns itself to be parallel to the ground surface. This way, the surface of the heliostat poses least drag as most of the air passes freely over it, which in turn minimizes the load. In this same way, the wind tunnel test was conducted, and the sharpest loads were captured close to the concentrator's windward side.

Considerations regarding placement of sensors on other regions of the heliostat might arise, such as on the pylon or carrier frame. Although these implementations would increase the realism of the simulation, their effects can be considered negligible compared to those on the solar glass mirror. While the overall drag on the solar glass mirror is relatively low, the fluctuating vertical component of the vortices creates a significant moment about the axis perpendicular to the elevation axis.

### 4.2.1 Data Implementation

The wind tunnel data is originally saved as a Comma-Separated Value (**.csv**) file, which had to be restructured and converted to either an extensible markup language (**.xml**) file for Ansys or a Binary Matlab (**.mat**) file for Simscape Multibody. The unit of the original datapoints is in Pascals (Pa), but because Simscape Multibody can only process force, the data had to be converted to force. The code listings used to process the wind tunnel data for this thesis are provided in Appendix A, specifically in Listings A.2 and A.3.

The data were recorded with the wind's direction of travel along the length of the heliostat. For all simulations in this thesis, the windward side is selected as its east side, relative to the T-mount's orientation, as seen in Figure 4.4.

### 4.2.2 Peak Moment

For initial tests, an approximate peak load is determined based on the wind tunnel data. The net moment is calculated by summing the moments generated by each individual load spread across the load tiles. Doing this for the entire load data timeseries will result in the maximum net moment in this 2D representation, which is assumed to correspond to the maximum peak load. The full-duration analysis will confirm or refute that assumption. Because the dataset displays the loads as pressure, they have to be converted into force by multiplying them with their corresponding area. Although the area of each load tile varies slightly, the symmetry of

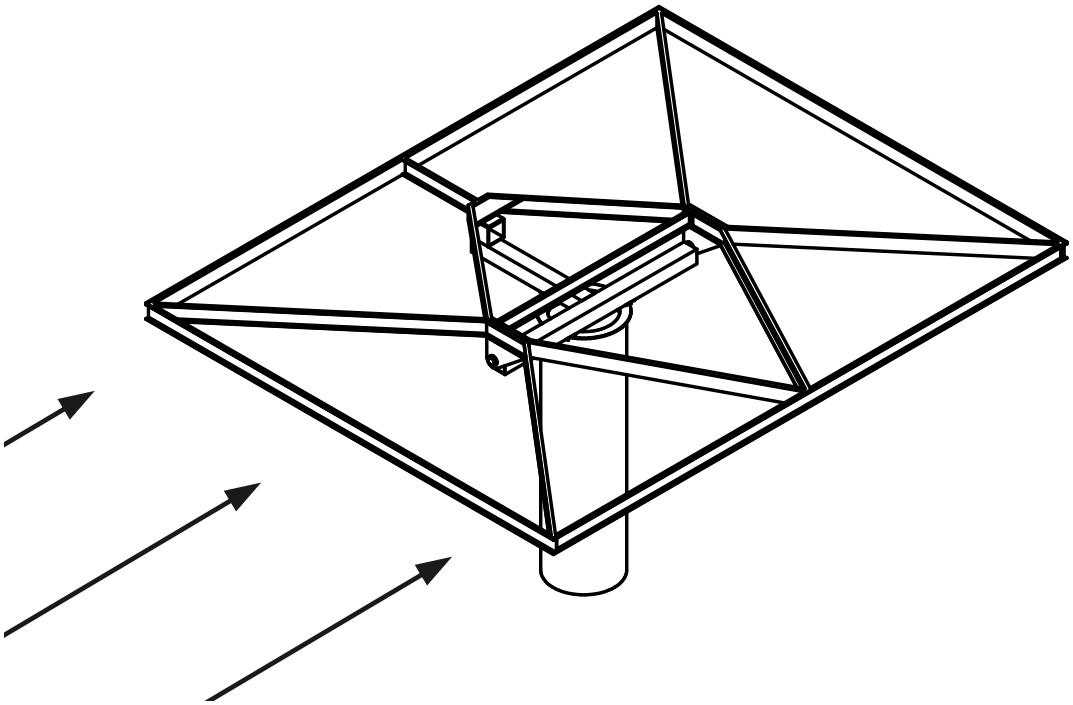


**Fig. 4.4** Windward side of Heliostat

the heliostats is exploited to obtain them as shown in Algorithm 1. The distribution of these load tiles is visualized in Figure 4.5.

**Algorithm 1** Calculate Load Tiles Area

**Function** `calculateArea`($x$, $y$, $ix$)**:**
- Set $x$ as an array [0.2279, 0.2872, 0.2872, 0.2872, 0.2872, 0.2279]
- Set $y$ as an array [0.2656, 0.3719, 0.3719, 0.2656]
- Initialize $ix[1]$ as an empty list
- **for** *index i from 0 to 23* **do**
  - Calculate area using:
    - $x\_coordinate \leftarrow x[\lfloor i/4 \rfloor]$
    - $y\_coordinate \leftarrow y[i\%4]$
    - $area \leftarrow x\_coordinate \times y\_coordinate$
  - Append *area* to $ix[1]$ list

## 4.3 Complex Models

Two complex models are made for DLR's Heliostat: one for another thesis (Section 4.3.3) and one for this thesis. The methods used to create these models are vastly different, as the prior is mainly modeled using shell elements whilst the latter with only solid elements. The solids element model is completely conformal along the solar glass mirror and carrier frame, and the element size is adaptive, changing

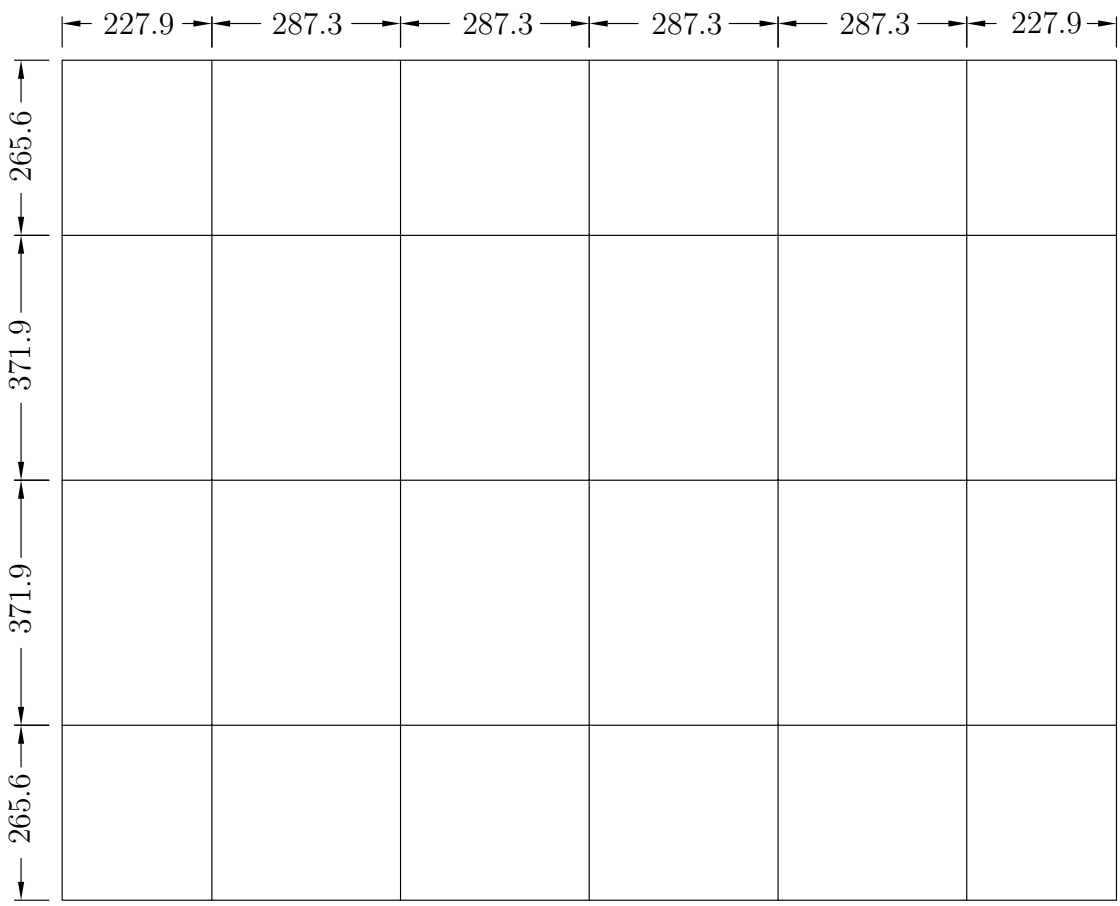



**Fig. 4.5** Solar Glass Load Tiles (Dimensions in mm)

based on the geometry's complexity. The shell element model relies only on defined contact-regions to transfer the forces experienced by the model.

### 4.3.1 Solid Element Model

The main complex FEM model of this thesis is entirely made from solid elements, most of them being quadratic hexahedral elements. Hexahedral elements have several advantages, the most significant one being that they achieve a higher accuracy for a lower element count, hence, they are more efficient in regards to computational effort. When stress gradients are not large, quadratic elements can also be sufficient for the model to only have a single layer of elements [7].

### 4.3.2 Meshing of the Solid Element Model

Due to solid elements potentially losing some accuracy in thin-walled structures as their aspect-ratio tends to be higher, the mesh was made adaptive in regions of higher complexity and criticality. Furthermore, to maximize the mesh quality, the elements were made conform throughout the solar glass mirror and carrier frame.

Achieving a fully conformal mesh like this is a tedious task and requires accurate parameters to describe the geometry for the modeling tool. *DesignModeler*, a Computer-Aided Design (CAD) program included with Ansys, allows the geometry to be sliced down to the most primitive shapes, helping the mesh generator to align elements together.

All faces of each shape should align perfectly with every other face. This is a relatively easy task for simple geometries, but the difficulty increases significantly as the geometries become more complex. Because the shapes that the geometry is sliced into are all infinitesimally close to each other, Ansys can auto-detect the contact regions with a tolerance setting that encapsulates the separation, which is effectively zero. Figure 4.6 shows an isometric view of the final mesh of the solid element model. Figure 4.7 provides a close-up of the same mesh from below, showing the region where two beams intersect with beam running along the solar glass perimeter (referred to as frame 8 later).

### 4.3.3 Shell Element Model

The shell element model, originally developed by Justus Blum [8], is implemented in this thesis and enhanced to serve as one of the two simplified models of this thesis

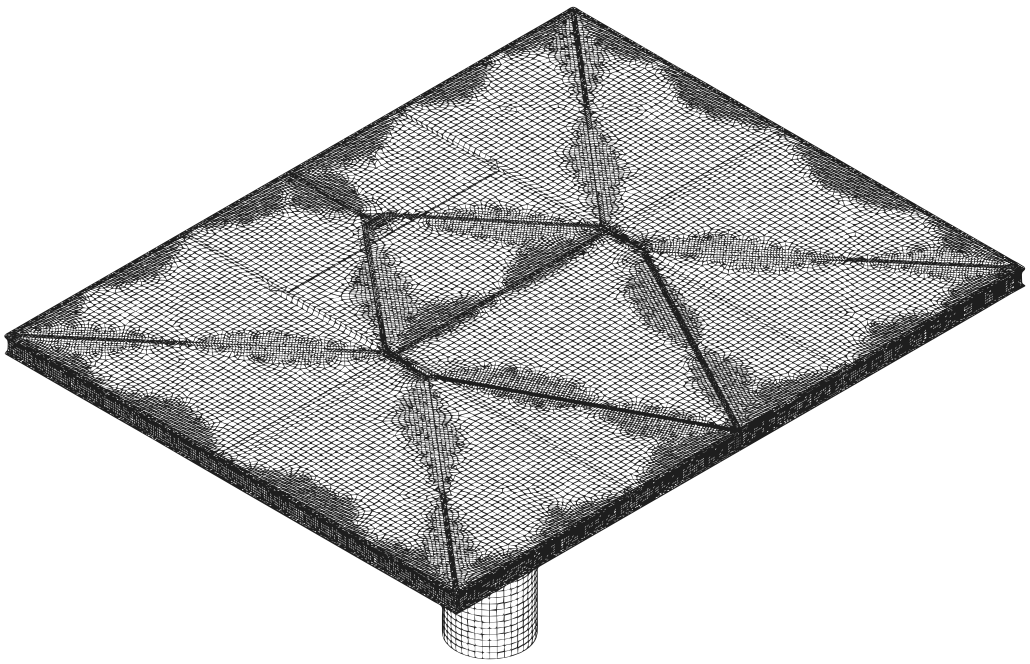


**Fig. 4.6** Full View of Meshed Heliostat

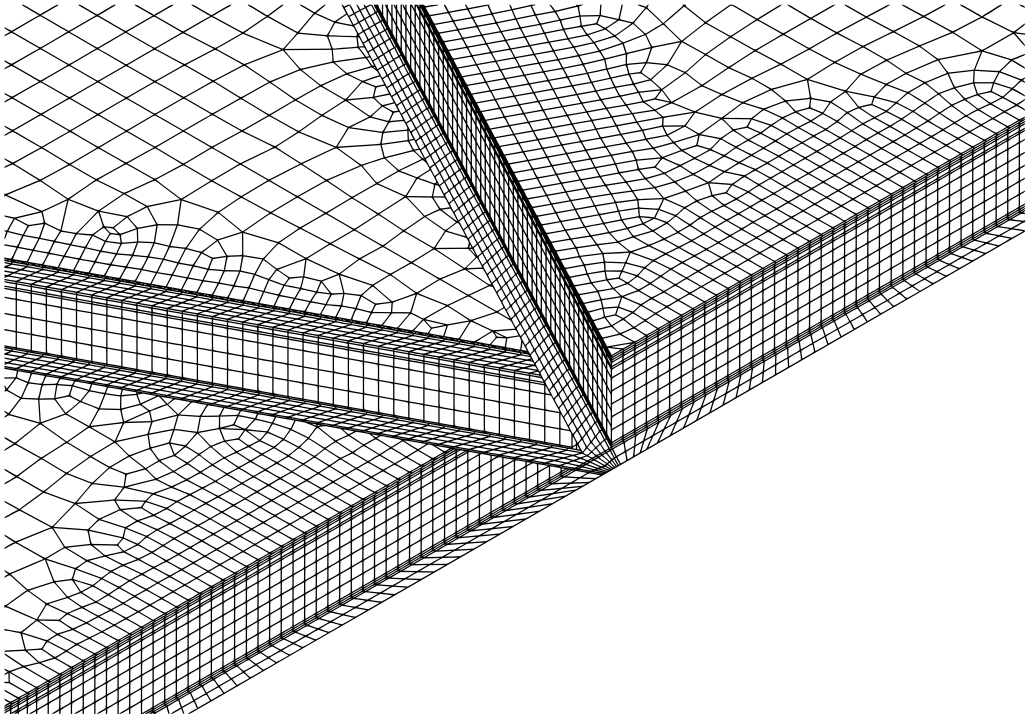


**Fig. 4.7** Close-up View of Meshed Heliostat

(see Figure 4.8). This is a radically different approach which comes with its own benefits and drawbacks. Shell elements can produce very accurate results relative to the element count when used for thin-walled structures, and are much easier to mesh compared to solid hexahedral elements. They also provide excellent optimization opportunities, as they essentially exist as 2D planes with their thickness defined by a single parameter.

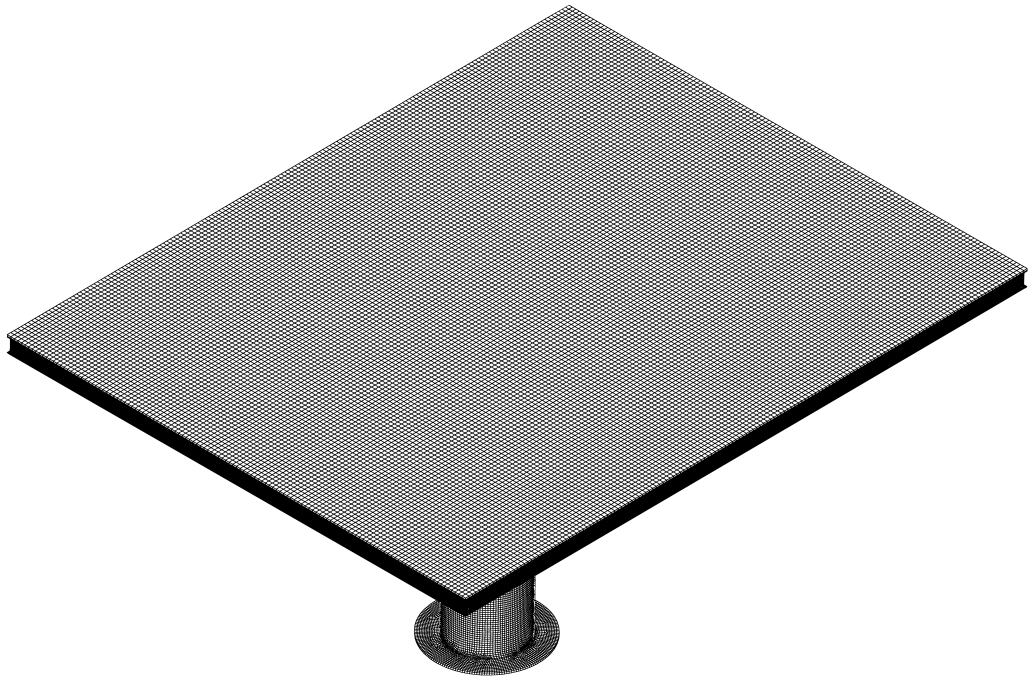


**Fig. 4.8** Full View of Meshed Shell Element Heliostat

Issues may present themselves though. By changing the wall thickness of shell elements, one needs to make sure that all original dependencies are maintained, which can be difficult for complex models. Shell elements rely on predefined contact regions, which can be disrupted if the model parameters are changed. Additionally, because of the 2D nature of shell elements, these contact regions could potentially lead to oversimplification, resulting in approximation issues for the model.

## 4.4 Simple Models

Two methods for simplifying the complex models are explored for the full-duration study. This section describes the making of those simplified models.

For many structural problems, it is safe to assume that the deformation experienced by a model is only linear. Although it's wise to take non-linear deformations

into account for certain simulations involving heliostats, for a full-duration study, this becomes impractical, given the large data-set.

### 4.4.1 Multibody Model

In the following section, the Multibody model developed in Simulink is described in three main steps in detail. The simplified model, as shown in Figure 4.9, is aimed at handling the extensive data involved in the full-duration study.

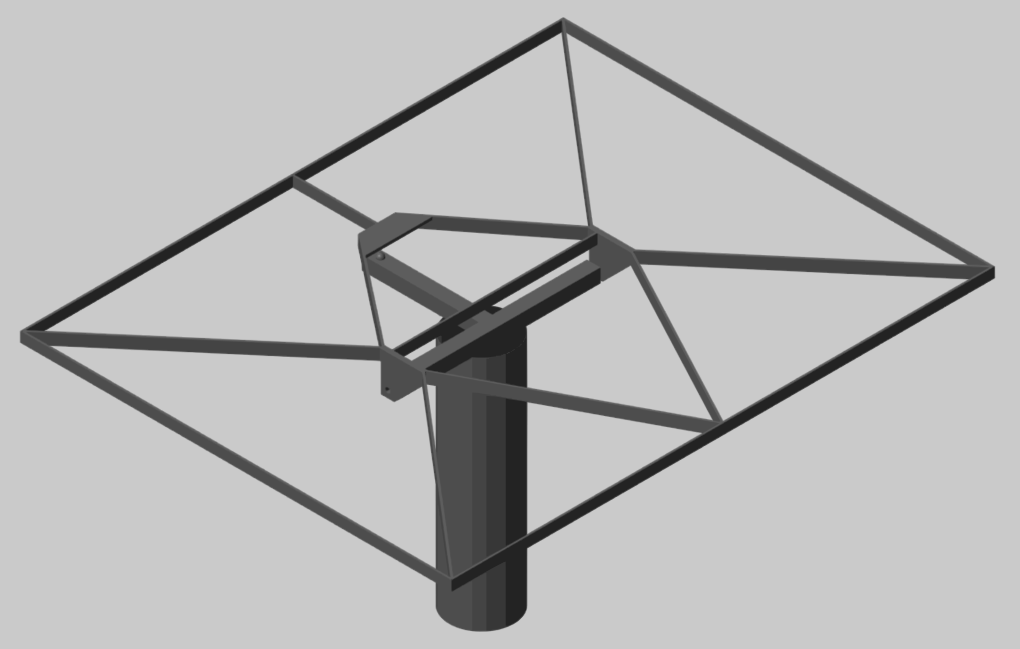


**Fig. 4.9** Full View of the Multibody Model in Simscape's Mechanics Explorer; The solar glass mirror was removed for clarity.

#### 4.4.1.1 Reduced Order Model

The *Flexible Body Model Builder* tool allows for the creation of models that are compatible with the *Reduced Order Flexible Solid* block, both of which are components found in Simulink's *Simscape Multibody*. The tool enables the setting of material properties, discretization parameters, number of modes to be extracted and the reference frames, which serve as connection points to join components.

The mesh consists only of quadratic hexahedral elements. Using the Craig-Bampton model order reduction method, the *Flexible Body Model Builder* tool generates a **.mat** file, consisting of a generalized stiffness and mass matrices representative of the extracted dominant modes, the origins and orientations of the interface frames, and the graphics matrices for representing the deformed geometry of the

flexible bodies [9]. Unfortunately, the tool does not give insights into the shape of the generated mesh and in its documentation it is even stated that the generator can go beyond the limits of the discretization parameters [10].

The solar glass mirror was generated without much issue, and the pillar was effectively represented with a simple flexible beam element, already built into Simscape Multibody. The carrier frame posed issues however, and required extensive reiterations in the *Flexible Body Model Builder* tool. These iterations were made in order to generate Reduced Order Model (ROM) data at all, as the generator would produce unknown errors for most adjustments. Therefore, simplifications had to be done. One such simplification involved going from the original C-profile of the beam to a basic solid beam with a rectangular cross section, adjusting the dimensions and material properties to maintain the original stiffness and mass of the C-profile, as described in Section 4.1.2

If only the profile dimensions were changed, the new beam would experience very different moments, as the solid volume is too large for the density of structural steel to make sense. The density of the new beam is adjusted using the following equation:

$$\rho_{\text{simplified}} = \frac{V_{\text{original}}}{V_{\text{simplified}}} \cdot \rho_{\text{steel}} \tag{4.1}$$

Where $V_{\text{original}}$ is the original volume, $V_{\text{simplified}}$ is the volume of the simplified beam, and $\rho_{\text{steel}}$ is the density of structural steel. $\rho_{\text{simplified}}$ is the resulting density which is derived from the change in volume between the original and simplified models. Inventor provides the volumes of both models, and using equation 4.1, the adjusted density is calculated as:

$$\rho_{\text{simplified}} = \frac{769510.240mm^3}{2269424.906mm^3} \cdot 7850kg/m^3 \approx 2661.76kg/m^3$$

The T-mount's wall thickness remains the same as from the original model, but is simplified by removing the motors and gearing from the elevation and azimuth joints respectively. The resulting missing mass is replaced with point masses (section 4.4.1.2).

Although ROM data was ultimately generated for all heliostat components, the process pushed the limits of *Flexible Body Model Builder*'s capabilities. Many errors occurred without the tool providing further explanation. Matlab's forum commu-

nity suggested the issues lie in the tool's limits in handling complex geometry. For this thesis this is only partially true, as there are two additional factors that disrupted the process. It was found out that the *Flexible Body Model Builder* handles most CAD formats poorly. The Jupiter Tessellation (**.jt**) format did best, although not free from errors, as a too complex geometry would import with missing faces and therefore hollow inside. More confusion is caused by the unpredictability of the *Partial Differential Equation Toolbox* and Linux-dependent *Integro-Differential Modeling Frameworks*, both options provided by the *Flexible Body Model Builder* to generate the ROM data. Some geometries were compatible only with one option, while others were compatible only with the other.

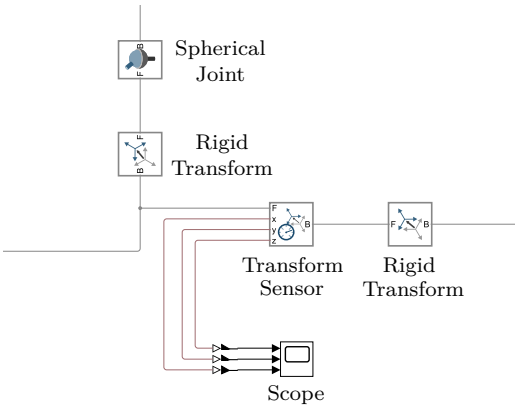
#### 4.4.1.2 Simscape Multibody

*Simscape Multibody* is an extension to the Simulink environment, where multibody systems can by modelled with various blocks. The advantage of Simscape Multibody is that it offers a wide range of joints with many configuration parameters, enabling reconstruction of the heliostat in a way that reflects its real-life mechanisms.

Joining the components has to be done with care, as joints can only be applied between frames that are perfectly aligned. As the component's reference frames can't be modelled with such perfect alignment, manual adjustment has to be carried out. *Transform Sensor* blocks measure the coordinates of two reference frames, and a new virtual reference frame is added that cancels out the positional difference. Here, Matlab's format needs to be set to `long` in order to retain the necessary precision for the *Rigid Transform* blocks that are used. Figure 4.10 shows, on the left, the slight misalignment in the connection between the carrier frame and the solar glass mirror, and on the right, how this misalignment is corrected using the blocks mentioned (example for a different connection).

The assembly is modelled from top to bottom, starting with the solar glass mirror and mounting it to the carrier frame below. The mounting points consist of 8 inserts along the component edges, and a single connection in the center. The center connection was modelled with a spherical joint [11], whereas the joints along the edges were rectangular joints, where only the x- and y-transformation weren't constrained [12]. To capture the elastic behavior of the glue that bonds the solar glass mirror to the carrier frame, the rectangular joints were given an internal stiffness property defined by Equation 4.2:

**(a)** Close-Up of Heliostat Perimeter; Two misaligned reference frames connect the carrier frame to the solar glass mirror.




**(b)** Transform Sensor, Rigid Transform, Spherical Joint and Scope Blocks; These blocks don't represent the connection in Figure 4.10a.

**Fig. 4.10** Illustration of How Blocks with Misaligned Frames Can Be Connected

$$k = \frac{E \cdot A}{L} \tag{4.2}$$

Where the Young's Modulus is defined with Equation 4.3:

$$E = \frac{\sigma}{\epsilon} \tag{4.3}$$

According to the datasheet of the Sikasil-AS-785 glue mentioned in Section 4.1, $\sigma = 2$ MPa and $\epsilon = 250\ \% = 2.5$, resulting in $E = 0.8$ MPa. From the model, the cross sectional area of the inserts is $A = 64.952\ \text{mm}^2$ and the average distance between them is $L = 0.72$ m, resulting in:

$$k = \frac{0.8\ \text{MPa} \cdot 65.952\ \text{mm}^2}{0.72\ \text{m}} \approx 72.17\ \text{N/m}$$

Following the connection of the solar glass mirror to the carrier frame, the T-mount is added. This component uses only rotational joints to accurately represent the real heliostat. The joints are labeled in Section 4.14. Their placement constrain any movement from happening, resulting in a locked structure. Finally, the pylon was added below the T-mount along with two point masses representing the worm gear and the elevation gear. Their corresponding masses are 0.2943 kg and 0.373 kg. The pylon was modelled with a *hollow cylinder* block, where its wall thickness was set to 0.6 mm.

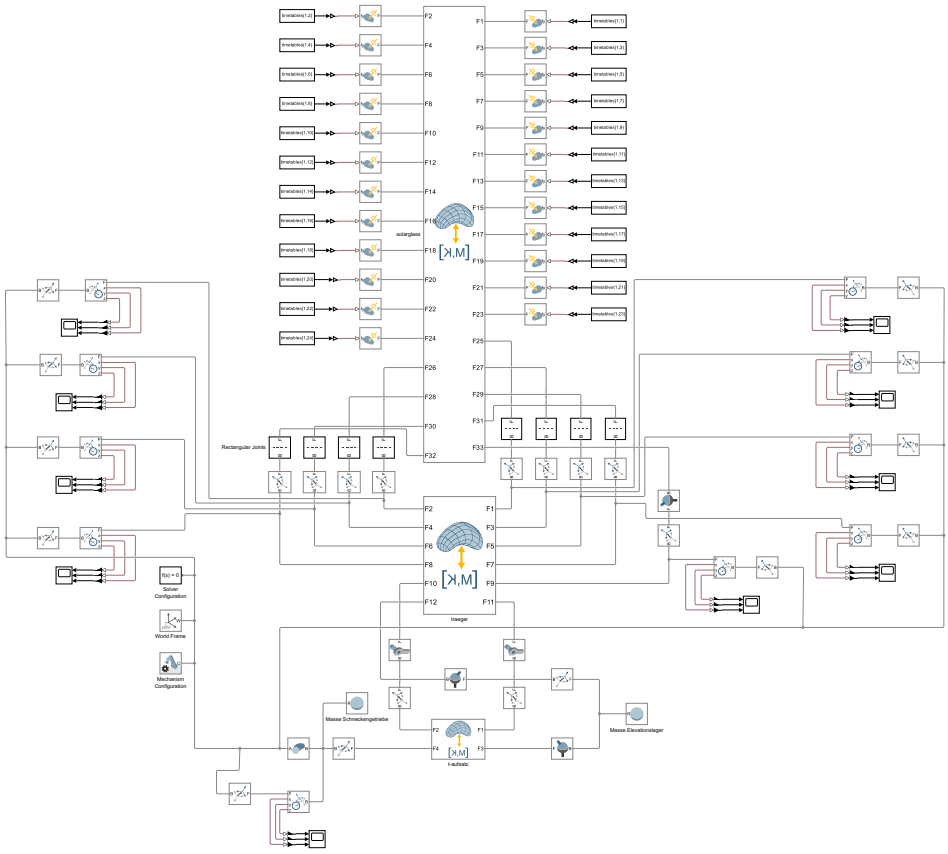


**Fig. 4.11** System of the Simple Mutlibody Model

Finally, 24 loads in total were applied onto the solar glass mirror. Simulink imports them from a **.mat** file as described in Section 4.2.1. The final system is shown in Figure 4.11.

#### 4.4.1.3 Simulation Setup

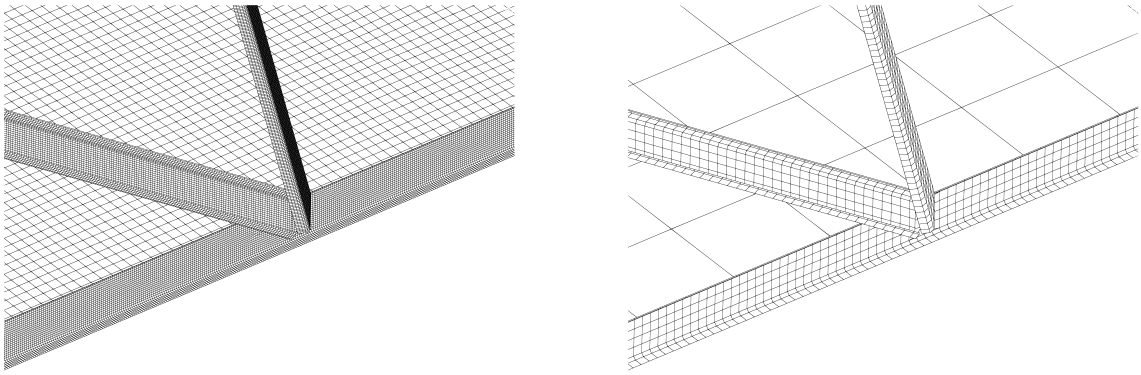
Simulink has many different solvers, each highly customizable. Step-size and tolerance settings are tuned to allow relatively high simulation errors, as for a long simulation, as long as each step converges, it is safe to assume that produced data has sufficient reliability for a simulation of this type. This is because a converging step suggests that the solution is reasonably accurate at that point. In Appendix B, different configurations are tested to find the most optimal simulation setting. The biggest improvement in simulation speed is achieved by reducing the amount of retained modes in the model, speeding up the execution time by up to 40× (refer to Appendix B.3). This result aligns well with the documentation of the *Reduced Order Flexible Solid* block [13].

### 4.4.2 Simplified Shell Element Model

The shell model from Section 4.3.3 has been enhanced to meet the memory constraints required for a long-duration study.

#### 4.4.2.1 Coarsening of Complex Shell Element Model

The shell model as described in Section 4.3.3 has 381,757 elements in total. This level of resolution makes transient simulations of long duration impractical. By changing the mesh parameters, the number of elements is reduced by an order of magnitude, down to 24,721 elements, as shown in Figure 4.12. As a result, this reduction leads to a two-orders-of-magnitude decrease in the unmodified element matrices, drastically reducing the required computational effort.



**(a)** Close-up View of Original Shell Element Model with 381,757 Elements

**(b)** Close-up View of Simplified Shell Element Model with 24,721 Elements

**Fig. 4.12** Comparison of Original and Simplified Shell Element Models with Different Meshing Parameters

However, such a radical simplification may also have a radical impact on the accuracy of the model's response. With proper validation, that risk can be avoided, and accuracy issues addressed with further optimization on the model parameters should they arise.

#### 4.4.2.2 Solver Optimization

The Ansys solver is optimized by setting all applicable options under `Output Controls` to `No`. This way, Ansys would only account for deformations and skip other calculations that aren't needed. Furthermore, to prevent Ansys from crashing when viewing the results of the measured datapoints (frames), the entire geometry is hidden in the graphics viewer.

Not all of the 30 modes will be retained as in the original shell element model from Section 4.3.3. Based on the participation factors of the model's modes, only the most dominant will be retained for the full-duration study, details of which are described in section 5.1, discussing the results from modal analysis of the complex models.

## 4.5 Benchmarking & Validation

Two complex models are used in this thesis specifically to create a reference benchmark. This means that both complex models are put through static and dynamic tests, with the goal for them to respond similarly. Such results don't inherently confirm that their responses are accurate to an equal real-world scenario, which is difficult given the complexity of the models, but it does increase the likelihood of them having such accuracy significantly.

Given that the complex models' responses align well with each other, the solid element model is used as a reference for the validation for the simplified models. Although most commonly, validation procedures in FEA involve real-world testing, a benchmark is a reliable enough metric for the purpose of this thesis and without going beyond its scope. The following sections describe how static and dynamic tests are carried out and apply both for obtaining a benchmark as well as validating the simplified models.

### 4.5.1 Static Approach

The peak loads that are determined in 4.2.2 are applied as a static load on the full-scale models, and the deformation along the vertical axis is compared at 8 different locations distributed evenly across the edges of the concentrator (Figure 4.13), as well as the deformations where the T-mount connects to the carrier frame (Figure 4.14). By assessing these components, issues are more easily identifiable should they arise, as reaction loads from the applied load move from bottom up. I.e., if the carrier frame doesn't deflect as expected but the T-mount does, it can be assumed that the issue lies in the carrier frame.

### 4.5.2 Dynamic Approach

A transient load response test is carried out by applying a 0.2-second interval from the wind tunnel data, centered on the peak load determined in Section 4.2.2. For

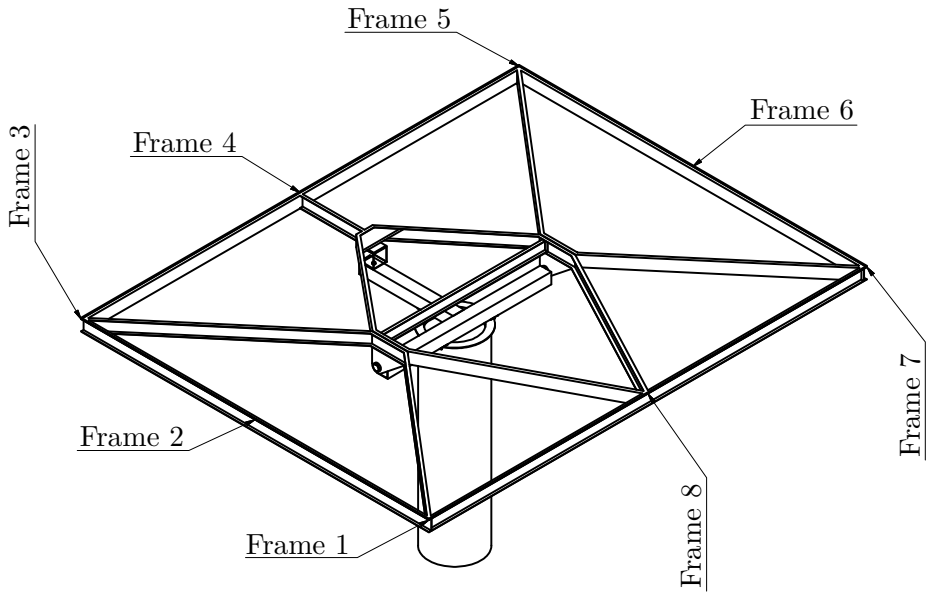



**Fig. 4.13** Static Frames on Carrier Frame

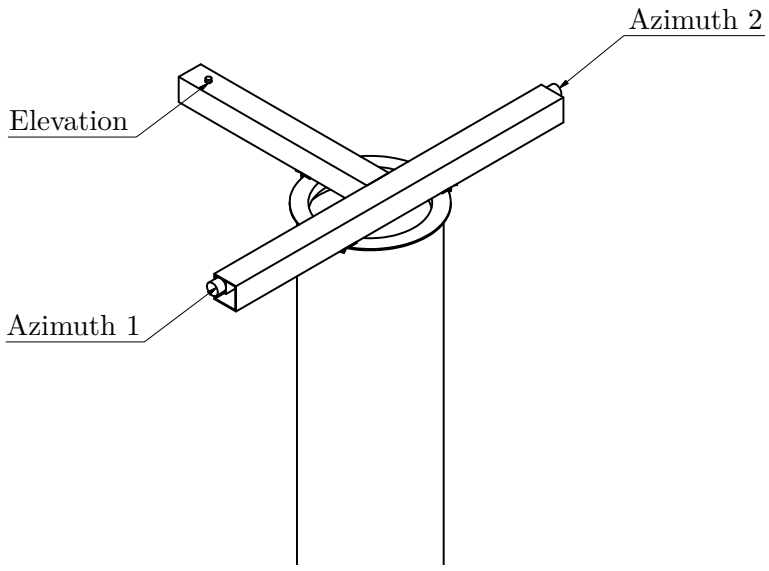



**Fig. 4.14** Static Frames on T-Mount

this, four locations are chosen to measure the deformation along the vertical axis as shown in Figure 4.15. This way, the regions subject to most deformation are taken into account while extracting a manageable amount of data.

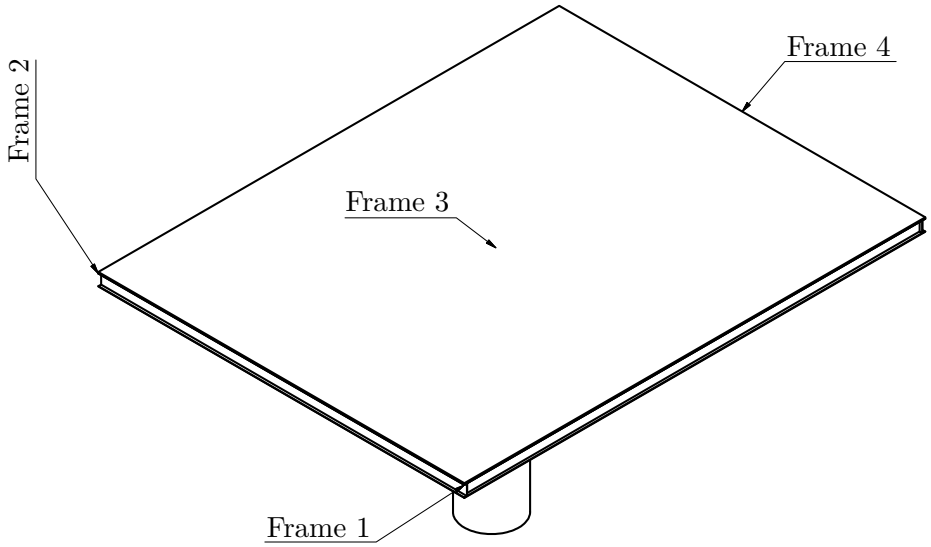



**Fig. 4.15** Dynamic Frames

## 4.6 Full-Duration Study

Based on the performance of the two simplified models, the better one is selected for the full-duration study. Deformations at the same frames are measured as in Figure 4.15, returning a time-series dataset. This way, multiple outputs can be analyzed without overloading the memory of the workstation used to carry out the simulation, as can happen when the entire deformed geometry is considered.

The workstation provided by DLR is equipped with a powerful Intel Xeon Platinum 8260 Processor and 256 GB of RAM, making it ideal for large simulations with numerous parameters.

Signal processing methods are used to analyze the resulting time series from

the recorded frames, as the raw output on a graph resembles a static, white noise signal. Extracting various signal components gives better insight into the frequencies at which the Heliostat, under the applied load, responds and the power of those frequencies throughout the course of the simulation, helping to identify the most critical timestamps.

## 4.7 Stresses at Critical Timestamps

Based on the results from the full-duration study, the most critical timestamps can be extracted. These are used to reevaluate short-term loads that can then be applied onto the original complex heliostat model from Section 4.3.1, enabling a more focused analysis.

According to the average dominant frequencies, the length of the applied load interval can be determined based on the number of periods to be considered. Equation 4.4 is used to determine the length of a short dynamic stress analysis.

$$t_{\text{interval}} = i_{\text{periods}} \cdot \frac{1}{f_{\text{average}}} \tag{4.4}$$

A duration considering two to three periods is sufficient to capture the peak load that is observed in the full-duration study. The critical timestamp is contained in the intervals' center, just like in Section 4.5.2.

In the post-processing, the equivalent von Mises stress within this new time interval is determined. Additionally, other regions of interest are analyzed and recorded, though further interpretation is beyond the scope of this thesis. The main focus is using the maximum stress to calculate the Dynamic Load Factor (DLF), described in the following section.

## 4.8 Dynamic Load Factor

The DLF defines the relationship between static and dynamic simulations with a single value. The proportion by which the peak stress in a critical dynamic simulation compares to the peak stress in an equivalent static simulation is expressed using Equation 4.5.

$$\mathrm{DLF} = \frac{\sigma_{\max}\ (\text{Transient})}{\sigma_{\max}\ (\text{Static})} \tag{4.5}$$

An equivalent static load is selected to be the same as used for the previous static load tests, described in Section 4.5.1. Depending on the amount of critical timestamps extracted, the largest DLF is selected.

CHAPTER 5

# Results

The Results chapter provides a detailed analysis of the static and dynamic performance of all models described in Chapter 4. Based on the validation results, one simplified model is selected for use in the full-duration study, which a most critical timestamp is extracted from and analyzed on the solid element complex model.

## 5.1 Modal Analysis & Resonances

All dynamic simulations of this thesis rely on the Mode Superposition (MSUP) method, where the models' most dominant natural frequencies are used to express the response of the structure. For both complex models, 30 modes were extracted, which is more than is needed for most engineering scenarios. Nevertheless, in order to retain as much accuracy as possible, this amount was chosen and only applies for the complex models. These modes ranged from

- **Solid Element Complex Model:** $f_1 = 13.23$ Hz to $f_1 = 162.28$ Hz
- **Shell Element Complex Model:** $f_1 = 12.57$ Hz to $f_1 = 116.34$ Hz

The lower end of the frequencies is similar for both models, settling on a range of 12-14 Hz, while the upper range diverges somewhat. However, the natural frequency's participation factors hint at that modes beyond $f = 100$ Hz are almost completely neglected when superposed, with participation factors quantifying how much each mode contributes to the overall response at a given frequency.

These results also give insight into how the heliostat could respond to wind loads, as they induce frequencies and vibrations themselves, potentially leading to resonance effects. Winds seem to occur at frequencies approaching 12 Hz, but don't go beyond that, and most remain well below at an average of 5-6 Hz [14]. Still, ensuring that resonances do not manifest requires a full-duration study.

In parallel to the simplifications done in Section 4.4.2, where the meshing parameters were adjusted, resulting in an order of magnitude reduction in element count, the modes should also be reduced in order to speed up simulation time. Doing the same modal analysis on the simplified shell model results in a very similar range of frequencies as seen in the complex models, listed in table 5.1. The participation factors are expressed for each Degree of Freedom (DOF), their signs indicating whether or not they are moving in phase of the direction of interest.

| Frequency Component | Frequency [Hz] | Absolute Participation Factor |
|---|---|---|
| $f_1$ | 13.604 | 149.6 |
| $f_2$ | 13.85 | 177.2 |
| $f_3$ | 14.815 | 94.1 |
| $f_4$ | 20.752 | 81.8 |
| $f_5$ | 21.563 | 7.9 |
| $f_6$ | 27.729 | 133.5 |
| $f_7$ | 31.553 | 143.1 |
| $f_9$ | 51.611 | 7.8 |
| $f_{10}$ | 56.132 | 6.0 |
| $f_{14}$ | 81.426 | 8.3 |

**Tab. 5.1** 10 Most Dominant Frequencies

By taking the sum of the absolute value of each participation factor corresponding to all frequencies, those that are most dominant are revealed. For the simple shell model, the six highest absolute participation factors are all above 100, with the largest being p = 177.2. The next four following are bigger than p = 5, and the remainder is below. Combining the 10 most dominant modes in the MSUP transient

simulation reduces the computational effort of the full-duration study sufficiently while keeping enough modes for redundancy's sake.

## 5.2 Benchmarking Complex Simulation Models

As described in Section 4.5, before the simplified models can be validated, confidence needs to be built in the validity of the responses produced by the complex solid element model. Comparing it with the complex shell element model described in Section 4.3.3 proves that the model is credible for validating the simplified models, provided both models produce consistent results.

Every table in the following sections summarizing comparisons between the frames described in Sections 4.5.1 and 4.5.2 includes the deviations of the models' results, calculated using Equation 5.1

$$\text{Deviation: } d_i = \left| \frac{f_{\text{model 1}}}{f_{\text{model 2}}} - 1 \right| \times 100\% \tag{5.1}$$

as well as the corresponding weighted deviations, calculated using Equation 5.2:

$$\text{Weighted Deviation: } d_{wi} = d_i \times \frac{\max\left(\left|f_{\text{model 1}}\right|, \left|f_{\text{model 2}}\right|\right)}{f_{\max}} \tag{5.2}$$

The weighted deviation ensures that the results are normalized relative to the largest deformations observed in the models, providing a fair comparison.

### 5.2.1 Static Test

The complex solid element model from Section 4.3.1 is directly compared with the equivalent shell element model described in Section 4.4.2 in Table 5.2. Positive values indicate upward deformation along the vertical axis while negative values are downward.

Seeing the weighted deviation being mostly within 10% confirms that the complex solid element model from Section 4.3.1 can be used for static tests. The largest deformation reads d $\approx$ 7.2 mm for both models, which occurs in the heliostat corner at frame 1. The overall deformations remain similar along the heliostat's border, although some discrepancies appear on the side opposite of frames 1 to 3. The T-mounts' deviations are bigger than anticipated, but as they are all within 1 mm,

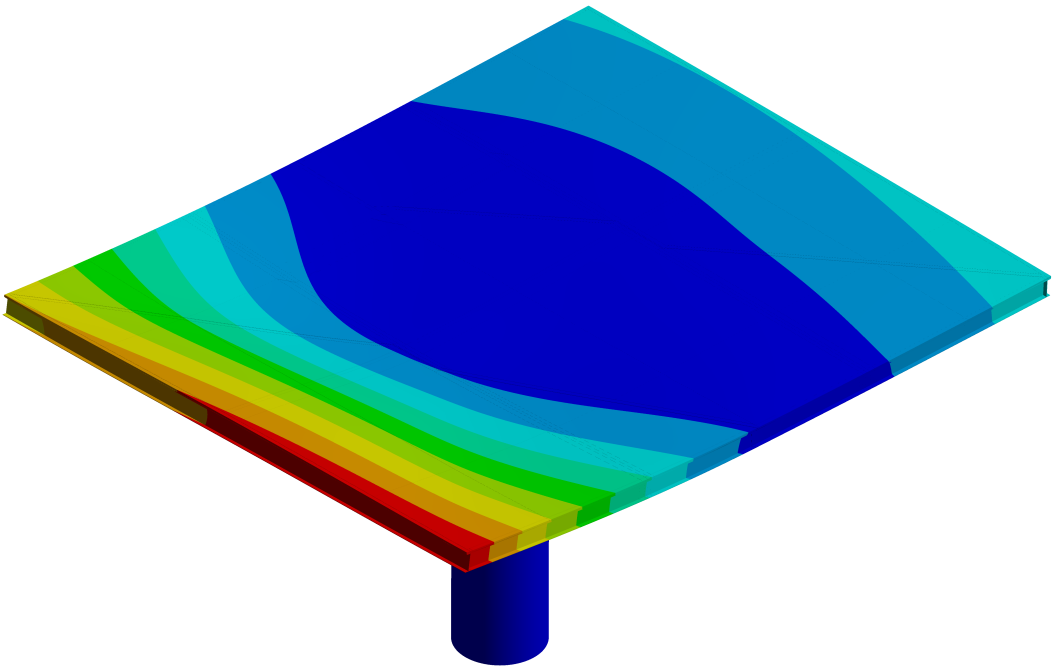


**Fig. 5.1** Deformation of Solid Element Model

| Component | Shell Model [mm] | Solid Model [mm] | Deviation (%) [%] | Weighted Deviation [%] |
|---|---|---|---|---|
| Frame 1 | 7.1838 | 7.2028 | 0 | 0 |
| Frame 2 | 6.8927 | 6.6609 | 3 | 3 |
| Frame 3 | 5.0891 | 5.6012 | 10 | 8 |
| Frame 4 | 0.4077 | 0.3304 | 19 | 1 |
| Frame 5 | -1.5700 | -1.6806 | 7 | 2 |
| Frame 6 | -2.0202 | -1.6312 | 19 | 5 |
| Frame 7 | -2.8108 | -2.0373 | 28 | 11 |
| Frame 8 | 0.3409 | 0.5880 | 72 | 6 |
| Azimuth 1 | 0.7001 | 0.5607 | 20 | 2 |
| Azimuth 2 | -0.4060 | -0.2945 | 27 | 2 |
| Elevation | 0.0226 | 0.0028 | 88 | 0 |

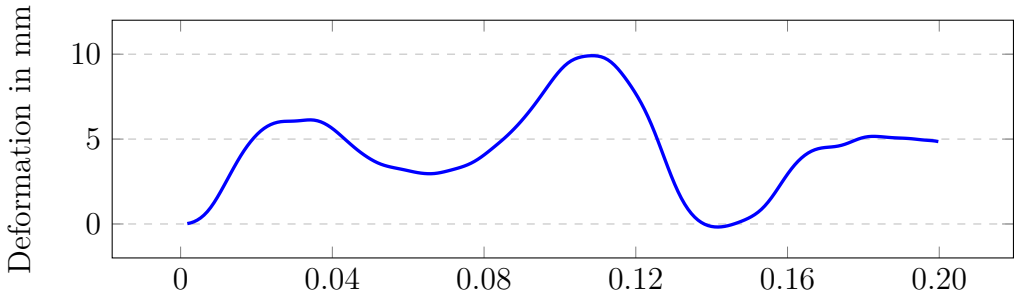
**Tab. 5.2** Static Comparison Between Shell and Solid Element Models

that difference can be neglected.

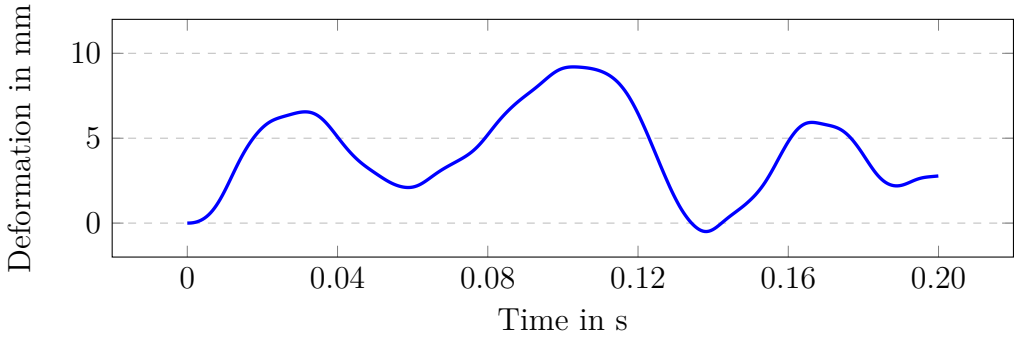
Overall, the alignment of the response for both models is satisfactory, with the largest weighted deviation being $d_{wi} = 11\%$, and the solid element complex model may be relied upon for the validation of the simplified models in static simulations. The resulting equivalent stress in the solid element complex model from this static test is discussed in Section 5.6 concering the determination of the DLF for DLR's heliostat.

### 5.2.2 Dynamic Test

The dynamic load response of the two complex models is plotted in Figures 5.2 to 5.5. Each figure consists of two subfigures: the first subfigure shows the reference plot with evenly distributed ticks marking the same timesteps as in the second subfigure, which displays the results from the solid element complex model (this applies to all dynamic tests in this chapter).

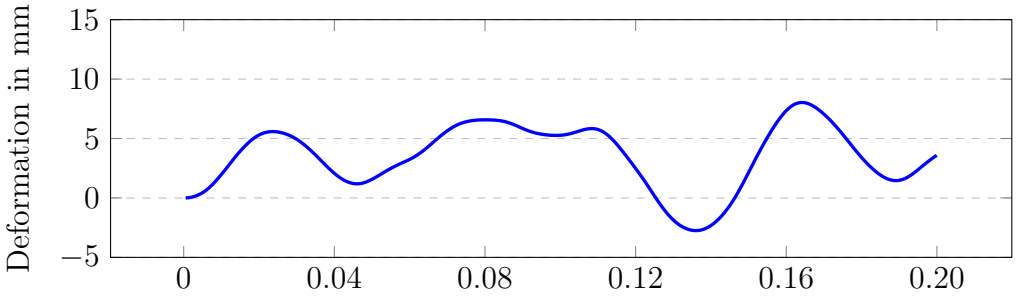



**(a)** Shell Element Complex Model

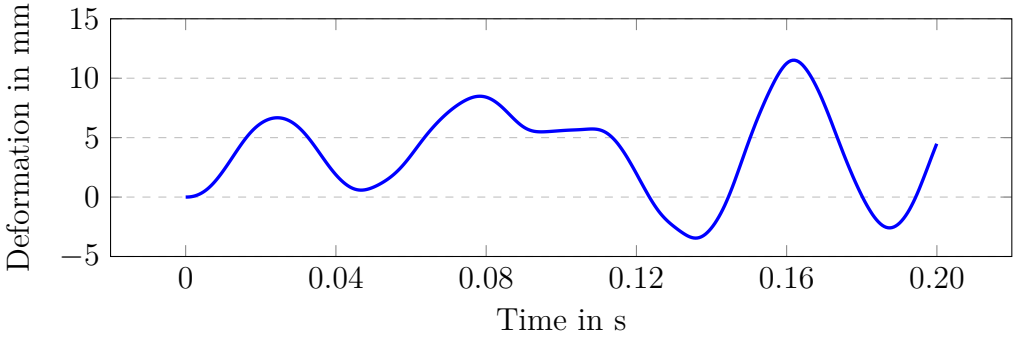



**(b)** Solid Element Complex Model

**Fig. 5.2** Dynamic Response for Complex Models at Frame 1

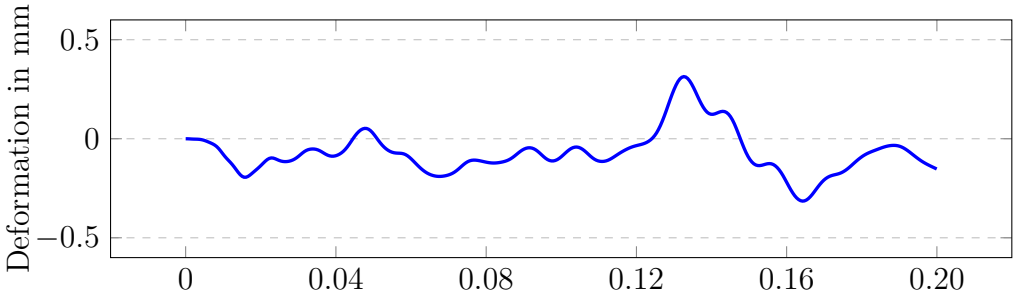



**(a)** Shell Element Complex Model

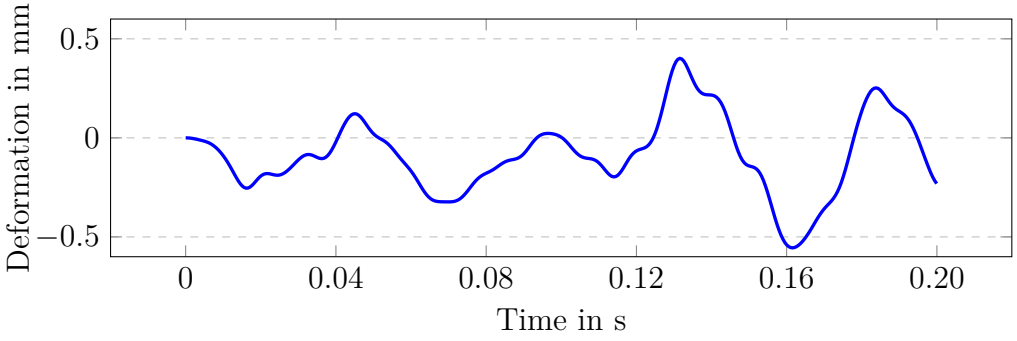



**(b)** Solid Element Complex Model

**Fig. 5.3** Dynamic Response for Complex Models at Frame 2

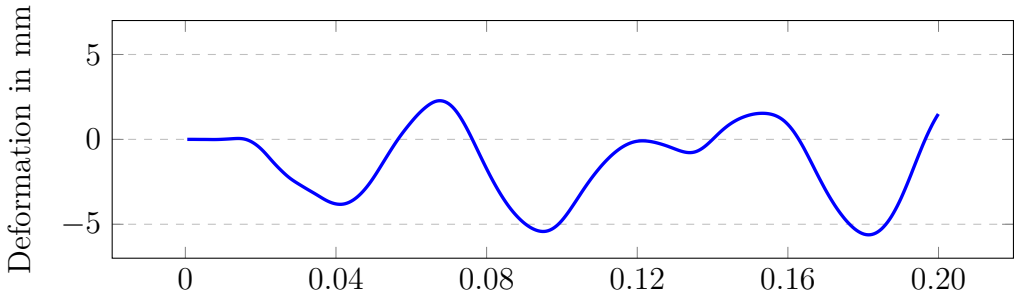



**(a)** Shell Element Complex Model

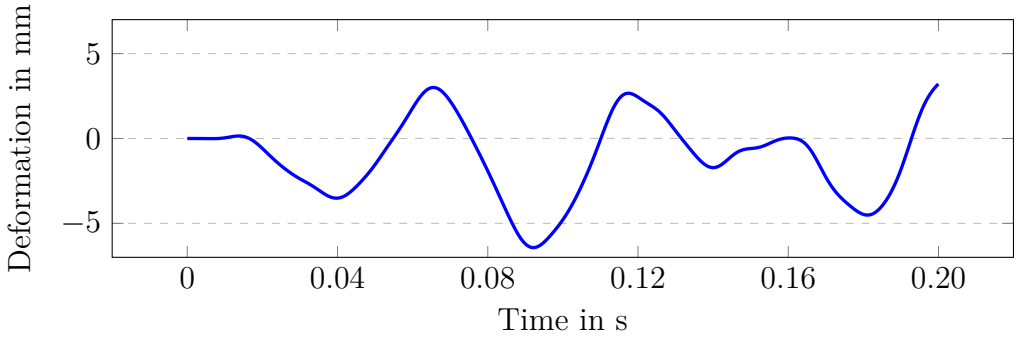



**(b)** Solid Element Complex Model

**Fig. 5.4** Dynamic Response for Complex Models at Frame 3




**(a)** Shell Element Complex Model




**(b)** Solid Element Complex Model

**Fig. 5.5** Dynamic Response for Complex Models at Frame 4

The peak load comparisons and their corresponding deviations are summarized in Table 5.3.

| Component | Shell Model [mm] | Solid Model [mm] | Deviation (%) [%] | Weighted Deviation [%] |
|---|---|---|---|---|
| Frame 1 | 9.9136 | 9.1987 | 8 | 7 |
| Frame 2 | 8.0324 | 11.5120 | 30 | 30 |
| Frame 3 | 0.3147 | 0.5551 | 43 | 2 |
| Frame 4 | -5.6224 | -6.4303 | 13 | 7 |

**Tab. 5.3** Dynamic Test Peak Loads Comparison Between Shell and Solid Element Models

When weighted, only frame 2 exceeds the targeted 10%. In Figure 5.3, the curves for frame 2 follow a consistent pattern, with their peaks occurring at the same timestamp, making the correspondence of their behavior satisfactory.

Figures 5.2 and 5.5, which show frames 1 and 4 respectively, also show good correspondence despite slight differences in the shape of their peaks and their timestamps. The plots for frame 3 in Figure 5.4 display modulation which appears to occur at a very similar oscillation, further pointing to good alignment between the two models.

Overall, the two models show good correspondence despite their large fundamental differences in the way they are modeled. Not only do these results increase the confidence in validation results for the following sections, but also do they prove that both modeling techniques are useful for these purposes, both possessing their strengths and weaknesses. In all tables and diagrams following this section, the complex solid element model is simply referred to as complex model or reference model.

## 5.3 Validation of Simplified Simulation Models

Section 4.5 describes in detail how the models are validated. The results obtained from the multibody and ansys models are compared to the same test done with another ansys model, meshed with shell elements.

### 5.3.1 Multibody Model - Static Test

Because the multibody model solely exists within the time domain, and steady-state analysis isn't directly available, performing a static test becomes a little bit non-ideal. The structure always starts with no deformation, which means that applying a direct load spontaneously results in the model deforming very rapidly, resulting in high oscillations which become unusable for this test. By introducing the constant load with a built-in ramp block in Simulink, the load is gradually applied. This way, all oscillations are minimized, and the model stabilizes quickly while deforming smoothly. The results are summarized in table 5.4.

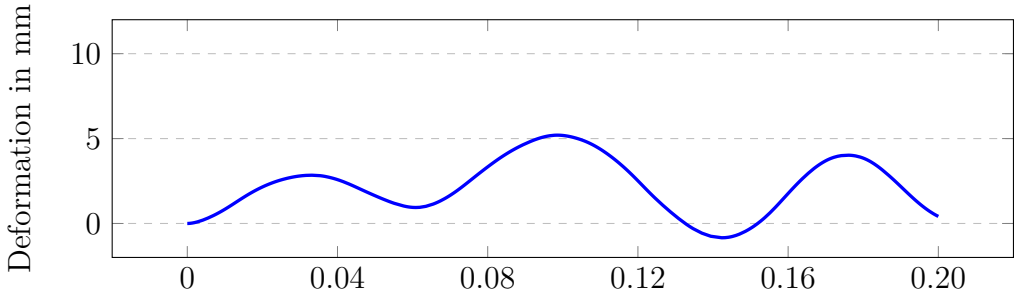
Despite resembling somewhat of a comparable response to that of the complex model, the overall results are poor. Deformations for frames 1 to 3 are all significantly smaller than from the reference model. On the heliostat's opposite end, the multibody structure deforms almost an equal amount in the downward direction, indicating that the carrier frame is excessively rigid. In fact, the T-mount below does respond similarly to the complex model. Moreover, replacing the solar glass mirror with a significantly more flexible material doesn't change the large discrepancies of Table 5.4, further backing the hypothesis that the underlying issue is in the carrier frame.

| Component | Simple Multibody Model [mm] | Complex Model [mm] | Deviation [%] | Weighted Deviation [%] |
|---|---|---|---|---|
| Frame 1 | 3.9636 | 7.2028 | 45 | 45 |
| Frame 2 | 3.9789 | 6.6609 | 40 | 37 |
| Frame 3 | 4.0170 | 5.6012 | 28 | 22 |
| Frame 4 | 0.1267 | 0.3304 | 62 | 3 |
| Frame 5 | -3.3285 | -1.6806 | 98 | 45 |
| Frame 6 | -3.3667 | -1.6312 | 106 | 50 |
| Frame 7 | -3.3838 | -2.0373 | 66 | 31 |
| Frame 8 | 0.0715 | 0.5880 | 88 | 7 |
| Azimuth 1 | 0.3327 | 0.5607 | 41 | 3 |
| Azimuth 2 | -0.2882 | -0.2945 | 2 | 0 |
| Elevation | 0.0189 | 0.0028 | 577 | 2 |

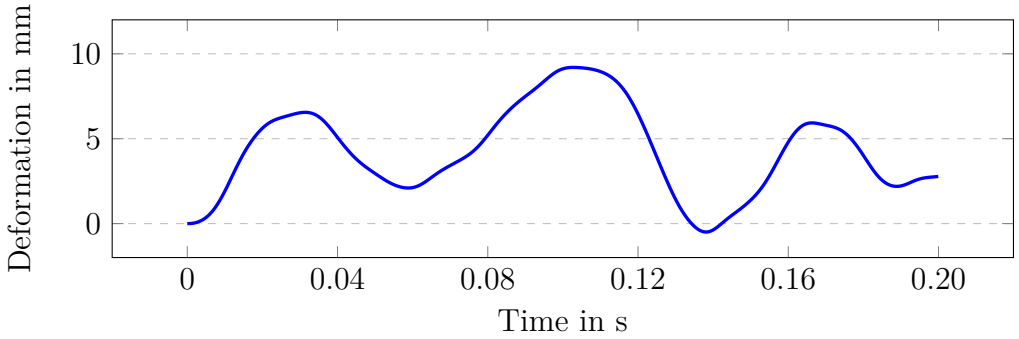
**Tab. 5.4** Static Comparison Between Multibody and Complex Model

### 5.3.2 Multibody Model - Dynamic Test

Just like in Section 5.2.2, the same four frames are simulated on the multibody model and compared to the complex model, as shown in Figures 5.6 to 5.9. Table 5.5 follows, summarizing their peaks.

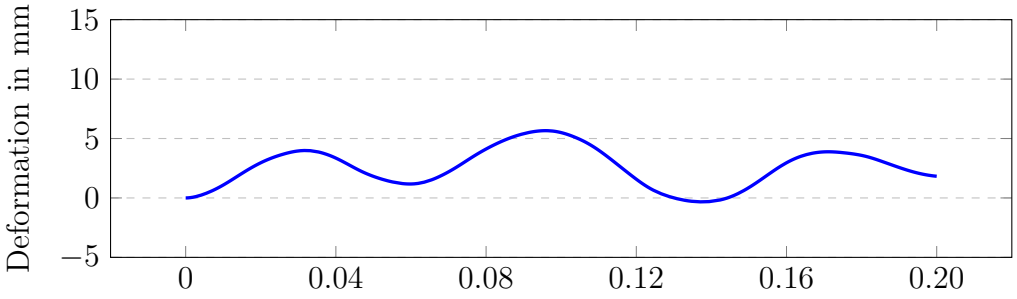



**(a)** Multibody Simple Model

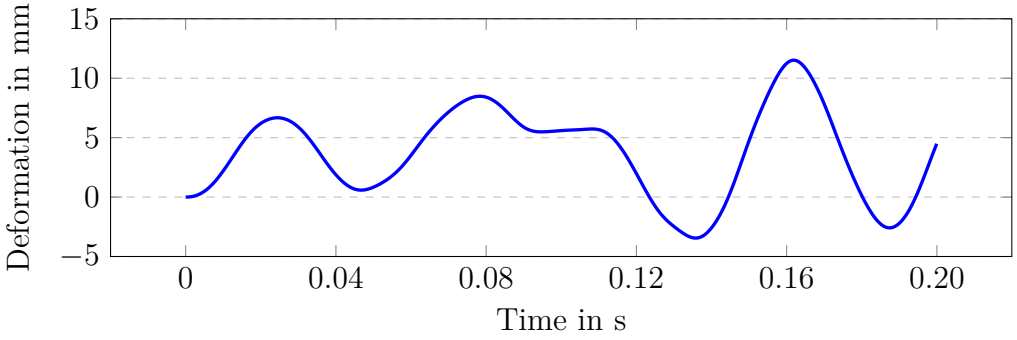



**(b)** Solid Element Complex Model

**Fig. 5.6** Dynamic Response for Multibody and Complex Model at Frame 1

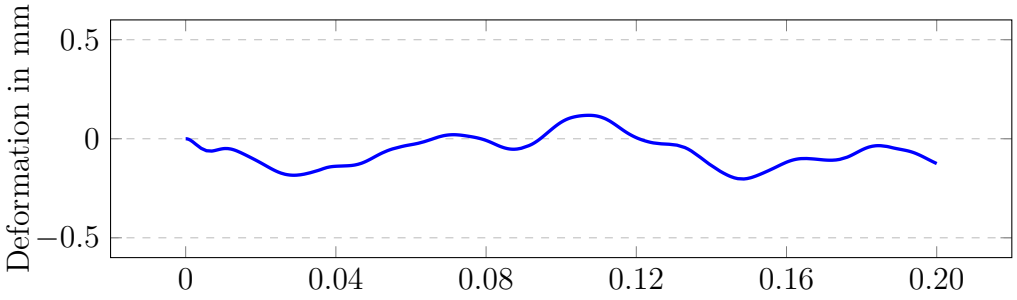



**(a)** Multibody Simple Model

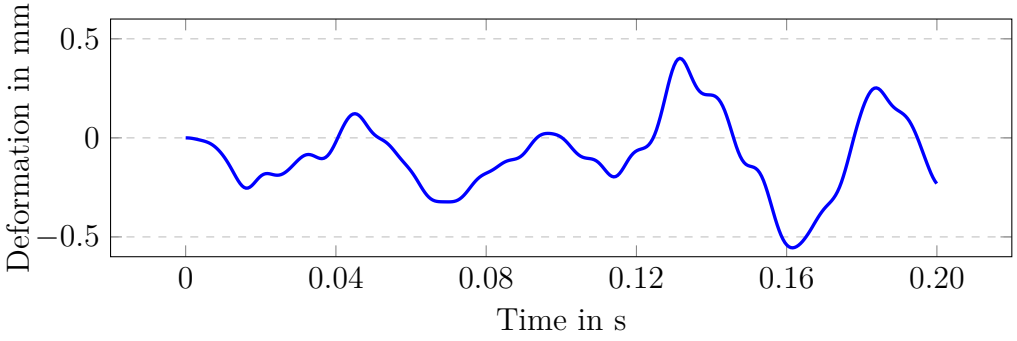



**(b)** Solid Element Complex Model

**Fig. 5.7** Dynamic Response for Multibody and Complex Model at Frame 2

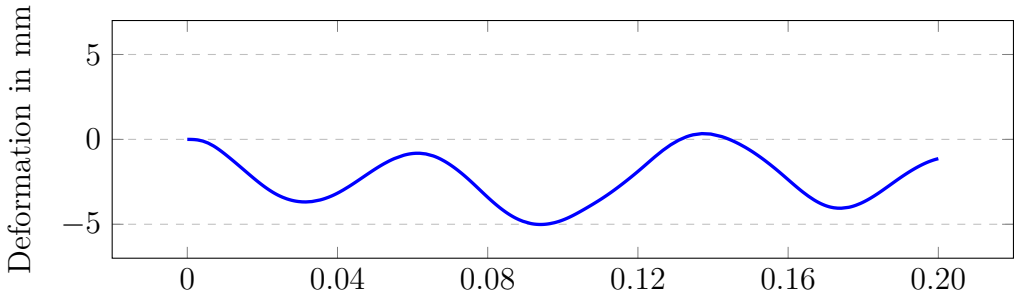



**(a)** Multibody Simple Model

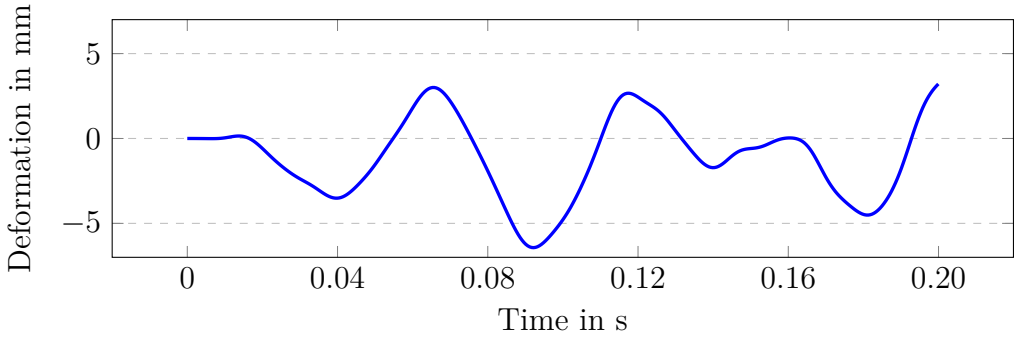



**(b)** Solid Element Complex Model

**Fig. 5.8** Dynamic Response for Multibody and Complex Model at Frame 3




**(a)** Multibody Simple Model




**(b)** Solid Element Complex Model

**Fig. 5.9** Dynamic Response for Multibody and Complex Model at Frame 4

| Component | Simple Multibody Model | Complex Model | Deviation | Weighted Deviation |
|---|---|---|---|---|
| | [mm] | [mm] | [%] | [%] |
| Frame 1 | 5.2006 | 9.1987 | 43 | 35 |
| Frame 2 | 5.6599 | 11.5120 | 51 | 51 |
| Frame 3 | -0.2030 | -0.5551 | 63 | 3 |
| Frame 4 | -5.0127 | -6.4303 | 22 | 12 |

**Tab. 5.5** Dynamic Test Peak Loads Comparison Between Simplified Shell and Complex Solid Element Models

Unfortunately, the response of the multibody model shows very little resemblance to the complex model, both in its plotted peaks as well as in their curves. The mode shapes seem to have similar frequencies, but the multibody model exhibits an overly smooth response compared to the complex model's more feature-rich and detailed graph, pointing to potentially excessive stiffness and damping.

### 5.3.3 Simple Shell Model - Static Test

The same table from Section 5.3.1 is repeated to compare the results between the simplified shell element model and the complex model (Table 5.6).

| Component | Simple Shell Model [mm] | Complex Model [mm] | Deviation [%] | Weighted Deviation [%] |
|---|---|---|---|---|
| Frame 1 | 7.2551 | 7.2028 | 1 | 1 |
| Frame 2 | 7.4705 | 6.6609 | 11 | 11 |
| Frame 3 | 6.4774 | 5.6012 | 14 | 12 |
| Frame 4 | 0.4423 | 0.3304 | 25 | 1 |
| Frame 5 | -2.2454 | -1.6806 | 25 | 8 |
| Frame 6 | -2.2319 | -1.6312 | 27 | 8 |
| Frame 7 | -2.7108 | -2.0373 | 25 | 9 |
| Frame 8 | 0.2088 | 0.5880 | 182 | 14 |
| Azimuth 1 | 0.6063 | 0.5607 | 8 | 1 |
| Azimuth 2 | -0.3768 | -0.2945 | 22 | 1 |
| Elevation | 0.0187 | 0.0028 | 85 | 0 |

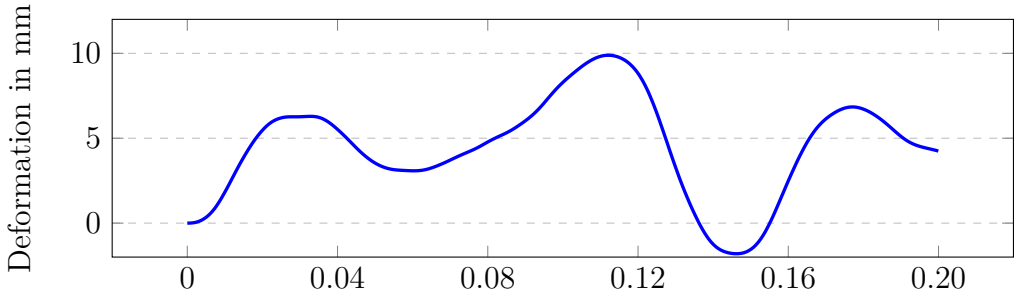
**Tab. 5.6** Static Comparison Between Simple Shell Element and Complex Model

In comparison to the results from the multibody model where most measurements in the carrier frame had deviations exceeding 30%, a significant improvement was observed with the simplified shell element model, with the critical frames only having 12% deviation from the reference model when weighted. Compared to the benchmark test in Table 5.2, the deformation distribution displays some inconsistencies, as some frames deform more while others deform less, indicating that the model may be nearing, but has not fully met, its convergence criteria. This is, however, to be expected when the element count is reduced by an order of magnitude from the original model.

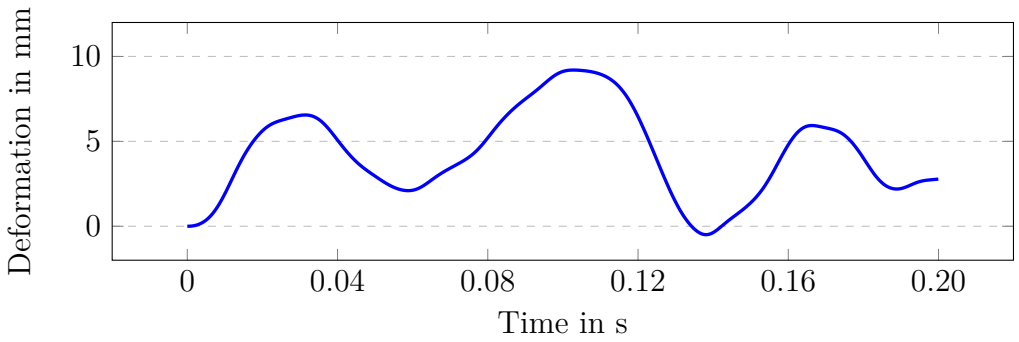
Similar to the benchmark test, frame 4 has the same deviation of 1% which is good considering the deformation being within 1 mm. Frames 4 and 8 are located at the heliostat's center relative to the wind's direction, and are expected to exhibit minimal bending. Although 14% weighted deviation at frame 8 is more than first anticipated, being only a 3% increase from the biggest deformation in Table 5.2, from the static benchmark test, it is still acceptable, considering the main purpose of the simplified models is to find the most critical response throughout the entire wind tunnel dataset. This confirms that the simplified shell element model is reliable for conducting static analyses.

### 5.3.4 Simple Shell Model - Dynamic Test

For the final dynamic comparison, the same procedure as in Sections 5.2.2 and 5.3.2 is performed to see the curve of the plots of the simple shells model and the original complex model, as shown in Figures 5.10 to 5.13. These are followed by Table 5.7, which summarizes their peaks.

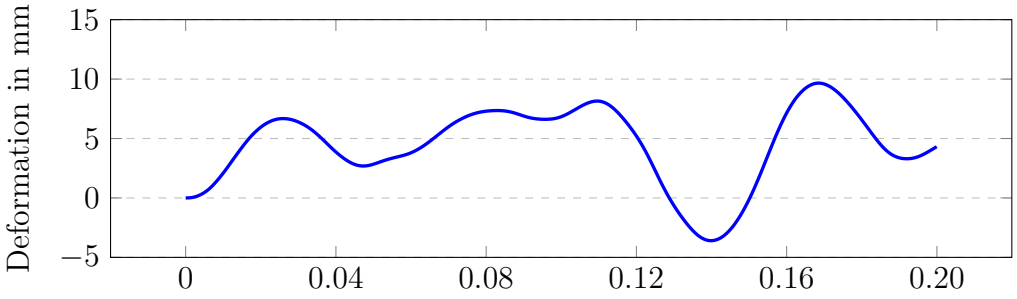



**(a)** Shell Element Simple Model

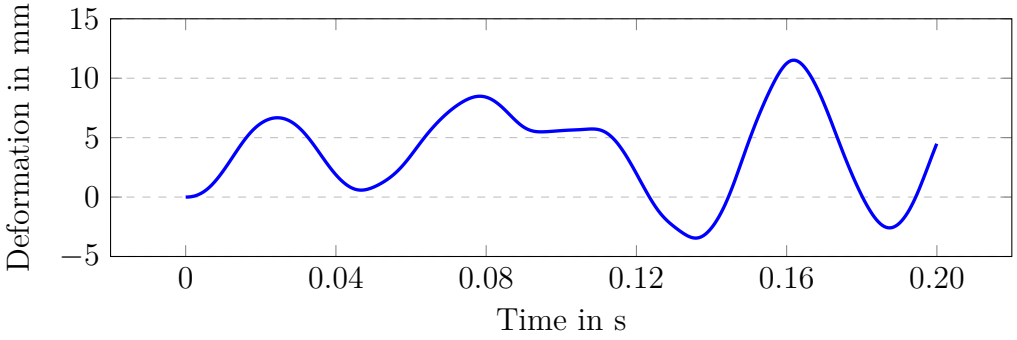



**(b)** Solid Element Complex Model

**Fig. 5.10** Dynamic Response for Shells and Complex Model at Frame 1

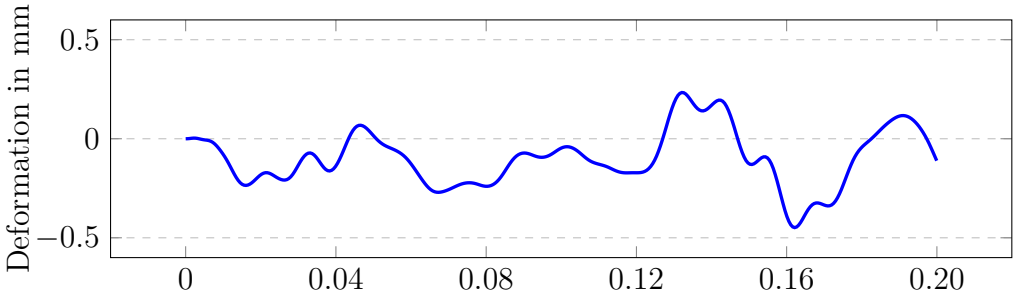



**(a)** Shell Element Simple Model

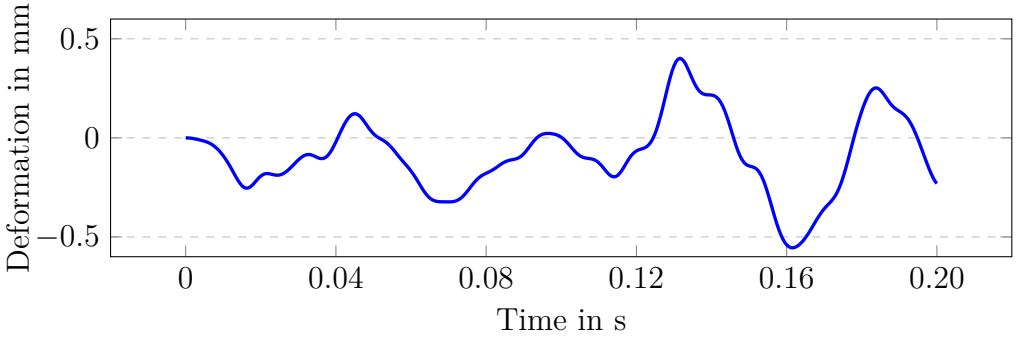



**(b)** Solid Element Complex Model

**Fig. 5.11** Dynamic Response for Shells and Complex Model at Frame 2

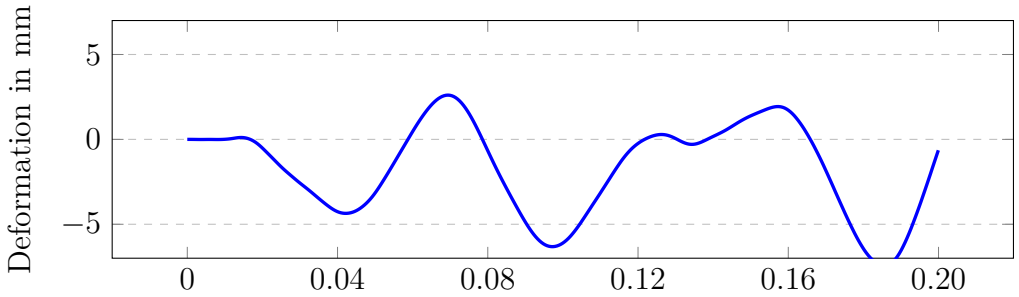



**(a)** Shell Element Simple Model

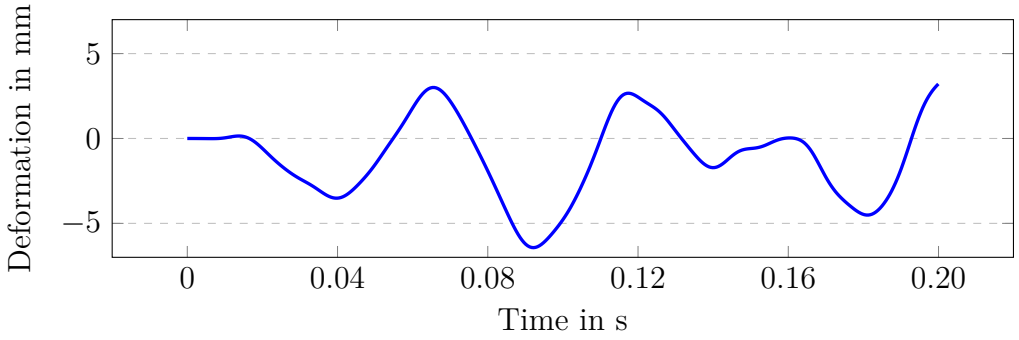



**(b)** Solid Element Complex Model

**Fig. 5.12** Dynamic Response for Shells and Complex Model at Frame 3




**(a)** Shell Element Simple Model




**(b)** Solid Element Complex Model

**Fig. 5.13** Dynamic Response for Shells and Complex Model at Frame 4

| Component | Simple Shell Model [mm] | Complex Model [mm] | Deviation [%] | Weighted Deviation [%] |
|---|---|---|---|---|
| Frame 1 | 9.8853 | 9.1987 | 7 | 6 |
| Frame 2 | 9.6606 | 11.5120 | 16 | 16 |
| Frame 3 | -0.4485 | -0.5551 | 19 | 1 |
| Frame 4 | -7.4443 | -6.4303 | 16 | 10 |

**Tab. 5.7** Dynamic Test Peak Loads Comparison Between Simplified Shell and Complex Solid Element Models

Again, a big improvement is observed, as the plots closely resemble each other. Figure 5.4 shows closely matching modulation in the curves, further demonstrating that the reduction in modes from Section 5.1 does not cause an interferance in the structure's freedom of movement. Combined with the fact that the peaks from table 5.7 don't deviate much, the simple shell model is selected to be implemented for the full-duration study.

## 5.4 Results of the Full-Duration Study

The full-duration study on the simplified shell model in Ansys was much quicker than initially anticipated. This can be attributed to continuous refinements of the Ansys solver over the years, the reduction model in complexity as described in Section 4.4.2 and the high-performance workstation used to carry out the simulation.

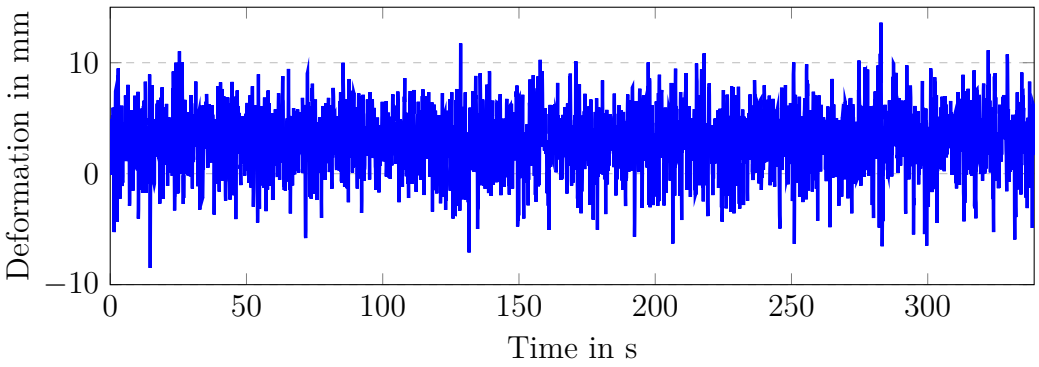



**Fig. 5.14** Deformation Signal of Frame 1

A Fast Fourier Transform (FFT) diagram gives more insight into the frequency components of the full-duration study. Because the mean value in Figure 5.14 is around 3 mm (instead of 0 mm), a DC component is introduced, resulting in a spike at 0 Hz in a FFT diagram. This occurs because the offset adds a constant value to the signal, which appears as a peak at 0 Hz, potentially distorting the true frequency components. To counter this, a high-pass filter that filters out all frequencies from 0 to 2 Hz is applied, as well as a low-pass filter to remove the symmetric component in the higher frequency spectrum. The first 1.5 seconds is shown in Figure 5.15.

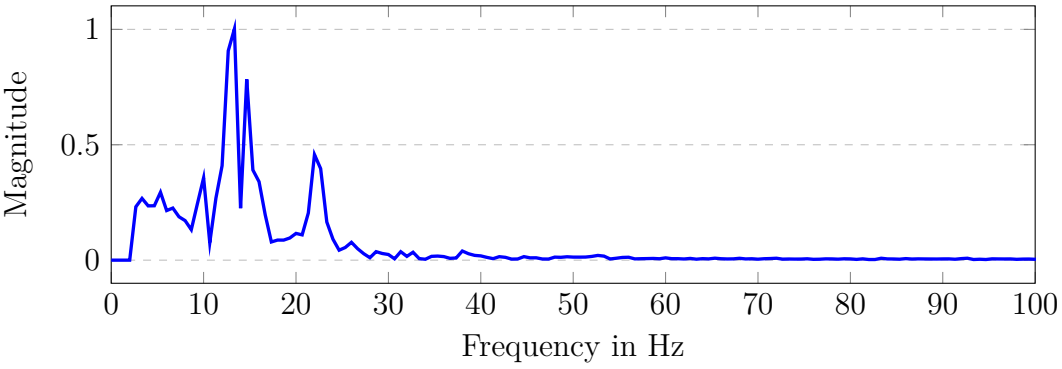



**Fig. 5.15** FFT diagram of 1.5-Second Snippet

By incorporating temporal information into the frequency analysis, the frequency components are visualized over the course of the full-duration study, both with a

Short-Time Fourier Transform (STFT) plot and a spectrogram. Here, the time-frequency representations both have Kaiser windows applied, and the spectrogram uses a threshold of -40 db/Hz. Figure 5.16 shows the STFT plot, which depicts the signal magnitude in colors ranging from blue (0 dB) to bright yellow (75 dB), while the spectrogram in Figure 5.17 returns the Power Spectral Density (PSD) ranging from -40 db/Hz to 15 db/Hz.

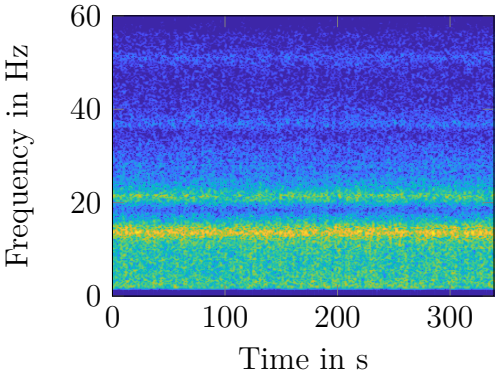



**Fig. 5.16** STFT Plot

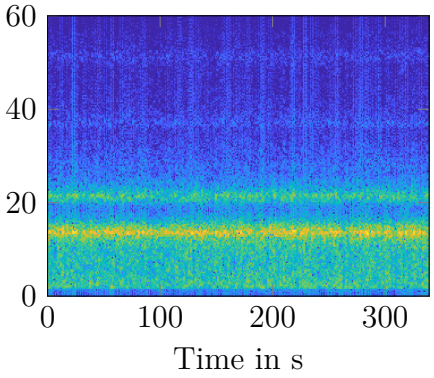



**Fig. 5.17** Spectrogram

Figures 5.16 and 5.17 show a stable and continuous signal that fits well with what was observed in the FFT diagram from Figure 5.15 where the most dominant frequency is $f_1 = 13.3$ Hz, followed by $f_2 = 22.0$ Hz.

From observation, it's still difficult to see where in the timeseries signal peaks occur. The time frequency representations from Figures 5.16 and 5.17 show bright marks on various timestamps throughout the plots, but these alone are insufficient for any conclusions and require further analysis. In Figure 5.14, which depicts the raw signal obtained from Ansys, a clear peak can be observed at $t = 282.9$ s. A 3D representation of the spectrogram in Figure 5.17 can better highlight spikes in the signals' spectral power.

Figure 5.18 shows that the signal indeed peaks at t = 282.9 s, emitting 11.830 kW/Hz at a frequency of f = 13.56 Hz. The deformation signals, their time-frequency representations, and the corresponding 3D representations for frames 2, 3, and 4 are summarized in Appendix C. Table 5.8 summarizes the peaks of the frames used in the full-duration study. Interestingly, the detected timestamps differ from the one determined in Section 4.2.2, which was $t = 215.151$ s.

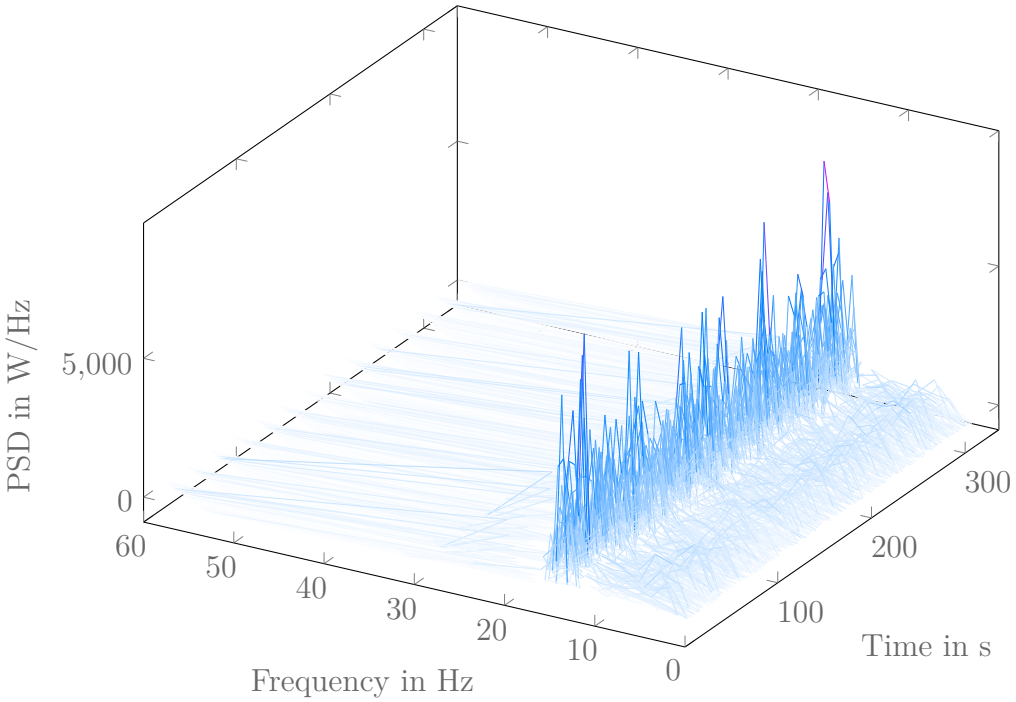



**Fig. 5.18** 3D Representation of Figure 5.17

| Component | Timestamp [s] | Frequency [Hz] | PSD [W/Hz] |
|---|---|---|---|
| Frame 1 | 282.927 | 13.56 | 11830.3 |
| Frame 2 | 217.944 | 13.87 | 9331.7 |
| Frame 3 | 282.927 | 13.56 | 517.5 |
| Frame 4 | 217.944 | 13.87 | 9094.9 |

**Tab. 5.8** Summary of Peaks from Full-Duration Study

## 5.5 Stresses at Critical Timestamps

As described in Section 4.7, the time interval for a dynamic short-duration load is based of the critical timestamps from the full-duration study. Seeing that there are two critical timestamps at different frequencies, two sets of load data are extracted accordingly. By considering three periods in the simulation, and using equation 4.4, the time intervals become

$$t_{\text{interval 1}} = 0.216 \text{ s} \quad \text{and} \quad t_{\text{interval 2}} = 0.221 \text{ s}$$

### 5.5.1 First Stress Interval

Subjecting the complex solid model of DLR's heliostat to the short-duration peak loads results in a peak stress of $\sigma_{\text{max}} = 403.07$ MPa. The location of $\sigma_{\text{max}}$ can be seen in Figures 5.19 and 5.20.

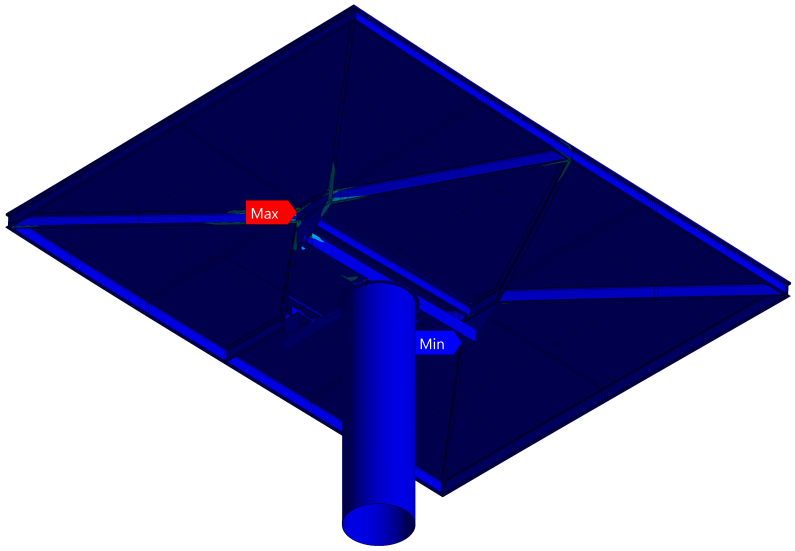



**Fig. 5.19** Equivalent Stresses of the Heliostat as Seen from Below During the First Stress Interval

Instead of manually selecting specific regions for probing to identify stress peaks, Ansys can automatically perform a probe to find local maximum stresses. By looking for the six highest stress peaks, it becomes clear that most of them are in close proximity to the highest peak, with the sixth peak being on the opposite end of the mounting bracket, showing a significantly lower stress of $\sigma = 339.94$ MPa (Figure

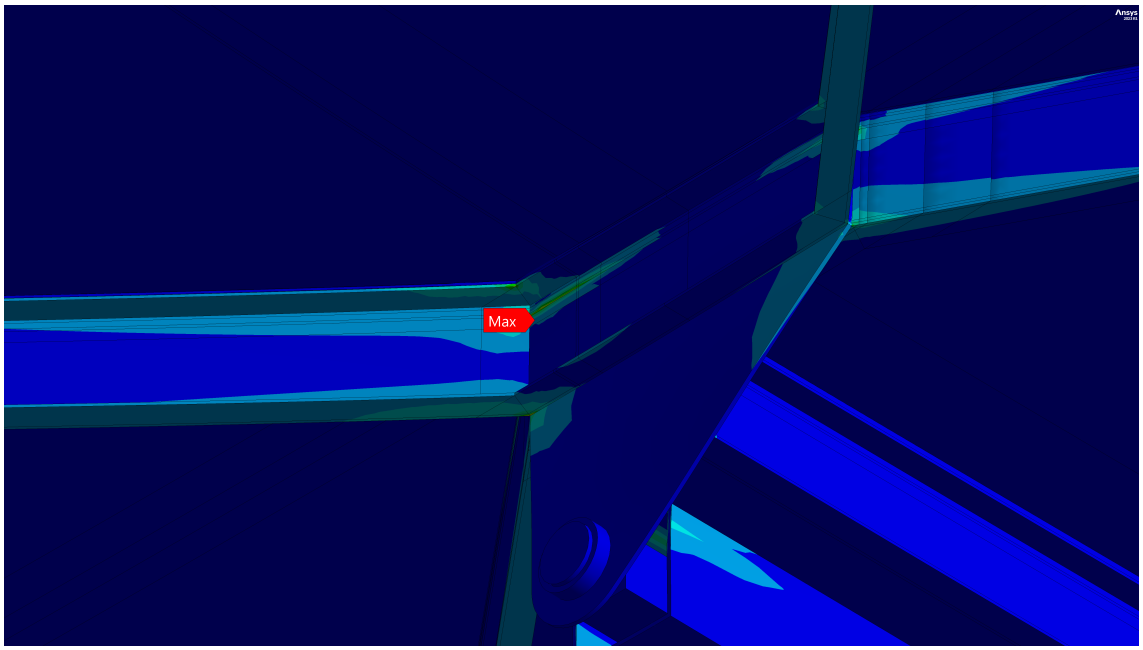



**Fig. 5.20** Equivalent Stresses of the Heliostat with Close-up on Peak During the First Stress Interval

5.21).

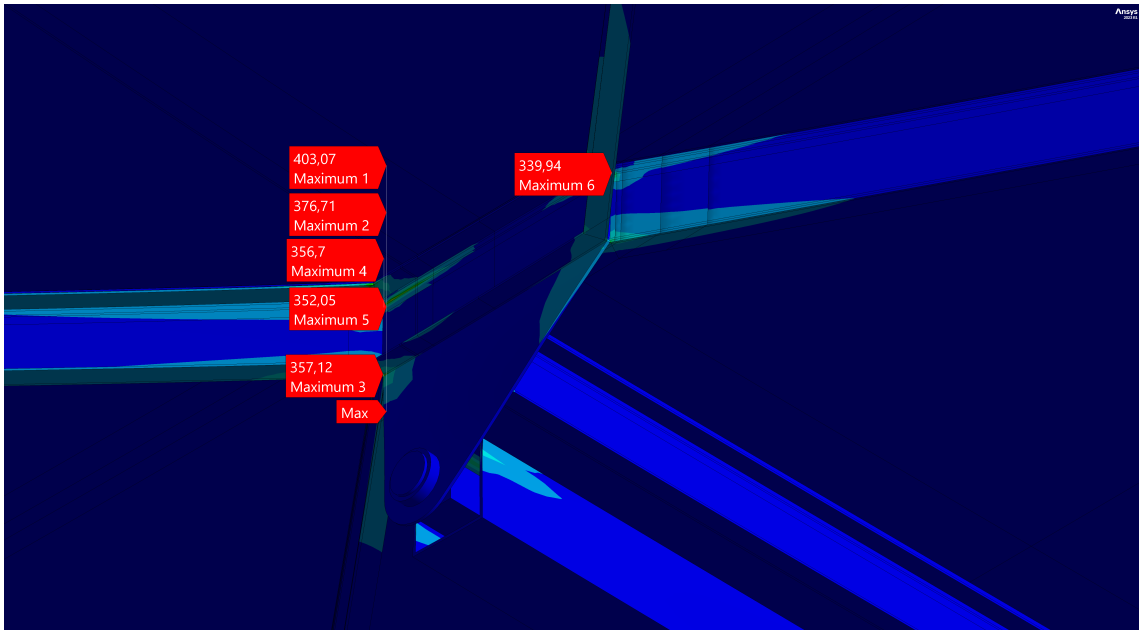



**Fig. 5.21** Local Maximum Probe of Equivalent Stress Solution During the First Stress Interval

These results point out that the highest stresses occur at the brackets connecting the carrier frame to the T-mount via the azimuth joints on the same side as where the wind hits the heliostat. Interestingly, the same bracket for the second azimuth joint experience the least stress, underscoring how much stress the windward side

of the carrier frame absorbs.

### 5.5.2 Second Stress Interval

Following the same steps as for the first stress interval results in Figure 5.22 and 5.23.

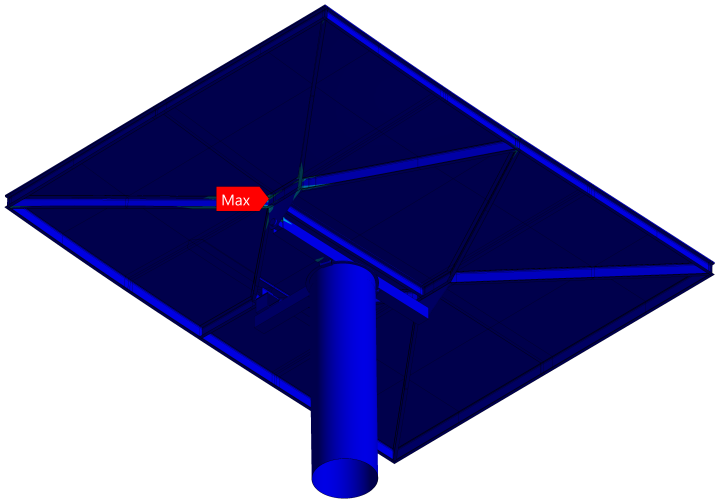



**Fig. 5.22** Equivalent Stresses of the Heliostat as Seen from Below During the Second Stress Interval

The peak stress ends up in the exact same region despite the slightly different frequency at which the heliostat vibrates. With $\sigma_{\max} = 582.63$ MPa, an almost 50% increase is observed, making the second interval much more critical. A local stress probe for the maximum stresses shows that the six highest stresses are all in the same region, with the seventh occurring on the opposite end of the same mounting bracket (Figure 5.24).

This confirms that, of the two intervals, the second one results in the highest stress, whose value can be implemented to obtain the DLF.

## 5.6 Determining the Dynamic Load Factor

It's worth looking at two maximum stresses resulting from a static load when determining the DLF. First, the same static load used in Section 4.5.1 is applied,

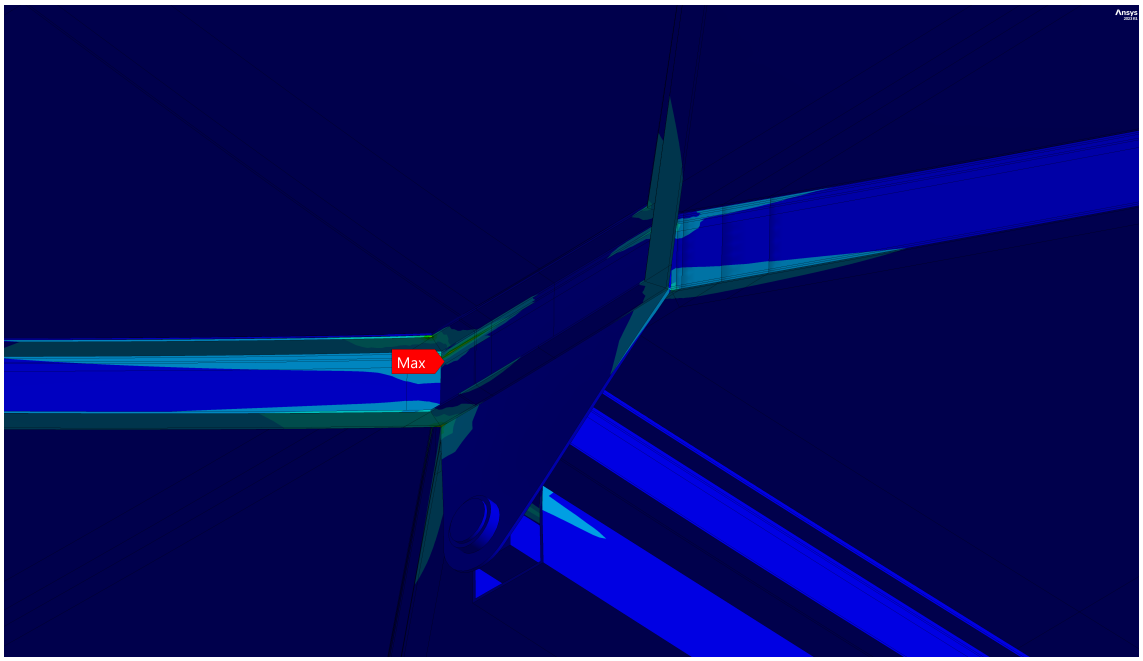



**Fig. 5.23** Equivalent Stresses of the Heliostat with Close-up on Peak During the Second Stress Interval

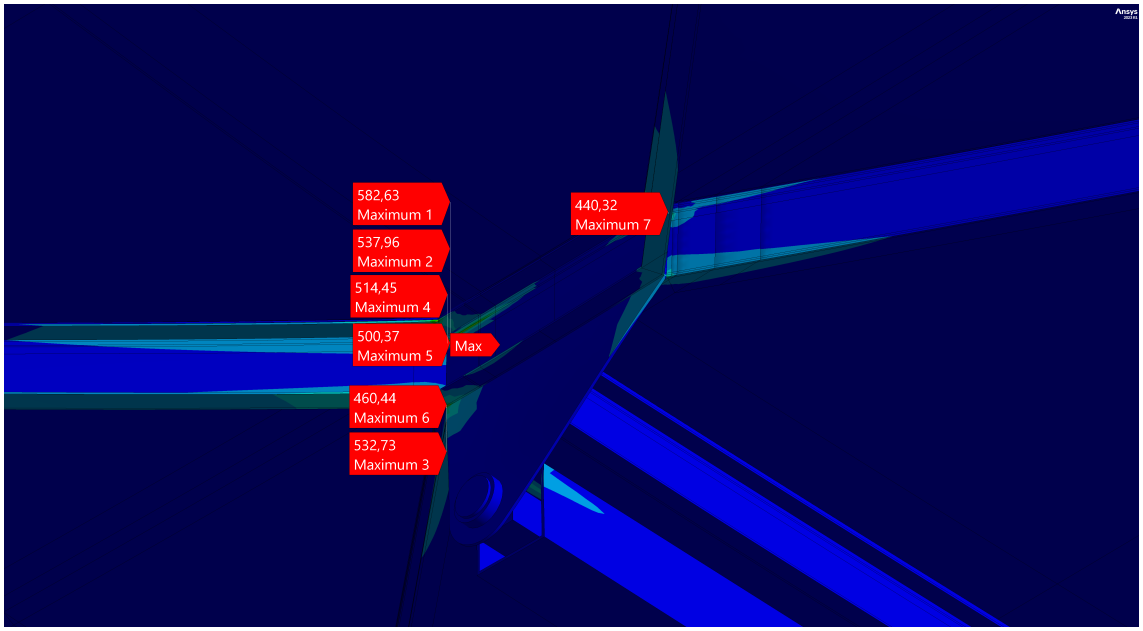



**Fig. 5.24** Local Maximum Probe of Equivalent Stress Solution During the Second Stress Interval

as it's an easy load to reference, having been used repeatedly throughout this thesis. Figure 5.25 shows the location of the peak stress resulting from the applied load.

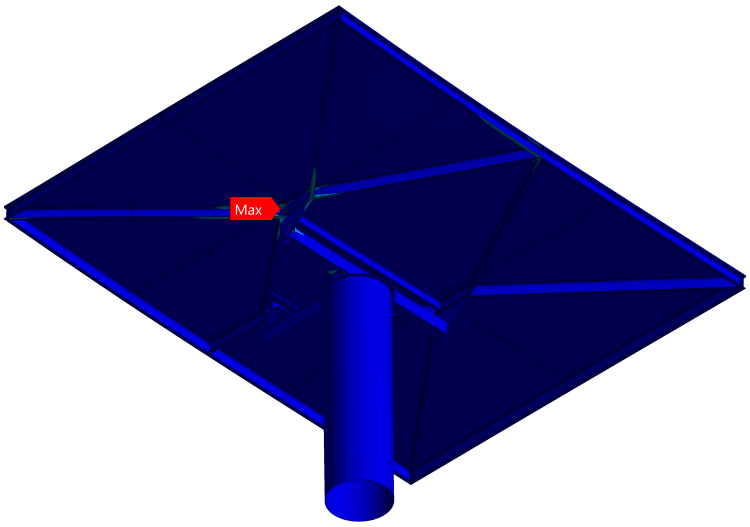



**Fig. 5.25** Location of Peak Stress from First Static Load

The maximum stress is $\sigma_{\max} = 422.46$ MPa and occurs at the exact same location as the stresses in Section 5.5. Using equation 4.5 described in Section 4.8, the DLF is calculated as

$$\text{DLF} = \frac{\sigma_{\max}\ (\text{Transient})}{\sigma_{\max}\ (\text{Static})} = \frac{582.63\ \text{MPa}}{422.46\ \text{MPa}} \approx 1.4$$

The second static load is based of the same timestamp at which the maximum dynamic stress response occurred in Section 5.5.2. Figure 5.26 shows the location of the peak stress response using this load.

Again, the response occurs at the same location as in the previous simulations, but now at a much lower value of $\sigma_{\max} = 289.53$ MPa, resulting in a DLF of

$$\text{DLF} = \frac{582.63\ \text{MPa}}{289.53\ \text{MPa}} \approx 2.0$$

A DLF of 2.0 matches the result obtained in Project MAHWIN, as discussed in Chapter 2 [15]. Both DLFs of 1.4 and 2.0 are reasonable in structural mechanics and fit within the expected range of the heliostat. It can be further implemented

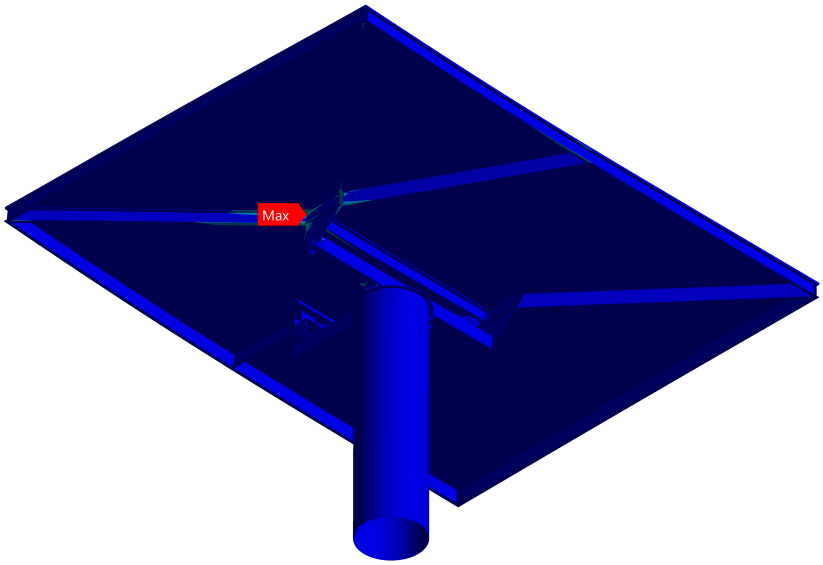



**Fig. 5.26** Location of Peak Stress from Second Static Load

for different iterations of the heliostat to get a quick estimate of the higher stresses introduced by vibrations, intertial forces and damping that occurs during dynamic analysis such as in storm conditions.

Note, the stress values obtained are all well above the common yield strength of $\sigma_y = 250$ MPa for structural steel, and repeated stress cycles for DLR's heliostat of this magnitude are unlikely to be sustained. However, since all simulations in this thesis are linear and do not account for non-linear deformations, the determined DLF remains accurate.

CHAPTER 6

# Summary & Outlook

This thesis experimented with different techniques to model a heliostat from the German Aerospace Center (DLR), aimed at dynamic wind load analysis.

## 6.1 Summary

The objective of this thesis was to discover critical loads on DLR's heliostat and their effects on its structure. Because wind patterns are highly heterogeneous, a critical event cannot be detected easily. Long simulations need to be processed and analyzed, which in the context of numerical simulations means durations lasting a few minutes. A large dataset of wind load data cannot be analyzed on its own, as it doesn't perfectly reflect the response it induces on a heliostat. In order for these simulations to not exceed memory constraints, the heliostat models were simplified. Two methods for simplifying the model were used, namely the Mode Superposition (MSUP) method and the Reduced Order Model (ROM) method. Both models ran simulations with relatively low computational time, but only the shell element model using the MSUP method passed the validation tests. In contrast, the multibody model, which used the ROM method, failed due to limitations in the tools

used to generate the simplified model data.

The long-duration study returned results in the form of four components on the heliostat that measured deformations along the vertical axis, resembling a noise signal on a plot. Signal processing tools enable disassembling the frequency components of these signals and easier detection of peaks, indicating critical loads. Two distinct peaks were found, and both were reapplied on a more sophisticated dynamic simulation model of DLR's heliostat. The results indicated that at the second peak, which occurs later in the long-duration study, the heliostat shows a higher stress response, with the stress accumulating in the upper region of the heliostat's bracket, near its azimuth joint. Finally, the Dynamic Load Factor (DLF) was determined by comparing the results of the dynamic simulation with an equivalent static simulation.

## 6.2 Outlook

For future iterations of the methods outlined in this thesis, several key improvements can be implemented to improve the accuracy and reliability of the models. One important improvement is ensuring that both the solid element and shell element models have consistent wall thicknesses across the models, which would lead to more accurate results. Furthermore, clear decisions about the fillets should be made and applied consistently across both models. Beyond these adjustments, this thesis serves as a guideline for creating dynamic simulation models of heliostats across a range of sizes, providing a good foundation for future iterations of DLR's heliostat. By incorporating the mentioned improvements, future iterations can achieve even higher precision and effectiveness, ultimately advancing the field of solar energy technology.

# Bibliography


[1] Wikipedia contributors. Concentrated solar power — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Concentrated_solar_power&oldid=1237747081`, 2024. Accessed: 14.7.2024.

[2] Robert Pitz-Paal, Elke Reschenschbach, and Klaus Hennecke, editors. *Solarthermische Kraftwerke: Wärme, Strom und Brennstoffe aus konzentrierter Sonnenenergie*. Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Köln, 2021. Institut für Solarforschung, Institut für Technische Thermodynamik, Institut für Vernetzte Energiesysteme.

[3] Andreas Pfahl, Joe Coventry, Marc Röger, Fabian Wolfertstetter, Juan Felipe Vásquez-Arango, Fabian Gross, Maziar Arjomandi, Peter Schwarzbözl, Mark Geiger, and Phillip Liedke. Progress in heliostat development. *Solar Energy*, 152:3–37, 2017. Progress in Solar Energy Special Issue: Concentrating Solar Power (CSP).

[4] Andreas Pfahl. *Wind loads on heliostats and photovoltaic trackers*. [phd thesis 1 (research tu/e / graduation tu/e), built environment], Technische Universiteit Eindhoven, 2018.

[5] DLR. Zehn jahre dlr-institut für solarforschung. https://www.dlr.de/de/das-dlr/ueber-uns/geschichte-des-dlr/zehn-jahre-dlr-institut-fuer-solarforschung, 2021. Accessed: 30.7.2024.

[6] Sika Deutschland GmbH. *PRODUKTDATENBLATT: Sikasil® AS-785*. Stuttgarter Straße 139, 72574 Bad Urach, 2020. Version 01.01 (09 - 2020).

[7] Robert D. Cook, David S. Malkus, Michael E. Plesha, and Robert J. Witt. *Concepts and applications of finite element analysis*. Wiley, New York, NY, 4th edition, 2002.

[8] Justus Blum. Dynamische auslegung eines heliostaten unter berücksichtigung der fluktuierenden windlasten mittels fem-simulation, 2024.

[9] Roy R. Craig and Andrew J. Kurdila. *Fundamentals of Structural Dynamics*. Wiley, 2nd edition, 2006.

[10] MathWorks. Flexible body model builder app. `https://www.mathworks.com/help/sm/ref/flexiblebodymodelbuilder-app.html`. Accessed: 11.06.2024.

[11] MathWorks. Spherical Joint. `https://www.mathworks.com/help/sm/ref/sphericaljoint.html`. Accessed: 1.7.2024.

[12] MathWorks. Rectangular Joint. `https://www.mathworks.com/help/sm/ref/rectangularjoint.html`. Accessed: 1.7.2024.

[13] MathWorks. Reduced order flexible solid. `https://www.mathworks.com/help/sm/ref/reducedorderflexiblesolid.html`. Accessed: 11.06.2024.

[14] Andreas Pfahl. Heliostat fatigue loads: Proposal for estimation of wind load collectives. In *SolarPACES 2023*. German Aerospace Center (DLR), Institute of Solar Research, Solar Power Plant Technology, Germany, 2023. © Authors. This work is licensed under a Creative Commons Attribution 4.0 International License.

[15] Andreas Pfahl, Kristina Blume, Natalie Hanrieder, and Ralf Uhlig. Mahwin: Erforschung einer methode für die effiziente und sichere auslegung von kostengünstigen heliostaten durch vermessung und modellierung dynamischer, nicht-linearer windlasten. Technical report, Deutsches Zentrum für Luft und Raumfahrt e.V. (DLR), Institut für Solarforschung, 2021.

[16] MathWorks. Supported compilers. `https://www.mathworks.com/support/requirements/supported-compilers.html`, 2024. Accessed: 12.6.2024.

*APPENDIX* A

# Listings

```
clear
clc

%Ixx = 12179.33; % Example value for Ixx
%Iyy = 435.18; % Example value for Iyy

Ixx = 17454.13;
Iyy = 612.15;

[b, h] = matchSMOI(Ixx, Iyy);

disp(['Width (b): ', num2str(b)]);
disp(['Height (h): ', num2str(h)]);


function [b, h] = matchSMOI(Ixx, Iyy)
% Define the function to solve
function F = equations(vars)
b = vars(1);
h = vars(2);
% Equations based on the moments of inertia for a rectangle
F(1) = (b * h^3) / 12 - Ixx;
F(2) = (h * b^3) / 12 - Iyy;
end

% Initial guesses for width (b) and height (h)
initialGuess = [1, 1];

% Solve the system of equations
options = optimoptions('fsolve', 'Display', 'off');
[sol, ~, exitflag] = fsolve(@equations, initialGuess, options);

% Check if the solution is valid
if exitflag <= 0
error('No solution found. Try different initial guesses.');
end

% Extract the results
b = sol(1);
```

```
h = sol(2);

% Display the results
fprintf('Width (b): %.4f\n', b);
fprintf('Height (h): %.4f\n', h);
end
```

**Listing A.1** Determining Rectangular Beam Dimensions from Second Moments of Inertia

```
import xml.etree.ElementTree as ET
import os
import csv

# Define the path to the CSV file
file_path = os.path.join(os.path.expanduser('~'), 'Documents', 'Pressures_Wind-Tunnel.csv')

# Define the time index
t_b = 217.8358529 # Start at the t_b-th second
column_index = int(t_b * 193.05) # Calculate the index for frequency of 193.05

# Function to create XML files from pressure values
def create_xml_files(pressure_values, index):
# Create the root element
root = ET.Element("ANSYS_EnggData")

# Add child elements
material_data = ET.SubElement(root, "MaterialData")
convection_data = ET.SubElement(root, "ConvectionData")
load_variation_data = ET.SubElement(root, "LoadVariationData")
matml_doc = ET.SubElement(load_variation_data, "MatML_Doc")
load_variation = ET.SubElement(matml_doc, "LoadVariation")
bulk_details = ET.SubElement(load_variation, "BulkDetails")

# Set the name for each iteration
bulk_name = ET.SubElement(bulk_details, "Name")
bulk_name.text = "Druck" if index == 0 else f"Druck {index + 1}"

bulk_form = ET.SubElement(bulk_details, "Form")
bulk_description = ET.SubElement(bulk_form, "Description")

# Convert pressure from MPa to Pa and add property data elements
property_data_values = [str(pressure_in_MPa * 1) for pressure_in_MPa in pressure_values]
property_data = ET.SubElement(bulk_details, "PropertyData", property="pr1")
property_data_data = ET.SubElement(property_data, "Data", format="float")
property_data_data.text = ",".join(property_data_values)
property_data_qualifier = ET.SubElement(property_data, "Qualifier")
property_data_qualifier.text = "Pressure"

# Create time step values based on the length of property_data_values
num_data_points = len(property_data_values)
time_steps = [i * 0.00518 for i in range(num_data_points)]
property_parameter_value = ET.SubElement(property_data, "ParameterValue", format="float",
    parameter="pa1")
property_parameter_value.text = ",".join(str(step) for step in time_steps)

metadata = ET.SubElement(load_variation, "Metadata")
parameter_details = ET.SubElement(metadata, "ParameterDetails", id="pa1")
parameter_details_name = ET.SubElement(parameter_details, "Name")
parameter_details_name.text = "Time"
property_details = ET.SubElement(metadata, "PropertyDetails", id="pr1")
property_details_name = ET.SubElement(property_details, "Name")
property_details_name.text = "Pressure"

beam_section_data = ET.SubElement(root, "BeamSectionData")

# Create the XML tree and save to a file
tree = ET.ElementTree(root)
file_name = f"C:\\Users\\helm_at\\Documents\\XML_writes\\Load_p1_{index + 1}.xml"
tree.write(file_name)

# Generate XML files for pressure data
# Define indices for processing
```

```
indices = [j for j in range(24) if j % 4 == 0 or j % 4 != 0]
pressure_data = []

# Read data from the CSV file
with open(file_path, 'r') as file:
reader = csv.reader(file)
for i, row in enumerate(reader):
if i in indices:
pressure_data.append([float(value) for value in row[column_index:column_index + 43]])

# Create XML files for each pressure data set
for i, data in enumerate(pressure_data):
create_xml_files(data, i)
```

**Listing A.2** Wind Tunnel Data in .xml File

```
import os
import csv
import numpy as np
import scipy.io

# Define the path to the CSV file
file_path = os.path.join(os.path.expanduser('~'), 'Documents', 'Pressures_Wind-Tunnel.csv')

# Define tile dimensions
tile_lengths_x = [0.23, 0.29, 0.29, 0.29, 0.29, 0.23]
tile_lengths_y = [0.25, 0.35, 0.35, 0.25]

# Initialize lists to store indices and calculated values
indices = [[], [], []]
trimmed_data = []

# Open the CSV file to read tile dimensions
with open(file_path, 'r') as file:
reader = csv.reader(file)

# Calculate tile positions and areas
index_counter = 0
for i in range(24):
# Calculate tile position using Pythagorean theorem
a = tile_lengths_x[int(i/4)] / 2 + (abs(2 - int(i/4)) + int(i/12)) * tile_lengths_x[1]
b = tile_lengths_y[int(i%4)] / 2 + abs(1 - int(i%4) + int((i%4)/2)) * tile_lengths_y[1 - int(i%4)]

# Append indices and calculated values to lists
indices[0].append(index_counter) # Index excluding specific lines from CSV
indices[1].append(tile_lengths_x[int(i/4)] * tile_lengths_y[i % 4]) # Tile area
indices[2].append((a**2 + b**2)**0.5) # Distance from origin
index_counter += 1
if index_counter % 4 == 0: # Adjust index counter if necessary
index_counter += 4

# Open the CSV file to read data
with open(file_path, 'r') as file:
reader = csv.reader(file)

# Extract and trim data based on indices
for i, row in enumerate(reader):
if i in indices[0]:
trimmed_row = [float(value) for value in row]
trimmed_data.append(trimmed_row)

# Specify the number of rows and columns to print
rows_to_print = 10
cols_to_print = 3

# Initialize variables for maximum force and corresponding time step
max_force = 0
max_time_step = 0

# Define the number of rows and columns
num_rows = len(indices[0])
num_cols = len(trimmed_data[0])
```

```
57
58 # Create a numpy array to store computed forces
59 forces_array = np.zeros((num_rows, num_cols))
60
61 # Compute forces and populate the array
62 for col in range(num_cols):
63 for row in range(num_rows):
64 force = trimmed_data[row][col] * indices[1][row]
65 forces_array[row][col] = force
66
67 # Define the directory and filename for saving the .mat file
68 save_directory = r'C:\Users\helm_at\Documents\MATLAB'
69 save_filename = 'forces_array.mat'
70 save_filepath = os.path.join(save_directory, save_filename)
71
72 # Save the array as a .mat file
73 scipy.io.savemat(save_filepath, {'forces': forces_array})
74
75 print(f"The data has been saved to '{save_filepath}'.")
```

**Listing A.3** Wind Tunnel Data in .mat File

*APPENDIX* B

# Simulink Simulation Optimization

The multibody model solver can be significantly optimized. This section outlines the steps taken to reduce the computational time for the multibody model.

## B.1 Solver, Step size & Tolerance Consideration

Models that work with *Reduced Order Flexible Solid* bodies should use stiff solvers, as recommended in their documentation [13]. These include *ode15s* and *ode23t*. Experimentation on the step size, relative tolerance (*reltol*) and absolute tolerance (*abstol*) gave insights into how these settings affect the execution time of a 0.01 s simulation:

| Solver | Step Size [s] | *reltol* | *abstol* | Duration [s] |
|---|---|---|---|---|
| ode15s | 5.18e-6 | 2e-1 | 1e-2 | 538.52 |
| ode15s | 2.59e-5 | 2e-1 | 1e-2 | 228.02 |
| ode15s | 2.59e-5 | 3e-1 | 2e-2 | 252.72 |
| ode15s | 2.96e-5 | 2e-1 | 1e-2 | 239.28 |
| ode23t | 5.18e-6 | 2e-1 | 1e-2 | 310.03 |
| ode23t | 2.59e-5 | 2e-1 | 1e-2 | 193.09 |

**Tab. B.1** Simulation Performance of Different Solvers and Parameters; Duration is the execution time for a 0.01 s long simulation

## B.2 Simulation Mode

Three simulation modes are included as standard features in Simulink: *Normal*, *Accelerator*, and *Rapid Accelerator*. Transitioning from the default mode, which is *Normal*, to *Accelerator* typically reduces execution time by approximately half, albeit with a slightly longer initialization time, which is negligible for simulations of the planned duration.

The *Rapid Accelerator* mode does not work with the default LCC compiler from MinGW64 included in the default Matlab installation. Several alternative compilers are available that can run the simulation in *Rapid Accelerator* mode [16]. For this thesis, Microsoft Visual C++ 2022 was chosen.

| Simulation Mode | Duration [s] | |
|---|---|---|
| | MinGW64 | Microsoft Visual C++ 2022 |
| Normal | 220.12 | 184.47 |
| Accelerator | 181.66 | 185.65 |
| Rapid Accelerator | - | 79.14 |

**Tab. B.2** Comparison Between Simulation Modes Using Different Compilers for a 0.01 s Long Simulation

Table B.2 shows how the two compilers compare in terms of execution time for the same 0.01 s long simulation as in Table B.1. In *Accelerator* mode, the compilers run at a similar speed, but in *Rapid Accelerator* mode, the Microsoft compiler reduces that time by more than half. However, the initialization time for the *Rapid Accelerator* mode is significantly longer.

Beyond optimizing the model's solver and compiler, the parameters of the model can also be adjusted. There are 24 data points that are loaded onto the Simscape model from the MATLAB workspace. These data structures are passed directly onto the model frames through a *Simulink-PS* block, which converts the data from Simulink to Simscape. This block has a built-in pass-through filter, which smooths the data and should generally improve the model's performance. Additional optimization steps include

- Turn on *Compiler Optimization Level* setting
- Disable *Echo Expressions Without Semicolons* setting
- Set *Detect Overflow* setting to none

## B.3 Model Fidelity

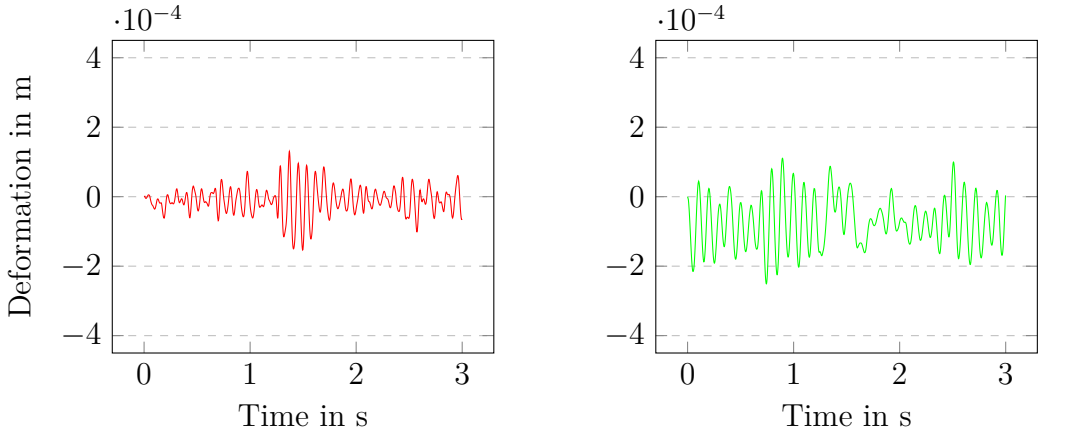
The ROM data was generated with the number of retained modes set to 12. In the documentation of the *Reduced Order Flexible Solid* blocks in Simulink, it is discussed

how computation time is affected by the number of modes, stating that the lower the amount of modes retained are, the shorter the computation time, with a setting of zero modes turning the model into a rigid body. To maximize the simulation speed without compromising the accuracy of the results, comparisons are made with different model configurations, where initially all modes are retained. Differences in transformation plots would indicate loss of data.

| Number of Modes | | | | Simulation | Duration |
|---|---|---|---|---|---|
| Solar Glass Mirror | Carrier Frame | T-Mount | Pylon | | [s] |
| 12 | 12 | 12 | 1 | Setup 1 | 744 |
| 8 | 12 | 12 | 1 | Setup 2 | 9.84 |
| 8 | 8 | 12 | 1 | Setup 3 | 0.07 |
| 6 | 6 | 12 | 1 | Setup 4 | 0.04 |
| 6 | 6 | 8 | 1 | Setup 5 | 0.02 |
| 6 | 6 | 6 | 1 | Setup 6 | 0.004 |

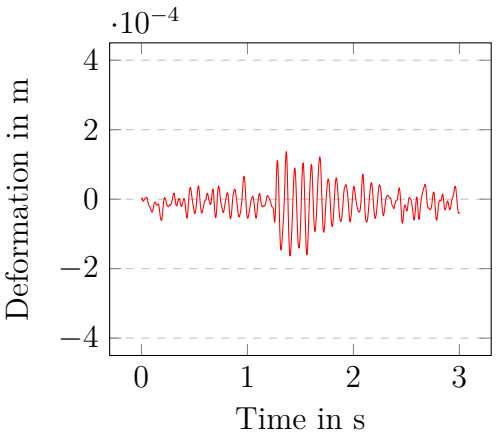
**Tab. B.3** Comparison Between Simulation Modes Using the MinGW64 Compiler; The entire simulation was 3 s but the duration only measures a 0.01 s simulation.

The setups described in Table B.3 are depicted below in Figures B.1 to B.6, which measure the deformation at the connection point from the pylon to the T-mount. The red graph represents the x-axis and green the y-axis. Deformation along the z-axis won't be shown here, as its values are negligible and to save space.

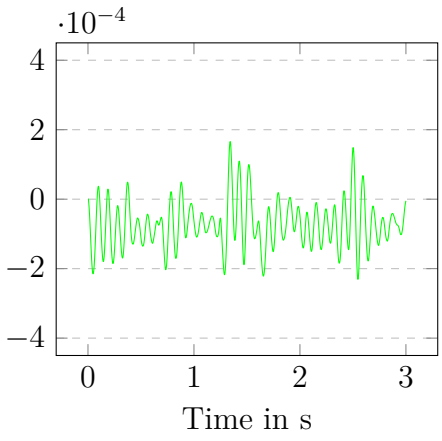



**(a)** Deformation Along the X-axis **(b)** Deformation Along the Y-axis

**Fig. B.1** Setup 1

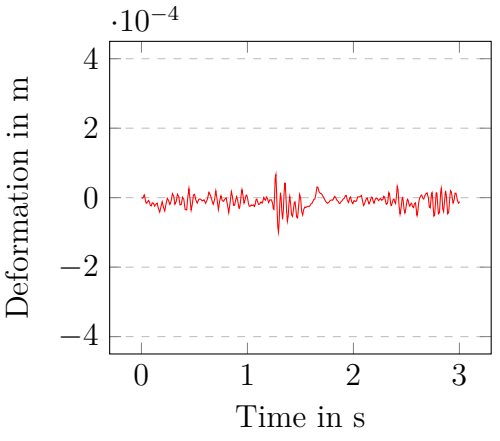



**(a)** Deformation Along the X-axis

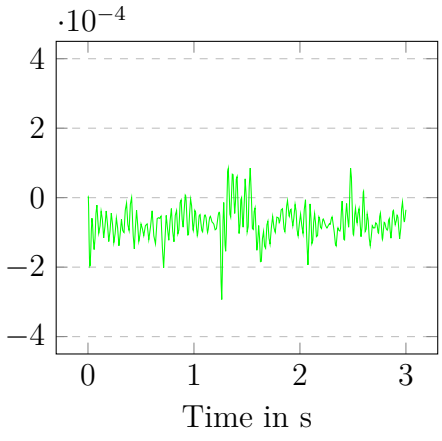



**(b)** Deformation Along the Y-axis

**Fig. B.2** Setup 2

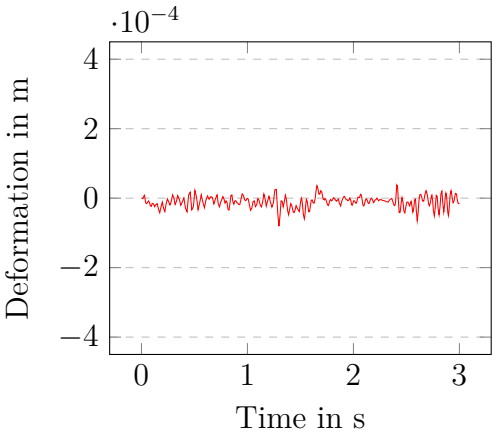



**(a)** Deformation Along the X-axis

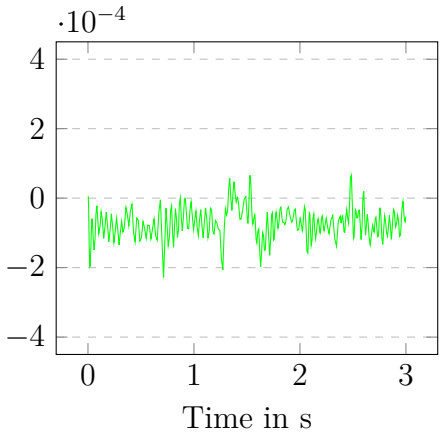



**(b)** Deformation Along the Y-axis

**Fig. B.3** Setup 3

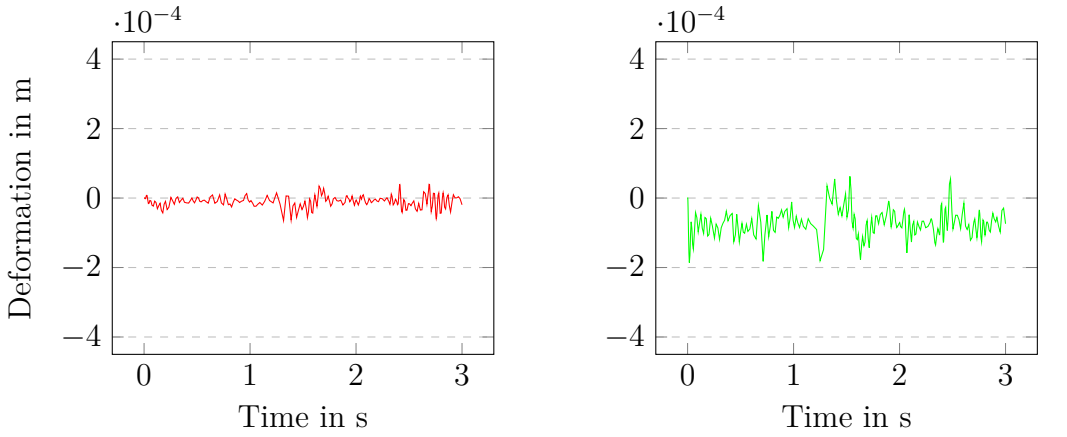



**(a)** Deformation Along the X-axis



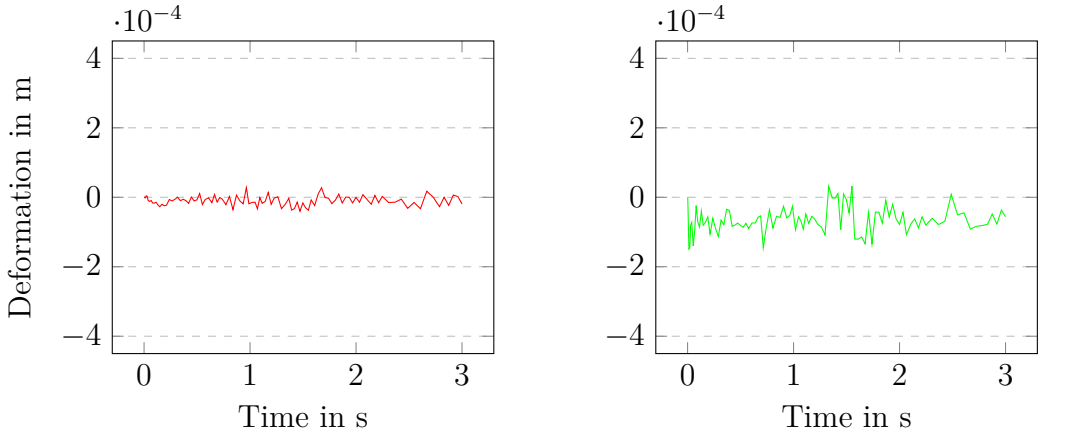

**(b)** Deformation Along the Y-axis

**Fig. B.4** Setup 4




**(a)** Deformation Along the X-axis **(b)** Deformation Along the Y-axis

**Fig. B.5** Setup 5




**(a)** Deformation Along the X-axis **(b)** Deformation Along the Y-axis

**Fig. B.6** Setup 6

Although the patterns of Setup 1 and Setup 2 differ slightly, as shown in Figures B.1 and B.2, respectively, they still resemble each other enough to justify the savings in the computation time for Setup 2, which are within two orders of magnitude. The setups that follow all have a severe loss of accuracy, deviating too far from Setup 1, regardless of the benefits in terms of computational time. For the final simulation, the modal reduction of Setup 2 is selected.

*APPENDIX* C

# Signal Data from the Full-Duration Study

This chapter of the appendix is the continuation of the results from Section 5.4, where the graphs for frames 2, 3, and 4 of the full-duration study are shown. This is meant to highlight the uniqueness of the measured frames of DLR's heliostat. The same methods as for the first frame were applied to obtain the following graphs.

## C.1 Full-Duration Deformation Signals

Figures C.1 and C.3 depict the raw signal over the entire duration of the load data, just like in Section 5.4.

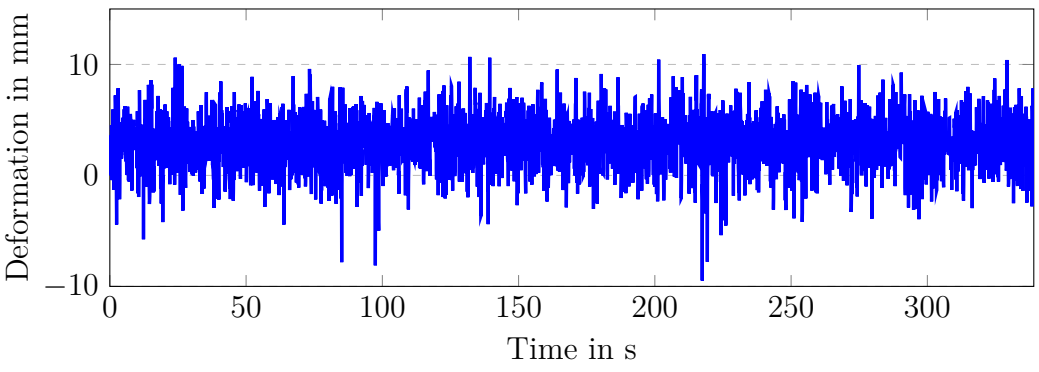



**Fig. C.1** Deformation Signal of Frame 2

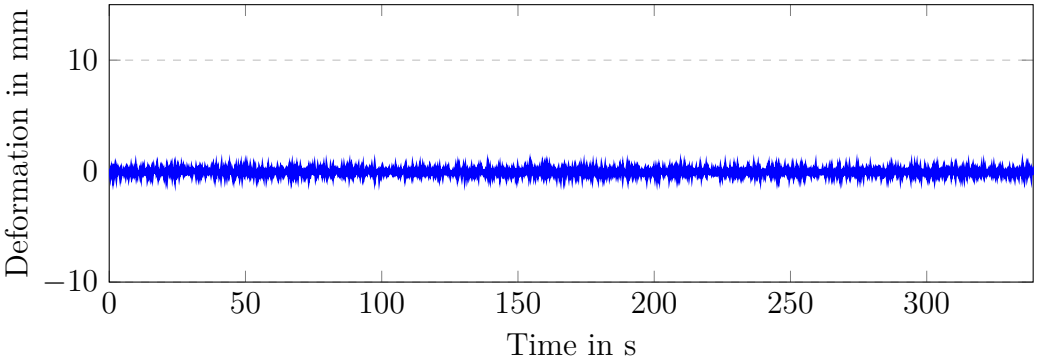



**Fig. C.2** Deformation Signal of Frame 3

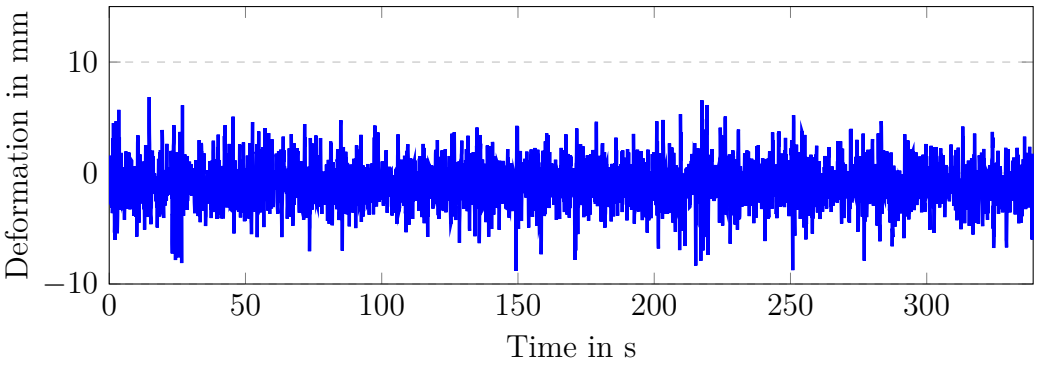



**Fig. C.3** Deformation Signal of Frame 4

## C.2 Time-Frequency Representations

Figures C.4 to C.9 include both STFT plots and spectrograms. The even-numbered figures display the STFT plots for the remaining frames from Section 5.4, while the odd-numbered figures show their corresponding spectrograms.

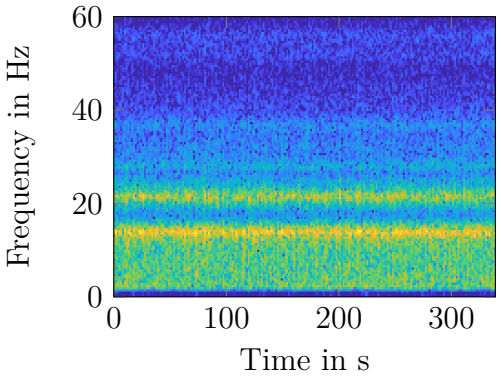



**Fig. C.4** STFT Plot for Frame 2

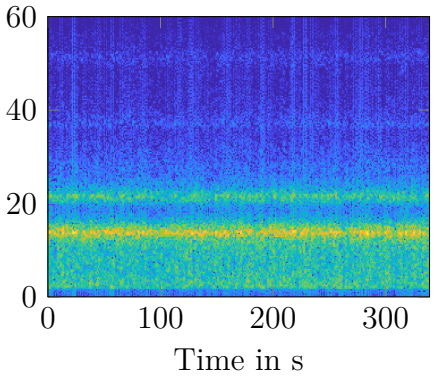



**Fig. C.5** Spectrogram for Frame 2

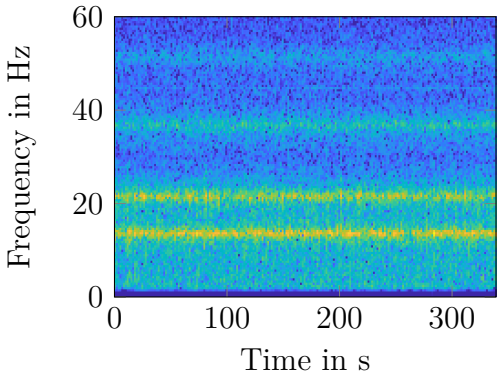



**Fig. C.6** STFT Plot for Frame 3

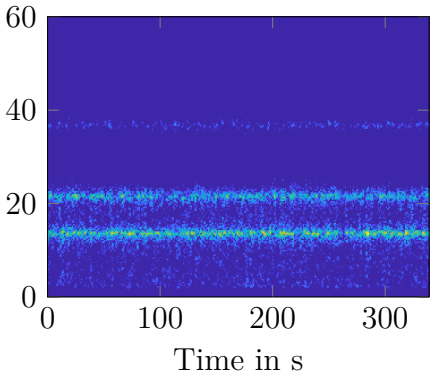



**Fig. C.7** Spectrogram for Frame 3

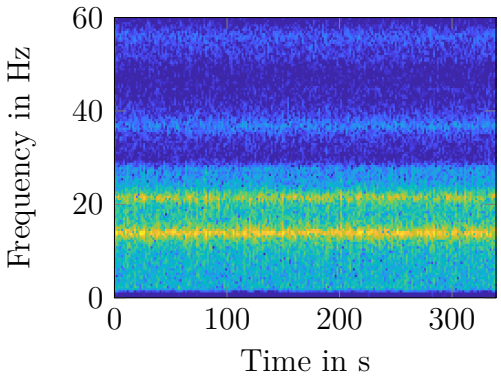



**Fig. C.8** STFT Plot for Frame 4

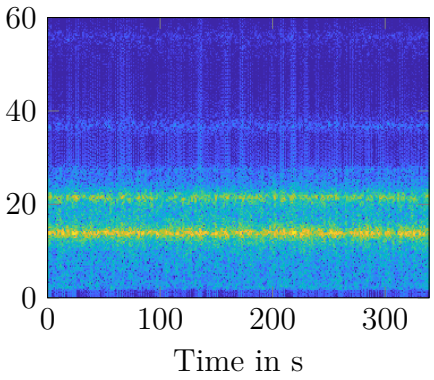



**Fig. C.9** Spectrogram for Frame 4

## C.3 3D Representations

The 3D Representation shown in Figure 5.18 in Section 5.4 is repeated for the remaining frames in Figures C.10 to C.12.

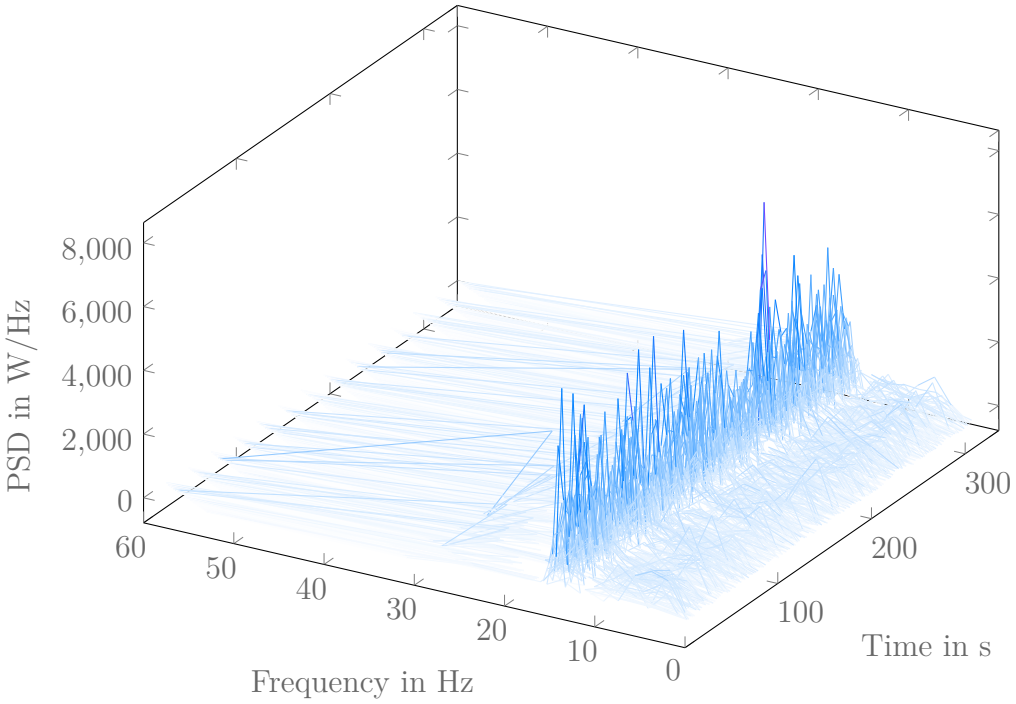



**Fig. C.10** 3D Representation of Figure C.5

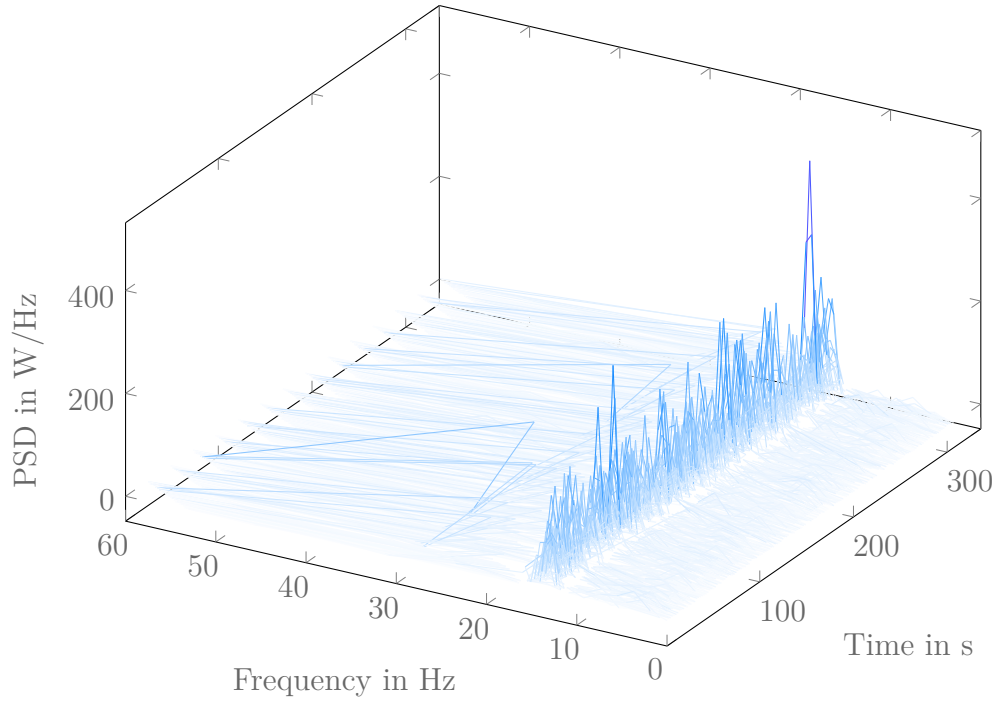



**Fig. C.11** 3D Representation of Figure C.7

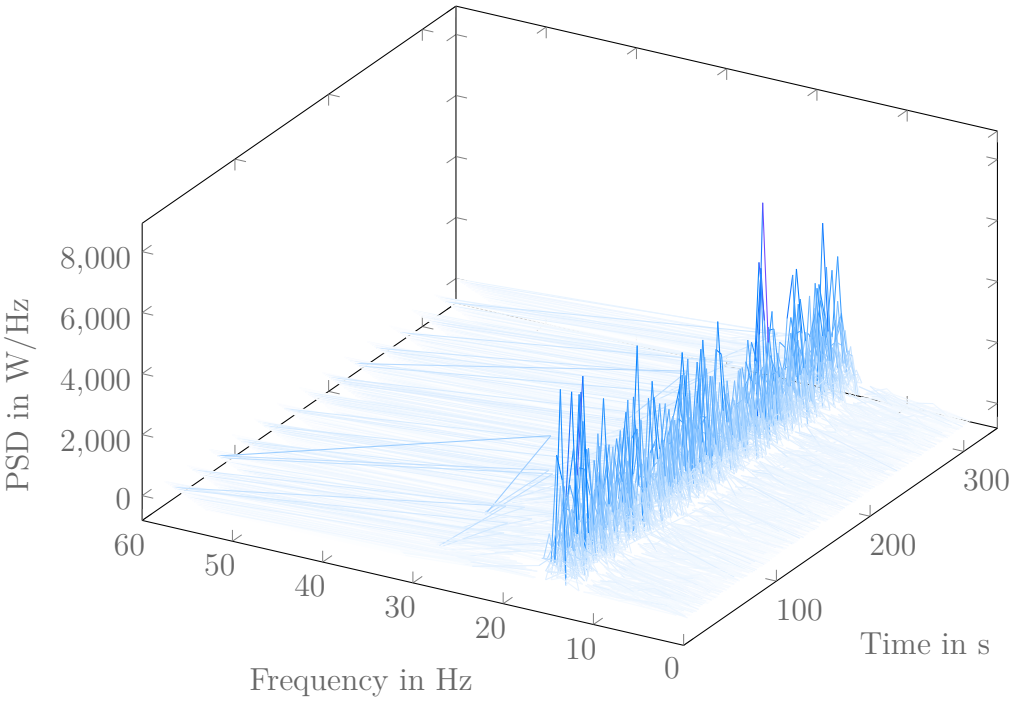



**Fig. C.12** 3D Representation of Figure C.9

# Declaration of Academic Integrity

I, Atli Tobiasson Helmer, hereby affirm that I have written this Master of Science thesis on my own and that I have used no sources or aids other than the ones stated.

This is the final version of the thesis, to be graded by the University of Applied Sciences Düsseldorf.

| | |
|---|---|
| Düsseldorf, 15.08.2024 | Atli Tobiasson Helmer<br>Atli Tobiasson Helmer<br>2024.08.15 17:49:41+02'00' |
| Place, date | Signature |